CENTRO UNIVERSITÁRIO DE JOÃO PESSOA - UNIPÊ
PRÓ-REITORIA ACADÊMICA - PROAC
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

SANSÃO FELICIANO COSTA

MÉTODOS ENSEMBLE COMO SUPORTE NO RECRUTAMENTO DE CANDIDATOS EM PROCESSOS SELETIVOS

JOÃO PESSOA – PB

2018

SANSÃO FELICIANO COSTA

MÉTODOS ENSEMBLE COMO SUPORTE NO RECRUTAMENTO DE CANDIDATOS EM PROCESSOS SELETIVOS

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação do Centro Universitário de João Pessoa - UNIPÊ, como pré-requisito para a obtenção do grau de Bacharel em Ciência da Computação, sob orientação do Prof. Ms. Fabio Falcão de França

JOÃO PESSOA - PB

2018

SANSÃO FELICIANO COSTA

MÉTODOS ENSEMBLE COMO SUPORTE NO RECRUTAMENTO DE CANDIDATOS EM PROCESSOS SELETIVOS

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação do Centro Universitário de João Pessoa - UNIPÊ, como pré-requisito para a obtenção do grau de Bacharel em Ciência da Computação, apreciada pela Banca Examinadora composta pelos seguintes membros:

Aprovada em _____/_____/2018.

BANCA EXAMINADORA

_____

Prof. Ms. Fabio Falcão de França (UNIPÊ)

_____

Prof. (título ex.: Dr./Ms./Esp.) Nome do professor Examinador (a) (UNIPÊ)

_____

Prof. (título ex.: Dr./Ms./Esp.) Nome do professor Examinador (a) (UNIPÊ)

DECLARAÇÃO

A empresa ....., representada neste documento pelo Sr.(a) ....., (cargo) ....., autoriza a divulgação das informações e dados coletados em sua organização, na elaboração do Trabalho de Conclusão de Curso intitulado (título) ....., realizados pelo aluno ....., do Curso de Bacharelado em Ciência da Computação do Centro Universitário de João Pessoa - UNIPÊ, com o objetivo de publicação e/ ou divulgação em veículos acadêmicos.

João Pessoa/PB, XX de XXXX de 2018.

_____

(assinatura)

(cargo)

(Nome da Empresa)

Dedicatória

A minha esposa Jesse Mariana e aos meus pais.

AGRADECIMENTOS

A Deus, por estar sempre me guiando pelos caminhos certos e me dado forças nos dissabores das decisões errôneas no meu dia a dia e em minha vida acadêmica;
Aos meus pais, por todo o esforço em meu favor;
Aos meus verdadeiros amigos e amigas.
A empresa Sol Saúde on Life, por permitir a realização deste experimento com candidatos para vaga de emprego disponibilizada pela mesma;

A Lydiene Fonseca, profissional de RH da empresa Sol Saúde on Life, por toda atenção ofertada e seu profissionalismo na realização do experimento proposto.

Ao meu orientador Fabio Falcão, por todas as experiências compartilhadas e por seu olhar crítico excelente que foi de grande auxílio na realização deste trabalho da melhor maneira possível.

RESUMO

A capacidade de recrutar e selecionar os melhores candidatos para uma vaga de emprego é uma das características que geralmente uma empresa espera encontrar ao contratar um Analista de Recursos Humanos. Tal profissional é incumbido do desafio de buscar possíveis candidatos que tenham o perfil mais aderente às vagas ofertadas pela empresa. Entretanto, embora esta atividade exija muita atenção e foco para alcançar uma boa assertividade nas escolhas realizadas, por vezes ela é realizada em paralelo a outras atribuições desse profissional, como treinar, integrar e motivar os demais colaboradores da empresa, além é claro das atividades burocráticas do setor. Diante disto, a ideia central deste trabalho é desenvolver um algoritmo computacional composto por um conjunto de algoritmos de inteligência artificial que venha a servir como suporte a um Analista de RH. Este algoritmo teria a função de encontrar, dentro de um conjunto de dados, os candidatos mais similares a uma vaga de emprego ofertada por uma empresa. Portando este trabalho objetivou concluir, por indução, se o uso deste algoritmo computacional pode vir a contribuir para maior assertividade no recrutamento de candidatos, proporcionando maior disponibilidade de tempo ao profissional de RH para realização das demais atividades deste setor.

Palavras-Chave: Candidatos. Vagas de Emprego. Recursos Humanos. Métodos Ensemble. Inteligência Artificial.

ABSTRACT

The ability to recruit and select the best candidates for a job opportunity is one of the characteristics that a company usually expects to find when hiring a Human Resources Analyst. Such a professional is entrusted with the challenge of finding possible candidates who have the profile most adherent to the job opportunity offered by the company. However, although this activity requires a lot of attention and focus to achieve a good assertiveness in the choices made, sometimes it is carried out in parallel with other duties of this professional, such as training, integrating and motivating the other employees of the company, in addition to bureaucratic activities of the sector. In view of this, the main idea of this work is to develop a computational algorithm composed by a set of algorithms of artificial intelligence that will serve as support to an HR Analyst. This algorithm would have the function of finding, within a set of data, the most similar candidates to a job opportunity offered by a company. This work aimed to conclude, by induction, if the use of this computational algorithm can contribute to greater assertiveness in the recruitment of candidates, providing a greater availability of time to the HR professional to perform the other activities of this sector.

Keywords: Applicants. Job Opportunity. Human Resources. Ensemble Methods. Artificial Intelligence.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

IA – Inteligência Artificial

RH – Recursos Humanos

SGBD – Sistema de Gerenciamento de Banco de Dados

SUMÁRIO

1 INTRODUÇÃO

O setor de Recursos Humanos pode ser considerado um dos principais setores das empresas que o possuem. Sua importância justifica-se, principalmente, por estar envolvido direta ou indiretamente com o funcionamento de todos os demais setores de uma organização, atuando como mediador entre os interesses dos colaboradores e os da empresa. Este departamento contribui para elaboração de planos de carreiras mais assertivos, estimulando o desenvolvimento de habilidades e competências de todos os profissionais contratados.

Contudo, algumas empresas, principalmente as de pequeno porte, ainda apresentam certa resistência em criar e estruturar um setor de RH, pois consideram que outros setores são mais merecedores de investimento, como, por exemplo, os setores comercial e financeiro (OLIVEIRA, 2010). De acordo com o RH Portal[1], conceituado site da área de recursos humanos, nas empresas de pequeno e médio porte geralmente o Departamento Pessoal e de RH se resumem em um setor único onde profissionais realizam tanto as tarefas relacionadas ao RH quanto ao próprio Departamento Pessoal. Contudo, como destaca CHIAVENATO (2014) existem distinções entre ambos os setores, pois enquanto o setor de Recursos Humanos é responsável por atrair, manter e desenvolver talentos em uma empresa enquanto o setor de Departamento Pessoal tem a atribuição de cuidar de toda a parte burocrática relacionada a contratações, demissões, elaborar folha de pagamento dos colaboradores, entre outras atividades.

Tal situação nas empresas pode ser considerada um dos motivos ocorrido para o aumento do índice de rotatividade (demissão e nova contratação de colaboradores) que atingiu a marca de 3,79% em 2017 e crescimento de 82% desde 2010, conforme levantamento da consultoria Robert Half[2], provocando um alto custo financeiro para empresas de todo o país. Em consonância com a realidade exposta acima, outra pesquisa, realizada pela HR Trends[3], constatou que 85% dos profissionais de RH entrevistados não desempenham por completo as atividades estratégicas que são de responsabilidade do setor de Recursos Humanos em uma

[1] Disponível em: https://www.rhportal.com.br/artigos-rh/recursos-humanos-x-departamento-pessoal/. Acesso em: 14 de outubro de 2018, às 12:12.

[2] Disponível em: https://www.catho.com.br/carreira-sucesso/colunistas/gestao-rh/brasil-tem-o-maior-indice-de-rotatividade/. Acesso em: 09 de setembro de 2018, às 20:01.

[3] Disponível em: https://www.propay.com.br/hrtrends. Acesso em: 14 de outubro de 2018, às 15:52.

empresa. Diante disto, torna-se evidente a dificuldade encontrada por estes profissionais em conciliar atividades burocráticas com as atividades estratégicas (treinamento, integração e motivação de funcionários) que podem ser consideradas as principais atribuições de um setor de RH.

1.1 RELEVÂNCIA DO ESTUDO

A relevância deste trabalho evidencia-se diante da identificação que o processo de recrutamento e seleção de candidatos às vagas de emprego de diversas empresas, principalmente as de pequeno e médio porte, não tem sido realizado da maneira mais adequada, seja por falta de interesse ou incapacidade financeira das empresas em possuir um setor exclusivo para os processos de RH, conforme já enunciado por OLIVEIRA(2010).

Como forma de tentar elucidar tal problemática, este trabalho sugere uma possibilidade de baixa complexidade de implantação, aliada a uma alta escalabilidade que consiste na utilização de um algoritmo computacional composto por um conjunto de algoritmos de inteligência artificial que possibilitem a identificação, dentre um grupo de candidatos, por aqueles que possuam maior aderência ao perfil de uma vaga disponibilizada por uma empresa, auxiliando a rotina do profissional de RH que durante a fase de recrutamento de candidatos tem muitas vezes que analisar dezenas de currículos de candidatos a uma vaga de emprego.

Caso a quantidade de vagas de emprego aumente ou venha a ser para cargos distintos a quantidade de currículos a serem analisados também aumenta consideravelmente, o que pode ser um dos motivos para a baixa produtividade, como destacada na pesquisa do Vagas.com, anteriormente citada, de um setor de RH composto por apenas um colaborador, ou com vários colaboradores, mas que se dividem entre as atividades do setor de RH e de Departamento Pessoal. Tal conjunto de algoritmos poderia resultar em uma fase de recrutamento de candidatos mais rápida e com um índice de assertividade maior em detrimento a mesma atividade sendo realizada por um único profissional ou por profissionais sobrecarregados com diversas atribuições difusas, acarretando em perda do foco e atenção que são necessários para realização deste procedimento tão importante para qualquer empresa.

Outro fator que o uso do algoritmo computacional poderia solucionar reside na possibilidade da escolha dos candidatos, realizada pelo Analista de RH, que serão recrutados

pode ser enviesada por critérios pessoais e/ou ideológicos, como foi relatado em uma pesquisa realizada pelo site Vagas.com[4], onde metade dos 3,2 mil candidatos entrevistados, compostos em sua maioria por pessoas do sexo feminino, negros e/ou deficientes, consideraram-se prejudicados em processos seletivos a qual foram participantes. Tal problema poderia ser eliminado com o uso do algoritmo proposto neste trabalho, que possibilitaria as mesmas chances de recrutamento para todos os profissionais que se candidataram a alguma vaga disponibilizada.

1.1 OBJETIVO GERAL

Implementar um algoritmo computacional composto por um conjunto de algoritmos de inteligência artificial com objetivo de auxiliar empresas no recrutamento de candidatos em processos seletivos.

1.2 OBJETIVOS ESPECÍFICOS

Como objetivos específicos tem-se:

- Analisar algoritmos de aprendizado supervisionado, não-supervisionado e por reforço, para escolha dos que serão utilizados na implementação do algoritmo computacional idealizado;
- Criar conjunto de dados contendo dados do currículo de candidatos;
- Criar grupo de candidatos por similaridade através de algoritmos de agrupamento e classificação, respectivamente;
- Realizar teste de classificação dos candidatos e vagas de emprego nos grupos pré-definidos pelo algoritmo de agrupamento;
- Validar acurácia do algoritmo computacional quanto a identificação dos candidatos mais similares a perfis de vagas de emprego.

[4] Disponível em: https://g1.globo.com/economia/concursos-e-emprego/noticia/metade-dos-profissionais-se-sentiu-prejudicada-em-processos-seletivos-aponta-pesquisa.ghtml. Acesso em: 15 de outubro de 2018, às 15:52.

1.3 INDICAÇÃO DA METODOLOGIA

A pesquisa realizada neste estudo é de natureza quantitativa que, de acordo com APPOLINÁRIO (2016) configura-se pela mensuração de variáveis pré-determinadas, visando verificar, descrever e explicar sua influência sobre outras variáveis, seguindo as seguintes etapas descritas abaixo:

1) Inicialmente foi realizada uma pesquisa bibliográfica sobre as técnicas de Inteligência Artificial, com foco nos tipos de algoritmos de aprendizado de máquina existentes, para subsequente identificação dos que melhor se adequam ao objetivo final do algoritmo computacional que será desenvolvido;
2) Foi adotado o procedimento experimental que, de acordo com GIL (2008, apud PRODANOV, 2013), consiste em submeter os objetos de estudo (neste caso o recrutamento de candidatos) à influência de certas variáveis (algoritmos de inteligência artificial), em condições controladas e conhecidas, para verificar os resultados no objeto analisado. Tal prática tem por objetivo verificar, por indução, pois é possível tecer generalizações derivadas de observações de casos da realidade concreta, como também defende PRODANOV (2013), se é possível identificar os candidatos mais similares ao perfil desejado para uma determinada vaga de emprego.

Para implementação do algoritmo computacional foram utilizados um conjunto pré-selecionado de algoritmos de inteligência artificial em Linguagem Python. Os dados dos candidatos e vagas de empregos foram disponibilizados pela empresa Saúde on Life Ltda e armazenado em banco de dados MySql. Foi utilizado um algoritmo de aprendizado de máquina para segmentar os candidato em grupos, de acordo com a similaridade de suas características, e elegendo um candidato hipotético como representante do grupo. Este representante será composto pela média das características dos candidatos que compõem o grupo. Esta segmentação foi realizada para que quando for disponibilizada uma vaga de emprego, o algoritmo computacional compare a mesma com os representantes de cada grupo, evitando que seja necessário comparar com toda a base de dados de candidatos. Este mesmo algoritmo também é utilizado para classificar um novo candidato que venha a ser inserido na base de dado, em algum dos grupos existentes.

Após identificar qual o representante de grupo mais similar a vaga disponibilizada, é realizado um cálculo de similaridade entre esta vaga e todos os candidatos que compõem o

grupo anteriormente identificado como possuidor do representante mais similar a ela, a fim de aferir qual o percentual de similaridade existente entre eles. Ao final da execução deste conjunto de algoritmos são retornados os candidatos mais similares aquela vaga de emprego em ordem decrescente de acordo com os percentuais de similaridades identificados e considerando um percentual mínimo de similaridades exigido pelo profissional de RH. Para validação do algoritmo computacional foi submetido o resultado da execução do mesmo a aprovação do profissional de RH que se pronunciou quanto a suas concordâncias e discordâncias quanto ao resultado alcançado. A assertividade do algoritmo foi calculada como a divisão entre a quantidade de escolhas realizadas pelo algoritmo e cujo profissional de RH foi de acordo pela quantidade total de escolhas realizadas pelo algoritmo.

.

1.4 ORGANIZAÇÃO DO TRABALHO

Após esse capítulo introdutório, o conteúdo deste trabalho organiza-se da seguinte forma:

- Capítulo 2 – RECRUTAMENTO DE CANDIDATOS E ENSEMBLE METHODS apresentará quais teorias e respectivos autores mais contribuíram para a realização do estudo e as bases teóricas para a realização deste trabalho;
- Capítulo 3 – DESENVOLVIMENTO apresentará passos realizados para construção do conjunto de dados, treinamento e teste dos algoritmos escolhidos para composição do algoritmo computacional e sua subsequente validação;
- Capítulo 4 – Considerações Finais apresentará de forma conclusiva, respostas aos objetivos específicos propostos pelo trabalho, apresentando também limitações desta pesquisa e trabalhos futuros.

2 ENSEMBLE METHODS E RECRUTAMENTO E SELEÇÃO DE CANDIDATOS

Este capítulo tem por objetivo enunciar, após pesquisa bibliográfica realizada, os principais conceitos fundamentadores do presente trabalho, bem como, demonstrar comparativo entre a aplicação desenvolvida e às aplicações similares existentes.

2.1. PROCESSO SELETIVO

Um processo seletivo consiste na identificação de conformidade entre as qualidades e/ou competências que uma empresa deseja e as pessoas que as possuem (CHIAVENATO, 2014). Contudo, encontrar tais possuidores de atributos essenciais e desejáveis para uma organização, por vezes tem se tornado um grande desafio para o setor de Recursos Humanos, quando existente, das empresas dos mais variados segmentos da atualidade. Embora subestimado, quanto à sua importância ou necessidade, por muitos gestores de empresas, o processo seletivo configura uma demonstração de planejamento antecipado, que visa evitar contratações precipitadas ou inadequadas, em função de urgência e/ou falta de planejamento (BANOV, 2015). Para desempenhar papel tão importante, um processo seletivo é subdividido em duas etapas essenciais, como descrito a seguir.

2.1.1 Recrutamento

Após realizada a definição do perfil do cargo e mapeadas as competências organizacionais e individuais, com uma ou mais vagas em aberto, é dado início o processo de recrutamento (BANOV, 2015). Segundo Chiavenato (2014), o recrutamento consiste em divulgar, através de diversos meios, as oportunidades que a organização pretende oferecer para as pessoas que possuam características específicas desejadas. Mediante a realização da exposição das vagas em aberto disponíveis, evidencia-se a importância de tal procedimento no sentido de atrair pessoas qualificadas dentro das necessidades da organização (BANOV, 2015).

Existem três tipos de recrutamento: recrutamento interno, recrutamento externo e recrutamento misto. Para Chiavenato (2014), o recrutamento interno tem foco nos candidatos que estão trabalhando dentro da organização, seus colaboradores, com objetivo de promoção ou transferência para outras atividades mais complexas ou mais motivadoras. Tal opinião é

semelhante à de Banov (2015), que conceitua o recrutamento interno como a divulgação de vagas sendo realizada dentro da própria empresa contratante, ofertando assim oportunidades a seu público interno. O recrutamento externo é conceituado por Chiavenato (2009), como sendo a busca por candidatos vindos de fora da organização, ou seja, havendo uma vaga de emprego a empresa busca este novo colaborador fora do seu quadro de funcionários. Esta modalidade de recrutamento é a que possui maior leque de possíveis canais de captação de candidatos para seleção. O quadro 1 descreve os principais canais de divulgação de vagas utilizados pelas organizações atualmente:

Quadro 1 – Principais canais de divulgação de vagas de emprego

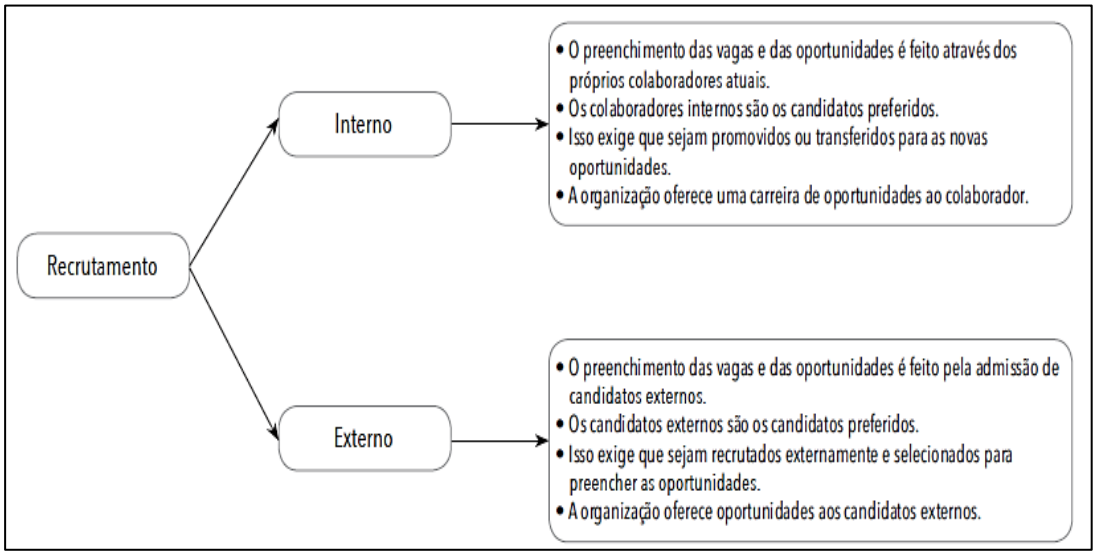
Canais de Recrutamento	Caracterização
Assessoria em Recursos Humanos	Empresas que fazem a triagem inicial e encaminham alguns candidatos para a organização contratante fazer a escolha.
Consultorias	Empresas que levantam o perfil do cargo e da cultura da empresa contratante para fazer o recrutamento e a seleção.
Networking	São redes de relacionamentos profissionais em que um profissional indica o outro por conhecer suas qualificações.
Jornais Impressos	Anúncios abertos ou fechados nos classificados de empregos existentes em periódicos diários de alcance local.
Anúncios em emissoras de radio	Geralmente são utilizados para cargos de menor prestigio social e quando há a necessidade de se recrutar muitas pessoas.
Serviços de alto-falantes	Realizadas em dispositivos de propagação de som nas imediações da empresa contratante, com o objetivo de atrair pessoas que morem próximo possível local de trabalho.

Fonte: (BANOV, 2015, p. 44-54).

O quadro 1 apresenta, de forma mais didática, as características e diferenças entre o recrutamento interno e externo. Os classificados em jornais impressos durante muito tempo foram o principal canal de divulgação de vagas de empregos, contudo, nas últimas décadas, com o vertiginoso aumento da facilidade quanto acesso por grande parte da população aos computadores, bem como, também, da possibilidade de acesso à internet por meio de dispositivos móveis, como smartphones e tablets, a maioria das empresas passaram a divulgar suas vagas em websites de classificados online de vagas de emprego, em suas próprias

páginas na internet, além, das principais redes sociais que possuem mais 130 milhões de usuários no Brasil.[5]

Figura 1 – Tipos de Recrutamento



Fonte: (CHIAVENATO, 2014, p. 102).

O recrutamento, contudo, não ocorre apenas de forma dicotômica como acima citado pois, se uma determinada empresa faz um recrutamento interno, um de seus colaboradores será realocado para a vaga em aberto, entretanto a vaga antes ocupada por este colaborador deverá ser suprida, provavelmente, através de um recrutamento externo, o que caracteriza, por fim, o processo conhecido como recrutamento misto.

2.1.2 Seleção

Segundo Banov (2015), devido às pessoas serem diferentes bem como, também, as empresas que as buscam, encontrar a pessoa certa para o cargo certo é o objetivo fundamental do processo de Seleção de Pessoal. Chiavenato (2014) define seleção como sendo a busca em meio aos candidatos recrutados por aqueles que sejam mais adequados aos cargos existentes na organização ou que possuam as competências requeridas pelo negócio, com objetivo de manter e, preferencialmente, aumentar a eficiência do desempenho humano, bem como a eficácia da organização. Evidencia-se, portanto, a importância de um recrutamento eficiente,

[5] Disponível em < https://www.techtudo.com.br/noticias/2018/02/10-fatos-sobre-o-uso-de-redes-sociais-no-brasil-que-voce-precisa-saber.html>. Acesso em 16 Abr. 2018.

que venha a atrair os candidatos com maior alinhamento a ideologia e objetivos da vaga disponibilizada, de modo a evitar desperdício de tempo e/ou recursos da empresa para realização de seleção com candidatos que não possuam os requisitos desejados e que, assim sendo, nem deveriam ter sido recrutados.

O quadro 2 apresenta as principais etapas realizadas em uma seleção de profissionais, segundo Banov (2015):

Quadro 2 – Principais etapas de uma seleção

Etapa	Procedimentos
Análise de currículo	Verificação se as habilidades, competências e características pessoais do candidato são compatíveis com o perfil da vaga disponibilizada.
Entrevista	Podem ocorrer de forma presencial, por telefone ou em videoconferência pela internet, tendo por objetivo avaliar no candidato o comportamento, a postura, entre outras características.
Aplicação de testes	Podem ser testes de conhecimentos gerais e/ou específicos, prova prática e testes psicológicos, que são realizados com objetivo principal de verificar a real existência das habilidades relatadas pelo candidato no currículo e/ou entrevista.
Dinâmica de grupo	Seu objetivo é observar como o candidato se comporta e se relaciona em grupo, suas características pessoais, como por exemplo, indecisão, iniciativa, argumentação ou como lida com pressões, conflitos, como resolve e soluciona problemas.
Exame médico específico	Realizado sempre em caráter eliminatório para verificar a aptidão física do candidato quanto a função a ser realizada pelo mesmo.

Fonte: (BANOV, 2015, p. 58-90).

Em vista disso, conclui-se, que realizado um recrutamento eficaz, é possível ao setor de Recursos Humanos de uma organização cumprir o principal objetivo de uma seleção, que consiste em escolher a pessoa certa para o lugar certo e no tempo certo. E, com objetivo de proporcionar maior eficiência na fase de recrutamento e seleção, dada a importância atribuída ao procedimento, é que o algoritmo computacional implementado neste trabalho faz uso de um conjunto de algoritmos de inteligência artificial, área de conhecimento e pesquisa a qual é definida a seguir.

2.2 INTELIGÊNCIA ARTIFICIAL

Embora esse termo, para algumas pessoas, remeta apenas a sua utilização na área da robótica, a Inteligência Artificial é utilizada por muitas outras áreas e com os mais variados fins, além de constituir-se uma vertente de pesquisa que demonstra estar longe de que sejam exauridas suas possibilidades de inserção nos mais distintos contextos ou ambientes. A seguir é feita a conceitualização deste tema, bem como, também, sua origem, exemplos de sua utilização na atualidade e dos principais algoritmos que a compõem, com foco para os relacionados ao aprendizado de máquina.

2.2.1 Conceito

O termo "Inteligência Artificial" ao longo das últimas décadas, mais precisamente desde a sua primeira utilização, em 1956, por John McCarthy em uma conferência no Dartmouth College, em Hanover, New Hampshire, já possuiu diversas e variadas tentativas de definição, contudo o termo demonstra ser de difícil conceitualização e gera controvérsias entre vários pesquisadores. Russel e Norving (2013) relaciona 8 definições para IA com foco em 4 estratégias de estudo utilizadas por diferentes pesquisadores, como consta no quadro 3:

Quadro 3 - Algumas definições de inteligência artificial

Pensando como um humano	Pensando racionalmente
"O novo e interessante esforço para fazer os computadores pensarem (...) máquinas com mentes, no sentido total e literal." (Haugeland, 1985) "[Automatização de] atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado..." (Bellman, 1978)	"O estudo das faculdades mentais pelo uso de modelos computacionais." (Charniak e McDermott, 1985) "O estudo das computações que tornam possível perceber, raciocinar e agir." (Winon, 1992)
Agindo como seres humanos	Agindo racionalmente
"A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas." (Kurzweil, 1990) "O estudo de como os computadores podem fazer tarefas que hoje são melhor desemprenhadas pelas pessoas." (Rich and Knight, 1991)	"Inteligência Computacional é o estudo do projeto de agentes inteligentes." (Poole *et al*., 1998) "AI... está relacionada a um desempenho inteligente de artefatos." (Nilson, 1998)

Fonte: (RUSSEL, NORVING, 2013, p. 25)

As definições dos autores que consta na parte superior da tabela são referentes ao conceito de IA em relação aos processos de pensamento e raciocínio, enquanto as definições da parte inferior relacionam a IA no âmbito do comportamento. De fato, a IA torna-se mais fácil de ser definida pelas propriedades que ela exibe: uma capacidade de lidar com novas situações, de solucionar problemas, de responder a questões, de elaborar planos e assim por diante. Em detrimento a toda a complexidade envolvida na definição de IA, a mesma pode ser conceitualizada, de forma mais simples e didática, como sendo uma forma de dar a uma máquina a possibilidade de realizar tarefas que uma criança é capaz de realizar, contudo que o mais poderoso computador existente ainda não é capaz de fazê-lo, pois mesmo podendo realizar cálculos demasiadamente complexos que um ser humano possivelmente levaria dezenas de anos para resolver o mesmo não possui capacidade para diferenciar uma cadeira de metal de uma cadeira de madeira, algo facilmente perceptível para uma criança de 3 anos (ROSA, 2011). Tal conceito expõe o principal objetivo da IA desde a sua idealização, que é assemelhar-se, o máximo possível, a capacidade de raciocínio e aprendizado do cérebro humano.

2.2.2 Aplicações da IA

Na atualidade é possível identificar diversas áreas nas quais já existem aplicações de técnicas da Inteligência Artificial. Independente de qual seja o ambiente que se observe, muito provavelmente será possível identificar o uso da Inteligência Artificial desde a implementação de jogos, demonstração de teoremas e até nas tarefas mais cotidianas das pessoas.

De acordo com Russel e Norving (2013), existem muitas atividades, em vários subcampos, que a IA já marcou presença, como seguem os exemplos:

- Reconhecimento de voz: Um viajante telefonando para uma empresa de transportes aéreos pode reservar um voo tendo toda a conversa guiada por um sistema automático de reconhecimento de voz e de gestão de diálogo;
- Tradução automática: Um programa de computador capaz de traduzir automaticamente de um idioma de origem para outro, permitindo o entendimento de textos escritos em qualquer idioma, por um leitor não nativo;

- Jogos: O DEEP BLUE da IBM se tornou o primeiro programa de computador a derrotar o campeão mundial em uma partida de xadrez, ao vencer Garry Kasparov em 1997;
- Robótica: A iRobot Corporation criou o mais robusto PackBot para o Iraque e Afeganistão, onde é usado para lidar com materiais perigosos, remover explosivos e identificar a localização dos franco-atiradores;

É facilmente perceptível, portanto, a importância da pesquisa e utilização das técnicas da IA no sentido de trazer melhorias cada vez mais significativas para a ciência e a humanidade. Dentre as diversas utilidades da IA anteriormente citadas, uma das vertentes de pesquisa que está sendo amplamente utilizada atualmente são os algoritmos de aprendizado de máquina.

2.3 APRENDIZADO DE MÁQUINA

Os seres humanos estão habituados a cotidianamente realizarem recomendações de diversos tipos a conhecidos ou mesmo solicitarem sugestões a especialistas em determinado assunto. Para que um ser humano realize tal ação é necessário que o mesmo possua prévio conhecimento sobre o assunto em questão, bem como também, sobre as preferências da pessoa a quem fornecerá a recomendação. Se essas recomendações forem solicitadas ou realizadas com frequência quem estiver realizando as recomendações poderá identificar essas preferências ou padrões de interesse inclusive sem que a pessoas que está solicitando a recomendação o informe de maneira explicita, ou seja, a pessoa irá aprender a preferência do outro de maneira implícita. Para que um sistema realize tal façanha é necessário que o mesmo tenha capacidade de, semelhantemente a um humano, aprender a identificar os mais diversos perfis de interesse que uma pessoa pode possuir. Uma vertente de pesquisa da IA que estuda e desenvolve algoritmos capazes de realizar tal procedimento é conhecida como aprendizado de máquina.

2.3.1 Conceito de Aprendizado de Maquina

Aprendizado de máquina é um segmento extremamente importante na Inteligência Artificial. Para Luger *(*2013 *apud* SIMON, 1983, p. 35) aprendizado de máquina é qualquer mudança em que um sistema que melhore o seu desempenho na segunda vez que ele repetir a mesma tarefa ou outra tarefa tirada da mesma população. Russel e Norving (2013) complementa que a aprendizagem pode variar do corriqueiro, como anotar um número de telefone, por exemplo, até o profundo, como mostrado por Albert Einstein, que inferiu uma

nova teoria para o universo. Atualmente ainda não é possível fazer computadores aprenderem tão bem quanto as pessoas, contudo algoritmos criados possuem eficiência na realização de várias tarefas de aprendizado, e os estudos teóricos mais recentes estão permitindo que novas técnicas sejam desenvolvidas.

Embora não consigam assemelhar-se de forma idêntica ao aprendizado o humano, o aprendizado de máquina tem sido utilizado em diversas aplicações e com os mais variados objetivos como reconhecimento de voz, detecção de fraudes, condução de automóveis de forma autônoma, diagnóstico de doenças, entre outras utilidades. Alguns fatores têm favorecido a expansão dessa área como o desenvolvimento de técnicas e algoritmos cada vez mais eficientes, bem como, o aumento expressivo na capacidade de recursos computacionais atualmente disponíveis.

É interessante constatar que enquanto o aprendizado humano é proveniente, principalmente, da observação de uma tarefa ou atividade sendo realizada por outro ser humano, e em algumas ocasiões podendo vir a ocorrer, também, de forma empírica, entre outras possibilidades de aprendizado, para uma máquina ou sistema, as possibilidades de aprendizado também não são únicas. Existem três tipos de feedback que determinam os três principais tipos de aprendizagem, de acordo com Russel e Norving (2013), que são: Aprendizado Supervisionado, Aprendizado Não Supervisionado e Aprendizado por Reforço. Como o aprendizado de máquina é uma das vertentes de pesquisa mais amplas da Inteligência Artificial (COPPIN, 2017), há dezenas de algoritmos desenvolvidos para cada tipo de aprendizado de máquina existentes, contudo, o presente trabalho terá enfoque nos principais algoritmos, com base em sua utilização na atualidade.

2.3.2 Aprendizado Supervisionado

Russel e Norving (2013) definem aprendizado supervisionado como a seguinte equação:

> “ Dado um conjunto de treinamento de N pares de exemplos de entrada e saída (X_1, Y_1), (X_2, Y_2),… (Xn, Yn), onde cada Y*j* foi gerado por uma função desconhecida y = f(X), descobrir uma função h que se aproxime da função verdadeira f.” (RUSSEL; NORVING, 2013, p. 808).

É possível expressar essa definição de Russel de forma mais didática, caracterizando o aprendizado supervisionado como sendo um treinamento realizado com a máquina

apresentando-lhe exemplos de entradas e saídas desejadas com objetivo que a mesma aprenda uma regra geral que mapeia as entradas para as saídas que se deseja predizer a partir dos dados existentes, possibilitando também prever o comportamento da saída para novas entradas possíveis. O termo "supervisionado" origina-se da simulação de um "supervisor externo" conhecedor da saída desejada para cada exemplo do treinamento, podendo este avaliar a capacidade da máquina predizer os valores corretos de saída para os valores de entrada fornecidos, bem como, para novos exemplos possíveis de entrada. Essa forma de aprendizado caracteriza-se como indutiva pois tem como objetivo derivar conclusões gerais a partir de observações específicas (FACELI *et al*, 2011). Além de possuír um viés indutivo, o aprendizado supervisionado também possui um viés de busca, pois é necessário encontrar dentro do espaço de hipóteses possíveis por aquela que terá um bom desempenho mesmo em novos exemplos além do conjunto de treinamento. (RUSSEL; NORVING, 2013). Existem 2 tipos fundamentais de problemas nos quais é utilizado aprendizado supervisionado, de acordo com Coppin, 2017, que são Classificação e Regressão.

Um problema de classificação consiste quando o resultado desejado pertence a um conjunto finito de possibilidades e os resultados da previsão são de natureza distintas, como "sim" ou "não" (RUSSEL; NORVING, 2013). Um exemplo clássico seria o de mapear uma pessoa e classifica-la em masculino ou feminino. Para classificação, a técnica mais utilizada para a avaliação consiste em comparar os valores obtidos de saída com o modelo treinado, utilizando os exemplos de teste como entrada, com os valores de saída existentes nesses exemplos. Quanto aos problemas de regressão, os mesmos podem ser caracterizados o valor que se deseja prever segue um espectro numérico e continuo, podendo ser usado, também, para estabelecer relações entre variáveis. Um exemplo de problema de regressão seria computar uma linha de tendência de dados de vendas. A seguir são descritos os principais algoritmos existentes que são utilizados para solução de problemas de classificação, universo no qual se enquadra o algoritmo computacional desenvolvido pelo autor.

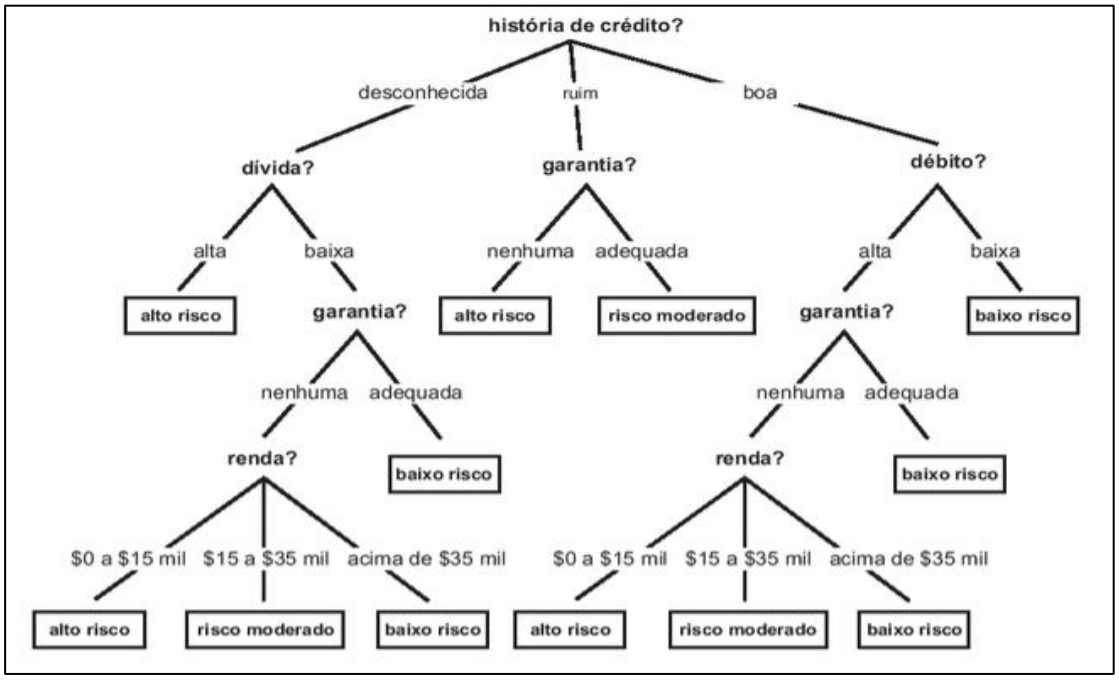
2.3.2.1 Algoritmo ID3

O algoritmo ID3 é um bastante utilizado para a implementação de uma Árvore de Decisão, um tipo de representação que permite determinar a classificação de um objeto testando seus valores para certas propriedades (LUGER, 2013). Ele é um algoritmo não-incremental, ou seja, consiste em uma estrutura construída a partir de um conjunto de dados a qual espera-se que as instâncias desconhecidas sigam a mesma distribuição do conjunto de

treinamento, de modo que para o modelo incluir novos exemplos, deve-se reaplicar o algoritmo para o novo conjunto de exemplos.

Tomando por base um conjunto de exemplos de treinamento e diferentes árvores que o classificam de maneira correta, poderia se perguntar qual árvore tem a maior probabilidade de classificar exemplos conhecidos e não conhecidos de uma população de dados? O Algoritmo ID3 infere que essa é a árvore mais simples que cobre todos os exemplos de treinamento. Tal suposição do algoritmo tem por fundamentação o conhecido principio da *Navalha de Occam*, o qual foi articulado pelo lógico medieval William de Occam, em 1324, o qual pressupõe que deve se preferir a simplicidade, evitando, sempre que possível, suposições desnecessárias. Não obstante as diversas utilizações deste famoso princípio, ele se encaixa de forma oportuna ao contexto do algoritmo citado pois se o mesmo possui exemplos de dados suficientes, obtidos no treinamento, para construir uma generalização válida então o mais eficiente seria, de fato, escolher a árvore que possui menor chance de incluir restrições desnecessárias. A figura 2 ilustra a metodologia de solução de uma árvore de decisão.

Figura 2 – Uma árvore de decisão para análise de crédito



Fonte: (LUGER, 2013, p. 340).

É perceptível que a árvore de decisão ilustrada acima possui muitas regras de restrições que, não necessariamente são essenciais para a configuração de uma generalização válida, devendo, portanto, pelo princípio da Navalha de Occam, ser rejeitada em detrimento a uma árvore mais simples e objetiva. Esta prática evidencia que o principal diferencial do

Algoritmo ID3 em relação à outros algoritmos de árvore de decisão existentes é a eliminação de redundâncias, de modo a resultar em uma implementação eficiente, pois performance configura ser um dos critérios mais exigidos para um algoritmo com tal finalidade, bem como, também, mais coerente e coesa, facilitando sua posterior leitura e entendimento, caso necessário. O algoritmo ID3 está apresentado na figura 3.

Figura 3 – Algoritmo ID3

Algorithm 1 Algoritmo ID3

ID3 (*Exemplos*, *Atributo_objetivo*, *Atributos*)
Exemplos: Os exemplos de treinamento
Atributo_objetivo: O atributo cujo valor deve ser predito pela árvore
Atributos: Lista com os atributos que serão utilizados para classificar os exemplos

1: Crie um nodo Raiz para a árvore
2: **if** todos os exemplos são positivos **then**
3: **return** Raiz da árvore, com o rótulo '+'
4: **else if** todos os exemplos são negativos **then**
5: **return** Raiz da árvore, com o rótulo '-'
6: **else if** Atributos for vazio **then**
7: **return** Raiz com o rótulo = valor mais comum do Atributo-objetivo em Exemplos
8: **else**
9: A ← o atributo de Atributos que melhor classifica Exemplos[1]
10: Raiz ← A (rótulo = atributo de decisão A)
11: **for** cada possível valor v_i de A **do**
12: Acrescenta um novo arco abaixo da Raiz, correspondendo à resposta $A = v_i$
13: Seja $Exemplos_{v_i}$, o subconjunto de Exemplos que tem valor v_i para A
14: **if** $Exemplos_{v_i}$ for vazio **then**
15: Acrescenta na extremidade do arco um nodo folha com rótulo = valor mais comum do Atributo-objetivo em Exemplos
16: **else** acrescenta na extremidade do arco a sub-árvore:
17: ID3($Exemplos_{v_i}$, Atributo-objetivo, Atributos -{A})
return Raiz

Fonte: (ROZA, 2016, p. 21).

Além do algoritmo citada acima outro algoritmo, também muito utilizado na atualidade com objetivo de classificação é o Classificador de Naive Bayes.

2.3.2.2 Classificador Naive Bayes

O Classificador Naive Bayes, tem por base o teorema de Bayes, formulado por Thomas Bayes, matemático e teólogo inglês que viveu de 1702 a 1761. O seu teorema é amplamente usado atualmente para lidar com situações nas quais não há certeza. O termo em língua inglesa “naive”, que traduzido para o português significa “ingênuo”, se deve ao fato que o algoritmo assume que a existência de uma característica particular em um objeto não é relacionada com a presença de nenhuma outra característica. Por exemplo, uma fruta pode ser considerada uma laranja se ela for de cor verde ou alaranjada, redonda e tenha 8 polegadas de diâmetro. Mesmo que essas características dependam entre si ou da existência de outras características, um classificador Naive Bayes iria considerar que todas essas características contribuem de forma independente para a probabilidade de que esta fruta seja uma laranja. A metodologia aplicado no Classificador de Bayes configura como raciocínio probabilístico, usado para discutir eventos, categorias e hipóteses sobre as quais não se tem 100% de certeza (COPPIN, 2017).

O teorema de bayes, fundamentador do classificador citado, é um corolário da lei da probabilidade total, expresso matematicamente na forma da seguinte equação:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad \text{em que A e B são eventos e P (B) } \neq 0. \qquad (2.1)$$

Basicamente o teorema acima funciona da seguinte maneira, estima-se a probabilidade do evento A ocorrer em função do evento B ocorrer, para tal multiplica-se a probabilidade do evento B ocorrer tendo o evento A já ocorrido pela probabilidade do evento A ocorrer, em seguida, divide-se o produto obtido pela probabilidade do evento B ocorrer. Evidencia-se, portanto, que foco do teorema é a probabilidade condicionada, ou seja, fala da probabilidade de uma teoria ou hipótese ser verdadeira se tiver ocorrido determinado acontecimento anterior. Coppin (2017, cap. 12) descreve um exemplo da aplicação do teorema de Bayes como expresso a seguir:

> *Vamos examinar um simples exemplo que ilustra o uso do teorema de Bayes em situações de diagnósticos médicos. Quando uma pessoa está gripada, geralmente tem temperatura alta (digamos que 80% das vezes). Podemos usar A para expressar "estou com temperatura alta" e B para, "estou gripado". Então, podemos estabelecer a probabilidade a posteriori como P(A|B) = 0,8. Observe que, neste caso, estamos usando A e B para representar fragmentos de dados que tanto poderiam ser hipóteses como evidências. É mais provável que usássemos A como evidência para nos ajudar a provar ou refutar a hipótese B, mas também poderia funcionar muito bem de forma inversa (pelo menos no sentido matemático). Agora vamos supor que também saibamos que, a qualquer momento, cerca de 1 em 10.000 pessoas fique gripada e de 1 em 1.000 pessoas tenha temperatura alta. Podemos escrever essas probabilidades a priori como P(A) = 0,001 e P(B) = 0,0001.*
>
> *Agora, suponha que você esteja com temperatura alta. Qual é a probabilidade de estar gripado? Isto pode ser calculado de forma bem simples pelo uso do teorema de Baye:*
>
> $$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \rightarrow P(A|B) = \frac{0{,}8 * 0{,}0001}{0{,}001} \rightarrow P(A|B) = 0{,}08$$
>
> *Então, mostramos que, só por ter uma temperatura alta, não necessariamente se torna muito provável que esteja gripado — na verdade, as chances de que você esteja gripado são de apenas 8 em 100.*

Com base no exemplo citado, é perceptível a vasta utilidade a qual um classificador de Naive Bayes pode ter, como o mesmo realmente é utilizado para os mais diversos fins como classificação de textos, filtragem de spams, análise de sentimento, entre outros. Contudo, em detrimento a suas variadas possibilidades de utilização, este algoritmo possui limitações como o problema de frequência zero, quando um determinado valor de atributo não aparece na base de treinamento, mas aparece no exemplo de teste, bem como, também, o problema de atributos correlacionados, os quais são votados duas vezes no modelo e podem levar a um excesso de importância. Tais limitações devem ser consideradas e corrigidas pelo implementador do algoritmo visando evitar ocorrências de falsos positivos ou falsos negativos. Outro algoritmo, que têm sido utilizado em diversas aplicações atuais é o algoritmo de Máquina de Vetores de Suporte (SVM).

2.3.2.3 Máquina de Vetores de Suporte (SVM)

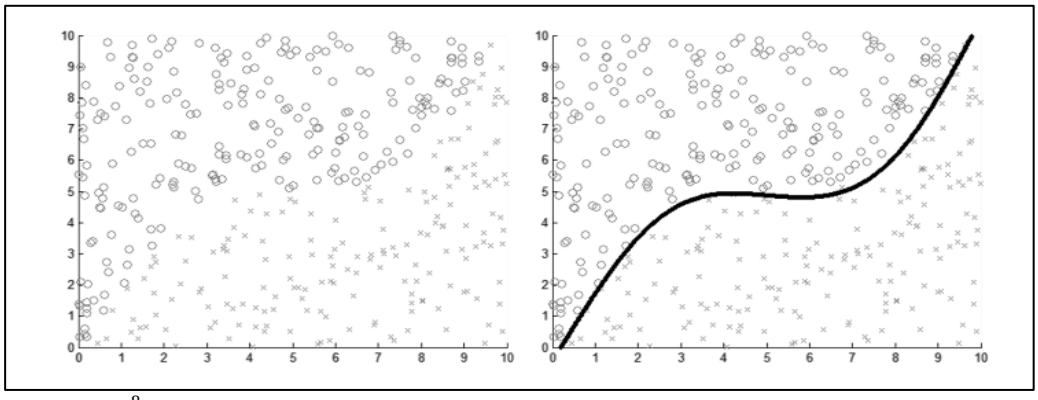
O algoritmo de Máquina de Vetores de Suporte (SVM), é fundamentado na Teoria da Aprendizagem Estatística, a qual foi desenvolvida por (VAPNIK, 1995), com objetivo de resolver problemas de classificacão de padrões. O mesmo toma como entrada um conjunto de dados, com o objetivo de predizer, para cada entrada dada, a qual de duas classes existentes a entrada faz parte, configurando, portanto, um SVM, diferentemente do classificador de Naive Bayes, como um classificador linear não probabilístico. Para o funcionamento deste algoritmo, cada dado é plotado como um ponto em um espaço n-dimensional (onde n é o número de características que se tem) com o valor de cada característica sendo o valor de uma coordenada particular. Por exemplo, se existem somente duas características como peso e comprimento do cabelo de um indivíduo, realiza-se a plotagem destas duas características no espaço onde cada ponto terá duas coordenadas, chamadas Vetores de Suporte.[6] Em outras palavras, o que uma SVM faz é criar uma linha de separação, mais comumente chamada de *hiperplano* entre dados de duas classes/atributos de entrada, com objetivo de maximizar a distância entre os pontos mais próximos em relação a cada uma das classes/atributos.

O número máximo de entradas para qualquer máquina de vetores de suporte especificada tende ao infinito, contudo, em termos práticos, a capacidade computacional limita a quantidade de entradas que podem vir a serem utilizadas. Se, por exemplo, N entradas

[6] Disponível em: <https://www.vooo.pro/insights/fundamentos-dos-algoritmos-de-machine-learning-com-codigo-python-e-r/>. Acesso em: 22 Abr. 2018.

são utilizadas para uma máquina de vetores de suporte específica esta máquina deve mapear cada conjunto de entradas no espaço de N dimensões com objetivo de encontrar um hiperplano de N-1 dimensões que possibilite melhor separa os dados de treinamento.[7] Uma máquina de vetores de suporte é, como facilmente se idêntica, uma máquina de entrada/saída a qual permite que um usuário inserira uma entrada e, com base no modelo desenvolvido através de treinamento, ela devolverá uma saída. A figura 4 ilustra o funcionamento de um SVM.

Figura 4 – Exemplo de funcionamento de um SVM



Fonte: Metatrader.[8]

É perceptível na figura 4, no lado esquerdo duas classes diferentes, representadas pelos círculos e pelas cruzes, as quais no lado direito da figura aparecem separadas por uma linha preta, chamada hiperplano divisor, indicando que realizou a classificação das mesmas. A SVM realiza primeiramente a classificação das classes, definindo assim cada ponto pertencente a cada uma dessas classes para que, em seguida, maximize a margem do hiperplano divisor, ou seja, após cumprida a primeira etapa de classificação ela subsequentemente, define a distância entre as margens em função dessa classificação. O último dos principais algoritmos classificadores existentes é o k-NN (vizinhos mais próximos).

2.3.2.4 kNN (Vizinhos mais próximos)

O algoritmo kNN pode ser usado para ambos os problemas de classificação e regressão, contudo, nas aplicações comercias da atualidade é mais amplamente utilizado em problemas de classificação. *K nearest neighbors*, que em tradução livre do inglês para o

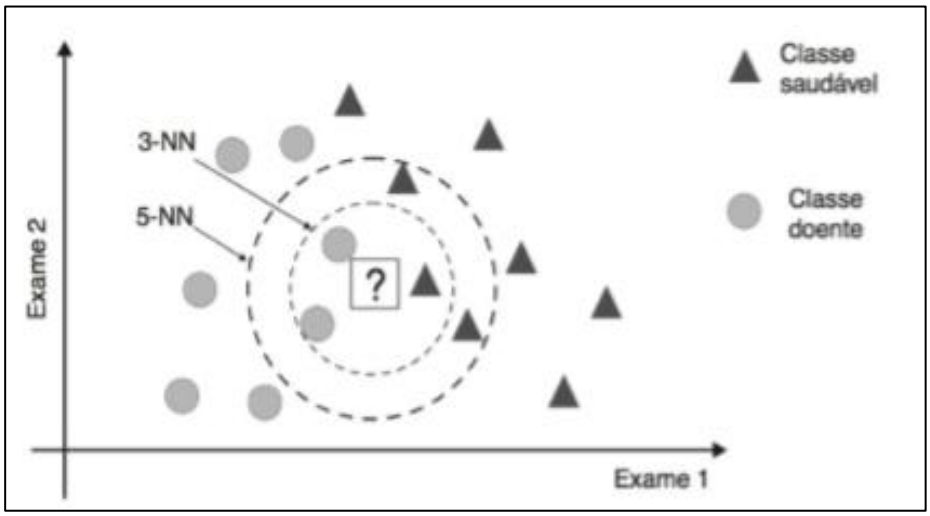
[7] Disponível em: <https://www.mql5.com/pt/articles/584>. Acesso em: 22 Abr. 2018.

[8] Disponível em: <https://www.mql5.com/pt/articles/584>. Acesso em: 22 Abr. 2018.

português seria K vizinhos mais próximos, é um algoritmo simples que consiste em armazenar todos os casos disponíveis e classificar novos casos por maioria de votos de seus k vizinhos mais próximos medidos por uma função de distância, ou seja, verifica-se quais são as classes desses k vizinhos e a que for mais frequênte será atribuída ao elemento desconhecido. (FACELI *et al*, 2011). Os dois componentes essenciais que devem ser determinados para aplicação do kNN são: a métrica de distância e o valor da constante k. Estre as métricas de distância utilizas por este algoritmo destacam-se a distância Euclidiana, de Manhattan, de Minkowski ou a de Hamming. As primeiras três funções são usadas para a função contínua e a quarta (Hamming) para variáveis categóricas.

A constante k, componente essencial do algoritmo, é definida pelo usuário. Se k definida como 1, então o caso é atribuído à classe de seu vizinho mais próximo. A escolha do valor para a constante k configura, muita das vezes como o maior desafio durante a execução de modelagem kNN. Tal dificuldade fica melhor compreendida através do exemplo contido na figura 5. Valores grandes atribuídos para a constante k podem provocar maior atenuação de distorções presentes nos dados, já que um maior número de exemplos será analisado, ou também suprimir características e tendências presentes em pequenos grupos de dados (ROZA, 2016). Essa dificuldade muitas vezes é contornada testando-se a qualidade do algoritmo para diferentes valores de k e, então, o que resultar em melhor desempenho classificatório é escolhido.

Figura 5 – Importância da constante k para o algoritmo kNN



Fonte: (FACELI *et al*, 2011).

Percebe-se, portanto, o grau de importância da definição do valor mais apropriado para o constante k cujo aumento ou diminuição pode impactar diretamente no resultado da

classificação realizada pelo algoritmo kNN. Para tal, em muitos casos, configura-se necessária a realização de treinamentos com objetivo de identificar qual o valor mais adequado para tal variável de modo a evitar classificações incorretas ou obtidas através do cálculo desnecessário da distância para uma quantidade exagera de vizinhos, aumentando o poder computacional exigido para execução do algoritmo e reduzindo sua performance. A figura 6 ilustra, em linguagem de pseudocódigo, o funcionamento do algoritmo kNN.

Figura 6 – Algoritmo kNN escrito em pseudocódigo

```
1  inicialização:
2      Preparar conjunto de dados de entrada e saída
3      Informar o valor de k;
4  para cada nova amostra faça
5      Calcular distância para todas as amostras
6      Determinar o conjunto das k's distâncias mais próximas
7      O rótulo com mais representantes no conjunto dos k's
8      vizinhos será o escolhido
9  fim para
10 retornar: conjunto de rótulos de classificação
```

Fonte: Computação Inteligente, Algoritmos.[9]

Embora os algoritmos mais populares e comercialmente utilizados sejam os de aprendizado supervisionado[10], a área de aprendizado de máquina também possui os algoritmos de aprendizado não supervisionado, os quais possuem diferenças conceituais bem como, também, de implementação e que são utilizados para finalidades, em sua maioria, relacionadas ao agrupamento de dados ou informações, entre outros fins os quais não necessitam da figura do "supervisor externo" para validar o resultado de sua execução. A seguir serão descritos os principais algoritmos dessa categoria bem como suas características e fins para os quais são utilizados.

2.3.3 Aprendizado Não Supervisionado

Os algoritmos de aprendizagem não-supervisionada recebem apenas dados de entrada e tem por função encontrar estrutura nas entradas fornecidas. Nestes algoritmos, não existem variáveis alvo, ou variáveis de saída para serem estimadas, por isso estes algoritmos realizam

[9] Disponível em: <http://www.computacaointeligente.com.br/algoritmos/knn-k-vizinhos-mais-proximos/>. Acesso em: 22 Abr. 2018.

[10] Disponível em: <http://joseguilhermelopes.com.br/introducao-ao-machine-learning-e-seus-principais-algoritmos/>. Acesso em: 22 Abr. 2018.

procedimentos conhecidos como tarefas de descrição (FACELI *et al*, 2011). O aprendizado não supervisionado pode ser um objetivo em si mesmo (descobrir novos padrões com base nos dados de entrada) ou um meio para atingir um fim, que em sua maioria consiste em representar um conjunto de dados através de um modelo de menor dimensão através de técnicas de agrupamento, associação ou sumarização.

Embora existam as 3 técnicas citadas, o que se verifica no uso do aprendizado não supervisionado nas aplicações atuais em sua maioria são voltadas para tarefas de agrupamento, com objetivo de reduzir o número de dimensões em um conjunto de dados para concentrar somente nos atributos mais úteis, bem como, também, para detectar tendências. Tal realidade é condizente com foco de utilização dos algoritmos desta categoria de aprendizado, descrito por Russel e Norving (2013), que é aprender padrões da entrada de dados com objetivo de detecção de grupos potencialmente úteis. A principal diferença desta forma de aprendizado em relação ao aprendizado supervisionado é que neste não há feedback com base nos resultados da previsão, ou seja, não há professor para corrigir o resultado obtido pelo algoritmo. A seguir são descritos os principais algoritmos de aprendizado não supervisionado com foco nos utilizados para tarefas de agrupamento, mais conhecida entre os estudiosos da IA como *clustering*.

2.3.3.1 Mapas de Kohonen

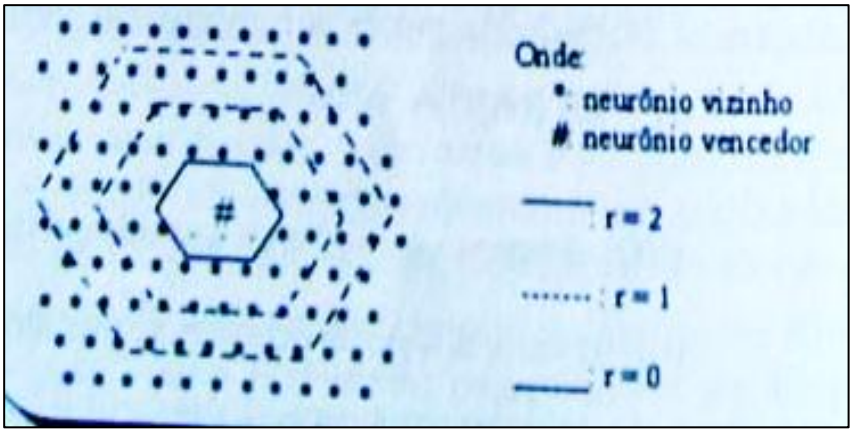
O Mapa de Kohonen, também conhecido como mapa de características auto organizáveis, é um tipo de rede neuronal desenvolvida por Teuvo Kohonen, cientista finlandês em 1982. O mapa de Kohonen usa o algoritmo “vencedor-leva-tudo”, o qual é um algoritmo de uma das subáreas do aprendizado não supervisionado conhecida como aprendizado competitivo. Este algoritmo usa o princípio de que apenas um neurônio fornece a saída da rede neuronal em resposta a uma entrada, este neurônio será o que tiver o maior nível de ativação. Um mapa de Kohonen não necessita ter informação de quais são as categorias que ele precisa criar, ele determina a segmentação mais útil por conta própria, tornando-o particularmente útil para tarefas cujo objetivo seja agrupar dados em grupos que não são conhecidos previamente (COPPIN, 2017).

A arquitetura de um mapa de Kohonen possui duas camadas sendo uma camada de entrada e outra de agrupamento, que serve como camada de saída. Cada nó de entrada é conectado a todo nó da camada de agrupamento e, tipicamente, os nós na camada de agrupamento são organizados em forma de grade. Há uma relação de vizinhança entre os neurônios (nós), mas deve haver uma relação entre os pesos dos neurônios no espaço de

dimensão igual ao número de entradas. O método usado para treinar um mapa de Kohonen consiste em inicialmente, atribuir pequenos valores aleatórios a todos os pesos. A taxa de aprendizado também é estabelecida, na maioria das vezes, como um pequeno valor positivo. (FERNANDES, 2005, *apud* STARKE, 1995).

Um vetor de entrada é inserido na camada de entrada do mapa. Esta camada obtém os dados de entrada para a camada de agrupamento. O neurônio (nó) da camada de agrupamento que melhor combine com os dados de entrada é declarado vencedor. Este neurônio fornece a classificação de saída do mapa e, também, tem seus pesos atualizados (COPPIN, 2017). O grid de saída pode ser de várias dimensões, com quantidade de elementos variável, como ilustra a figura 7.

Figura 7 - Arquitetura típica de uma rede de Kohonen



Fonte: (FERNANDES, 2005, p. 68).

Com base na figura acima o neurônio vencedor e seus vizinhos são chamados regiões que realizam o reconhecimento dos padrões de entrada e, com objetivo de que não exista conflito entre as diferentes regiões faz-se a redução do raio de vizinhança durante o treinamento, com a redução do número de vizinhos, em um momento final, a vizinhança tenderá a se resumir em apenas um neurônio, o vencedor, que terá seus pesos ajustados. A seguir será descrito um algoritmo semelhante que também é utilizado por muitas aplicações com objetivo de realizar agrupamentos.

2.3.3.2 Algoritmo Genético

De maneira semelhante ao Mapa de Kohonen, um tipo de rede neural com base no funcionamento do cérebro humano, os algoritmos genéticos, de maneira análoga, são baseados em uma metáfora biológica, onde seu funcionamento assemelha-se a uma competição em uma população de soluções evolutivas candidatas a solucionar um determinado problema (LUGER, 2013). Esse tipo de algoritmo consiste em uma técnica de

busca utilizada com objetivo de achar soluções aproximadas em problemas de otimização, o qual fundamentado principalmente pelo americano John Henry Holland, em 1975, sendo considerado uma classe particular de algoritmos evolutivos que usam técnicas inspiradas pela biologia evolutiva, idealizada por Charles Darwin, como hereditariedade, mutação e recombinação.

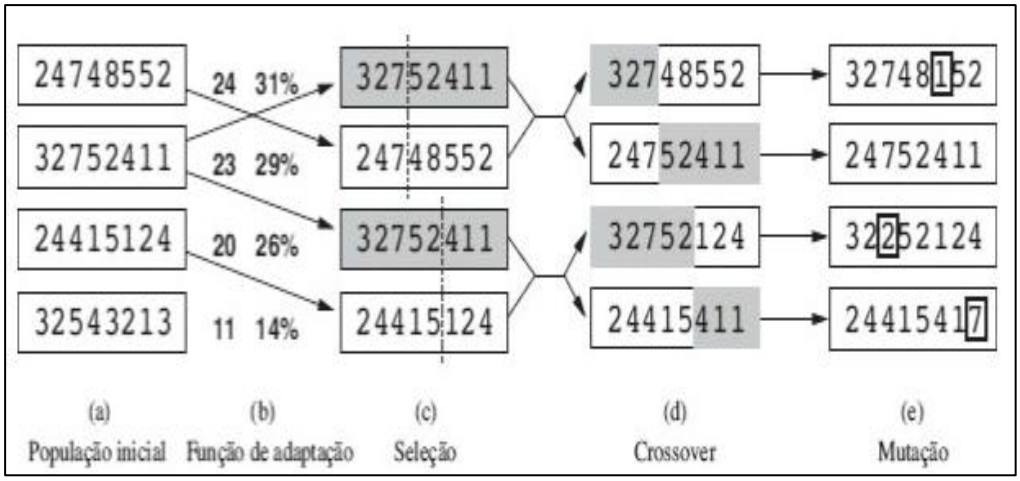
Este algoritmo consiste em receber um conjunto aleatório de soluções iniciais (população), em seguida realizará combinações dos melhores representantes dessa população, obtendo uma nova população que substitui a anterior, configurando uma nova geração. Tal mecanismo é repetido e a cada nova iteração a população é refinada gerando soluções novas e melhores para o problema que se quer solucionar, por fim essas iterações irão culminar na geração que representa a melhor solução (FERNANDES, 2005). Para Coppin (2017), o processo de execução de um algoritmo genético ocorre como a seguir:

1. Gere uma população aleatória de cromossomos (esta será a primeira geração).
2. Se o critério de terminação for satisfeito, pare. Caso contrário, siga para a etapa 3.
3. Determine a aptidão de cada cromossomo.
4. Aplique cruzamento e mutação a cromossomos selecionados, a partir da geração atual, para gerar uma nova população de cromossomos — a próxima geração.
5. Retorne à etapa 2.

Este algoritmo exige atenção a seus componentes essenciais como o tamanho da população, que deve ser determinado antecipadamente e que, na maioria das vezes, vai permanecer constante de uma geração para outra, exceto em algumas situações não quais configure ser útil ter uma população que mude de tamanho. Outro componente do algoritmo que exige cuidado é quanto ao tamanho de cada cromossomo, que deve permanecer o mesmo para que o cruzamento seja aplicado. Embora seja possível executar um algoritmo genético que possua tamanhos variáveis de cromossomo, isso não é comum (COPPIN, 2017).

No algoritmo genético ilustrado na figura 8, é possível identificar a população inicial em (a) a qual é classificada pela função de adaptação em (b), resultando em pares de correspondência em (c), estes produzem descendentes em (d) que serão subsequentemente sujeitos à mutação em (e). Por ser um algoritmo extremamente simples e eficiente, existem muitas variações deste algoritmo genético básico com objetivo de se obter resultados melhores ou mesmo vir a tratar novas classes de problemas.

Figura 8 - Fases de funcionamento de um algoritmo genético



Fonte: (RUSSEL; NORVING, 2013).

Outro algoritmo também muito utilizado para tarefas de *clustering* é o DBSCAN, descrito a seguir.

2.3.3.3 DBSCAN

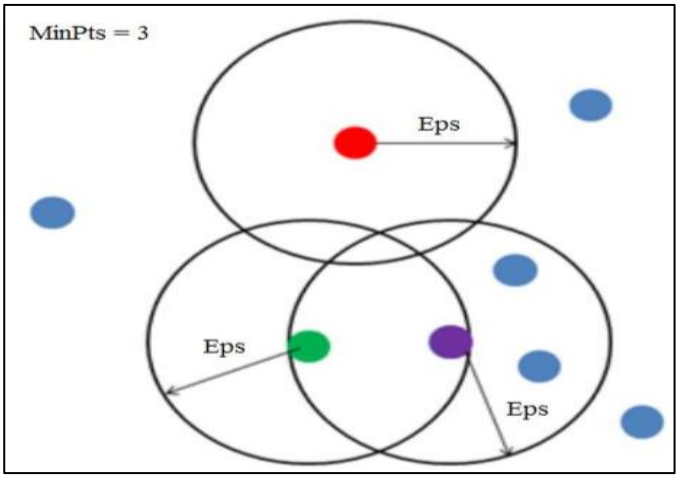
O algoritmo DBSCAN, é um tipo de agrupamento espacial baseado em densidade de aplicações com ruído o qual configura um método de agrupamento não paramétrico baseado em densidade, que foi incialmente proposto por Ester *et al* (1998). A funcionalidade do algoritmo consiste em identificar clusters de diferentes tamanhos de forma arbitrária, bem como, identificar e separar os ruídos dos dados para detectar clusters "naturais" e seus arranjos dentro do espaço de dados, não possuindo, contudo, qualquer informação preliminar sobre os grupos.

Para funcionamento do algoritmo utiliza-se apenas dois parâmetros de entrada: o primeiro refere-se ao número de pontos mínimos que um agrupamento deve possuir, denominado MinPts, o segundo é o raio em que será efetuada a busca por vizinhos a partir do ponto selecionado, denominado Eps (DANIEL, 2016 *apud* ESTER et al, 1996). Contudo, é importante ressaltar que a quantidade de clusters é encontrada pelo próprio algoritmo, diferentemente dos algoritmos anteriores, não sendo informada previamente pelo usuário (CASSIANO, 2015, cap. 5)[11]. A figura 9 ilustra o funcionamento do algoritmo DBSCAN. Nela é possível verificar o comportamento do algoritmo no agrupamento dos pontos, onde o

[11] Disponível em: <https://doi.org/10.17771/PUCRio.acad.24787>. Acesso em: 23 Abr. 2018.

ponto roxo é um centro, o verde é uma borda, o vermelho é um ruído e os azuis ainda não foram processados. Para encontrar os agrupamentos, o DBSCAN começa a busca por um ponto aleatório e compara a distância deste ponto com todos os demais da base de dados. Este ponto é marcado como centro caso uma circunferência de raio Eps tenha o número mínimo de pontos para formar o agrupamento. O mesmo pode ser considerado borda caso na circunferência Eps existam pontos alcançáveis, mas não o suficiente para formar um agrupamento. Por fim, o ponto pode ser denominado ruído, caso não seja alcançável por nenhum outro numa circunferência de raio Eps. Este procedimento é repetido até que todos os pontos sejam marcados (DANIEL, 2016).

Figura 9 - Funcionamento do DBSCAN na busca por agrupamento



Fonte: (VALÊNCIO et al, 2013)

As principais vantagens do uso do algoritmo DBSCAN, além da ausência de especificação do número desejado de clusters existente em outros algoritmos e que ele pode encontrar clusters com formas geométricas arbitrárias, como por exemplo um cluster completamente cercado, mas não conectado, por outro cluster. Outra vantagem deve-se ao parâmetro MinPts cujo efeito de link único é reduzido possibilitando que diferentes clusters possam serem conectados usando uma linha pontilhada fina (KRIEGEL, 2011). Contudo a principal vantagem dele configura-se por sua simplicidade, exigindo apenas dois parâmetros e não sendo suscetível à ordem em que os pontos estão localizados no banco de dados. Contudo, o mesmo também possui desvantagens como o fato de sua qualidade depender da noção de distância (medida de distância) utilizada na função regionQuery (P, e), bem como, também, não poder agrupar conjuntos de dados com grandes diferenças em densidades, uma vez que a combinação MinPts - *e* não pode ser escolhida corretamente para todos os grupos. O último dos principais algoritmos de clustering, K-means, é descrito a seguir.

2.3.3.4 K-means

É um dos mais utilizados algoritmos para resolver problemas de agrupamento. O termo "*K-means*" foi utilizado primeiramente por James MacQueen, estatístico escocês, em 1967, embora a ideia original do algoritmo ter sido concebida por Hugo Steinhaus, matemático polonês, em 1957. O K, de K-means, é referente a quantidade de centróides (pontos centrais dos grupos) que serão criados com objetivo de ajudar a encontrar a similaridade dos dados, essa variável k é definida previamente pelo usuário. O K-means configura-se, portanto, como uma heurística de agrupamento não hierárquico que busca minimizar a distância dos elementos a um conjunto de k centros dado por $X = \{x_1,x_2,...,x_k\}$ de forma iterativa (LINDEN, 2009). A distância entre um ponto p_i e um conjunto de clusters, dada por $d(p_i,x)$, é definida como sendo a distância do mesmo ao centro mais próximo dele.

O funcionamento deste algoritmo é bem simples, como descrito abaixo:

1. Escolher-se k diferentes valores, possivelmente, de forma aleatória, para os centros dos grupos a serem criados;

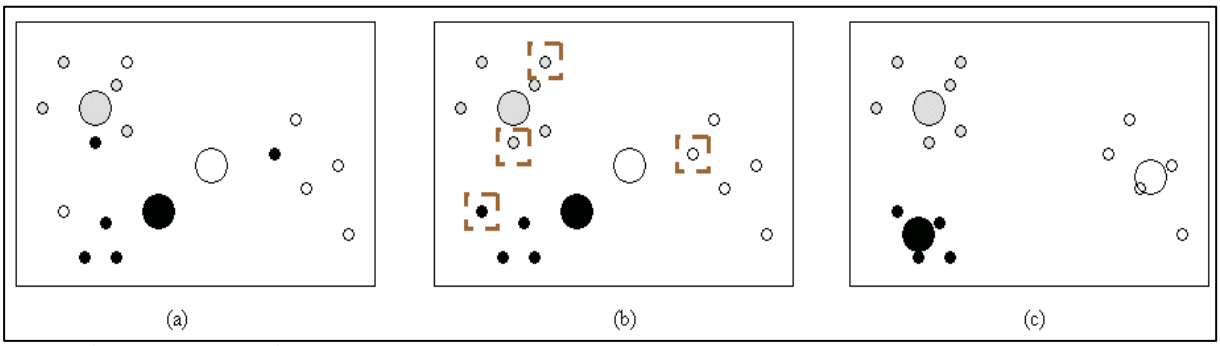
2. Realiza-se a associação de cada ponto ao centro mais próximo;

3. Recalcular-se o centro de cada grupo;

4. Repete-se os passos imediatamente anteriores (2 e 3) até que nenhum elemento mude de grupo.

Este algoritmo é extremamente rápido e geralmente converge dentro de poucas iterações para uma configuração estável, na qual nenhum elemento está designado para um cluster cujo centro não lhe seja o mais próximo. Outra vantagem desse algoritmo consiste no fato de que, devido a sua simplicidade, um programador experiente pode implementar uma versão própria do mesmo em cerca de uma hora de trabalho (LINDEN, 2009). A figura 10 ilustra o funcionamento deste algoritmo.

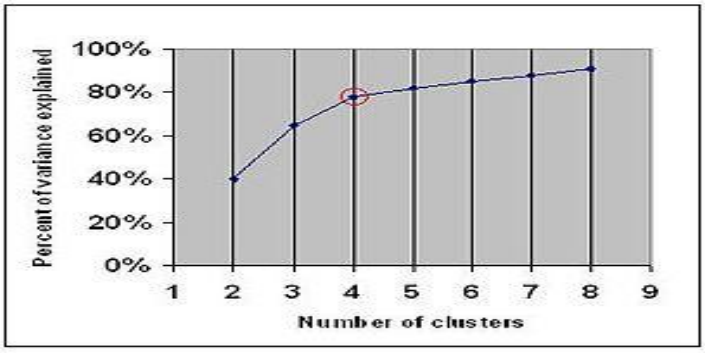
Figura 10 – Funcionamento do algoritmo K-means



Fonte: (LINDEN, 2009).

Em detrimento a sua simplicidade, um dos grandes desafios na implementação deste algoritmo consiste em determinar o valor adequado para a variável k. Existem várias categorias de métodos que podem ser utilizados para auxiliar ao usuário a tomar essa decisão, contudo descreve-se abaixo um dos mais práticos e mais utilizados para este fim, que é o Elbow Method, também conhecido como "método do cotovelo". Este método, idealizado por Robert L. Thorndike, psicometrista e psicólogo educacional estadunidense, em 1953, consiste em examina a porcentagem da variação como uma função do número de clusters, onde deve-se escolher um número de clusters o qual a adição de outro cluster não forneça uma modelagem muito melhor aos dados. Por exemplo, se alguém traça a porcentagem de variância mínima em relação ao número de clusters, os primeiros clusters adicionarão muita informação (implicarão em muita variação), mas em algum momento o ganho marginal será mínimo, dando um ângulo no gráfico. O número de clusters é escolhido neste ponto, o que explica "critério do cotovelo" (SHOOK, 1996). A figura 11 ilustra esse método.

Figura 11 – Funcionamento do Elbow Method.



Fonte: Wikipédia, Determinando o número de clusters. em um conjunto de dados.[12]

Mediante a figura 11 é possível identificar que o valor adequado para a variável k, com base no contexto próprio, é 4, pois verifica-se ser o ponto onde se forma o "cotovelo invertido" no gráfico. Este algoritmo, K-means, foi o escolhido pelo autor para realizar o agrupamento do currículo de candidatos, como base na similaridade entre eles. A razão desta escolha foi devido a sua simplicidade e eficiência, em comparação com outros algoritmos de agrupamento. Comparando-o com o algoritmo Mapa de Kohonen, a principal vantagem do K-means evidencia-se por possuir melhor performance (velocidade de execução), independente da quantidade de elementos a serem agrupados. Quando comparado com um algoritmo genético o *K-means* é considerado de menor complexidade, por possuir uma quantidade

[12] Disponível em: <https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set>. Acesso em 23 Abr. 2018.

reduzida de etapas de execução, além de ser mais intuitivo para implementação. E, por fim, confrontando o *K-means* com o algoritmo DBSCAN, é verificado que o segundo exige que seja informado o dobro de parâmetros obrigatórios que o primeiro, além de não garantir que, ao fim de sua execução, todos os elementos analisados sejam atribuídos a um grupo, devido a sua maior sensibilidade a *outliers* (valor atípico ou que apresenta um grande afastamento dos demais em uma série). Os algoritmos *k-medians* e *k-medoids*, que são variantes do algoritmo *K-means*, também foram considerados pelo autor, contudo a razão da escolha pelo algoritmo *K-means*, em detrimento a eles, é descrita a seguir.

Em relação ao *k-medians*, este algoritmo calcula a mediana entre os componentes de um grupo para definição dos centroides, enquanto o *K-means* calcula a média como já citado, o que vem a exigir maior poder computacional pois o cálculo da mediana é mais complexo comparado ao da média. Outro motivo da escolha pelo *K-means* é o fato do *k-medians* ser mais sensível a *outliers*, porém não apresentar resultado tão satisfatório como o *K-means* quando trabalha com dados normalizados, como os dados das características dos candidatos as vagas de emprego que serão fornecidas pelo autor.

Quanto ao *k-medoids* os fatores preponderantes para sua não utilização foram sua exigência que o centroide fosse um dos integrantes do grupo aliado a percepção que embora este algoritmo seja menos sensível a *outliers* do que o *K-means*, em contra partida ele precisa de maior poder computacional para tratar dados de alta dimensão.

Uma última forma de aprendizado além do supervisionado e não supervisionado já descritos, é o aprendizado por reforço, que possui suas características e utilidades descritas a seguir.

2.3.4. Aprendizado por reforço

Essa forma de aprendizado é normalmente utilizada nas áreas da robótica, jogos e navegação. Com ela, o algoritmo descobre através de testes do tipo “tentativa e erro” quais ações rendem as maiores recompensas (RUSSEL; NORVING, 2013). Este tipo de aprendizagem tem três componentes principais: o agente, caracterizado como aprendiz ou tomador de decisão, o ambiente que consiste em tudo com que o agente interage e ações, as possíveis atitudes que o agente pode executar.

O objetivo desse tipo de aprendizado é que o agente escolha ações que maximizem a recompensa esperada em um período determinado. O agente atingirá o objetivo muito mais rápido se seguir uma boa política. Então o foco da aprendizagem de reforço é descobrir a

melhor política (MONTEIRO; RIBEIRO, 2004). Aprendizado por reforço se distingue do problema do aprendizado supervisionado no sentindo em que pares de entrada e saída corretos nunca são apresentados, nem as ações sub ótimas são explicitamente corrigidas. Diversos algoritmos usados para solucionar problemas, incluindo planejadores, tomadores de decisão e realização de busca podem ser vistos no contexto do aprendizado por reforço, evidenciando sua importância e utilidade para a área de Inteligência Artificial e Aprendizado de Máquina, em especial (LUGER, 2013). Para implementação do algoritmo computacional proposto serão utilizado um conjunto de algoritmos de aprendizado de máquina escolhidos pelo autor, dentre os citados anteriormente, o que consiste em uma estratégia conhecida como *Ensemble Methods*, cujas características são descritas a seguir.

2.4 ENSEMBLE METHODS

Em tradução livre do inglês o termo *Ensemble Methods* seria algo como Conjunto de Métodos, e de fato a ideia central de tal abordagem consiste em ser uma estratégia de utilização de dois ou mais métodos, ou algoritmos, para resolução de um problema com objetivo de melhorar o resultado que seria obtido em caso de utilização de apenas um algoritmo[13]. Tal prática pode ser considerada, em alguns casos, como mais assertiva que utilizar apenas um algoritmo e esperar que ele seja o melhor, ou mais preciso, em detrimento a uma miríade de algoritmos que poderiam ser levados em consideração para produzir um algoritmo final com desempenho possivelmente superior (ZHOU, 2012).

Esta estratégia do uso de algoritmos combinados já se mostrou muito eficiente, inclusive sendo vencedora de diversas competições de Aprendizado de Máquina, como por exemplo o *Netflix Prize*[14], em 2009, cuja equipe ganhadora obteve o prêmio de 1 milhão de dólares por criar um algoritmo combinado que melhorou substancialmente a precisão das previsões sobre o quanto um usuário iria gostar de um filme com base nas preferências do filme. Os principais tipos de algoritmos conjuntos diferenciam-se de acordo com a estratégia escolhida para fazer uso do resultado da execução dos algoritmos que os compõem, como é descrito a seguir.

2.4.1 Bagging

[13] Disponível em: < http://scikit-learn.org/stable/modules/ensemble.html>. Acesso em: 23 Abr. 2018.

[14] Disponível em: < https://www.netflixprize.com/.> Acesso em: 23 Abr. 2018.

O nome Bagging consiste na abreviatura de *Bootstrap Aggregating* (Breiman, 1996), onde *Aggregating* é um termo em inglês que significa agregação em tradução livre para o português e *Bootstrap* remete a um poderoso método estatístico que tem por objetivo estimar uma quantidade de uma amostra de dados[15].

Esta estratégia consiste em reduzir a variação de do resultado de um algoritmo através do calculo da média dos resultados de vários algoritmos, em casos cujo objetivo é a classificação, ou os resultados com maior ocorrência, para casos de regressão. Um exemplo do uso desta estratégia é o algoritmo de aprendizado supervisionado denominado Florestas Aleatórias que faz uso de vários algoritmos de árvores de decisão que combinados possibilitam obter uma predição com maior acurácia e mais estável[16]. Esse algoritmo, por exemplo, funcionaria da seguinte maneira: caso fossem utilizadas 5 árvores de decisão agregadas com objetivo de realizar a predição de classe para uma amostra de entrada: azul, azul, vermelho, azul e vermelho, o resultado seria a classe mais frequente e portanto a previsão de classe retornada pelo algoritmo seria azul.

2.4.2 Boosting

Boosting é um termo em inglês que significa impulsionando, em tradução livre para o português. Este tipo de conjunto de algoritmos refere-se a uma família de algoritmos capazes de converter algoritmos de aprendizado de máquina considerados “fracos” em algoritmos “fortes” (ZHOU, 2012). São considerados algoritmos “fracos” aqueles que são apenas ligeiramente melhores do que adivinhações aleatórias, como pequenas árvores de decisão, por exemplo. Por algoritmos fortes, entendem-se aqueles que possuem um resultado considerado como quase perfeitos.

A diferença da estratégia utilizada neste algoritmo em comparação com o algoritmo anterior (*Bagging*) é que o seu resultado final é composto pelos resultados ponderados de cada algoritmo que o compõem, ao invés do resultado mais frequente ou a média dos resultados

[15] Disponível em: https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/. Acesso em: 15/10/2018 às 22:53.

[16] Disponível em: https://medium.com/machina-sapiens/o-algoritmo-da-floresta-aleat%C3%B3ria-3545f6babdf8. Acesso em: 15/10/2018 às 23:10.

obtidos. Desta forma cada algoritmo executado determina os recursos nos quais o próximo algoritmo se concentrará e assim, mesmo algoritmos que tenderiam a proporcionar “fracos” resultados, caso executados isoladamente, poderão obter resultados melhores ao serem “impulsionados” por receberem como dados de entrada o resultado da execução de um algoritmo considerado mais “forte”.

Um exemplo de algoritmo que utiliza essa estratégia de *Boosting* é o AdaBoost. Este algoritmo de aprendizado por reforço se concentra em problemas de classificação e tem por objetivo converter um conjunto de classificadores “fracos” em um “forte”. O funcionamento deste algoritmo consiste em, de forma iterativa, escolher o conjunto de treinamento para um algoritmo de classificação com base na precisão do resultado obtido em treinamento de um algoritmo anterior. Para tal ele atribui um peso para cada classificador treinado em qualquer iteração dependendo da precisão obtida. Deste modo um classificador com 50% de precisão receberá um peso de zero, um classificador com menos de 50% de precisão recebe peso negativo e caso obtenha mais que 50% de precisão recebe peso positivo.[17] Com isto os coeficientes de ponderação serão aumentados para pontos de dados que são classificados incorretamente e diminuídos para pontos de dados que são classificados corretamente. O AdaBoost, portanto, trabalha na fase de treinamento, verificando que caso algum algoritmo “falhou” durante a classificação este deveria receber maior atenção (aumentando seu peso) porque “cuida de casos especiais de classificação” que podem ser importantes na formação de um classificador mais eficiente.

2.4.3 Stacking

Stacking é um termo em inglês que significa empilhamento, em tradução livre para o português. A estratégia utilizada neste algoritmo consiste em gerar um algoritmo que é treinado a partir do resultado combinado de dois (ou mais) algoritmos anteriores. Os algoritmos executados inicialmente são treinados com base em um conjunto de treinamento completo, em seguida, um algoritmo combinador é treinado tendo o resultado das saídas dos algoritmos inicial como dados de entrada. Esses algoritmos utilizados na fase inicial são geralmente heterogêneos, embora mesmo não sendo algo frequente é possível a utilização de algoritmos homogêneos (ZHOU, 2012).

[17] Disponível em: https://www.kdnuggets.com/2016/02/ensemble-methods-techniques-produce-improved-machine-learning.html. Acesso em: 16 Out. 2018.

De certo modo, o empilhamento é uma estrutura geral a qual pode ser compreendida como uma generalização de vários métodos de conjunto. Por outro lado, esta estratégia pode ser vista como um método de combinação de algoritmos de aprendizado para obtenção de um resultado específico. É importante estar ciente, entretanto, que realizar um empilhamento de algoritmos homogêneos e/ou heterogêneos, não significa obrigatoriamente que será obtido um algoritmo combinado com desempenho melhor, embora diversos testes realizados identificaram que o uso de um empilhamento de algoritmos atinge maior precisão do que algoritmos individuais, e que baseado em curvas de aprendizado, este não mostra sinais de *overfitting*, o qual é um termo usado para descrever quando um algoritmo se ajusta muito bem a um determinado conjunto de dados anteriormente observado, entretanto se mostra ineficaz ao tentar prever novos resultados[18].

Esta estratégia de conjunto de algoritmos, S*tacking*, foi a escolhida pelo autor para implementação do algoritmo computação neste trabalho. Esta escolha foi motivada pela necessidade da utilização sequencial de algoritmos para alcançar o resultado desejado que consiste em obter um conjunto de candidatos mais similares a uma determinada vaga de emprego. O descarte da possibilidade de utilização das outras duas estratégias citadas (*Bagging* e *Boosting*) deve-se a sua utilidade como

A seguir é descrita a linguagem de programação escolhida para implementação dos algoritmos de aprendizado de máquina que irão compor o algoritmo computacional proposto.

2.5 LINGUAGEM PYTHON

Criada por Guido van Rossum em 1991. É uma linguagem de programação cujos principais objetivos são: produtividade e legibilidade, ou seja, essa linguagem foi criada para produzir código de fácil entendimento e manutenção rápida[19].

A linguagem Python[20], entre as muitas vantagens do uso desta linguagem destacam-se as seguintes:

- Baixo uso de caracteres especiais, tornando-a uma linguagem muito parecida com um pseudocódigo executável;

[18] Disponível em: https://estatsite.com/2016/08/18/overfitting-e-cross-validation/. Acesso em: 16 Out. 2018.

[19] Disponível em: <http://pyscience-brasil.wikidot.com/python:python-oq-e-pq>. Acesso em: 23 Abr. 2018.

[20] Disponível em: <https://www.python.org/.> Acesso em: 23 Abr. 2018.

- Possibilidade do uso de endentação para marcar blocos;
- Quase inexistência de palavras chave voltadas a compilação;
- Possui coletor de lixo que gerencia automaticamente o uso da memória.
- Possui estruturas de dados complexas, como tuplas, listas e dicionários, que facilitam o desenvolvimento de algoritmos complexos.

Foi por estas e outras razões que o autor realizou a escolha desta linguagem para implementação dos algoritmos de aprendizado de máquina necessário para pleno funcionamento da aplicação conceituada. Além disso a linguagem possui diversas bibliotecas de código como a scikit-learn[21] que é uma biblioteca de aprendizado de máquina de código aberto desenvolvida por terceiros e distribuída separadamente. Essa biblioteca foi projetada para interagir com as bibliotecas Python numéricas e científicas, como a NumPy, por exemplo, que suporta *arrays* e matrizes multidimensionais, bem como, também possui maior desempenho para realizar operações matemáticas que demandam mais processamento, algo extremamente necessário para a aplicação desenvolvida pelo autor. No capítulo a seguir será melhor conceituada o algoritmo computacional que será implementado bem como as etapas necessárias para seu desenvolvimento.

[21] Disponível em: <http://scikit-learn.org/>. Acesso em: 23 Abr. 2018.

3 DESENVOLVIMENTO E ANÁLISE DE RESULTADOS

Neste capítulo serão descritas as etapas de construção do conjunto de dados de currículos de candidatos e da implementação do conjunto de algoritmos de inteligência artificial que irão compor o algoritmo computacional a qual já foi iniciado sua conceitualização no Capítulo 2 e que será descrito de maneira mais extensiva adiante. O conjunto de dados de currículos de candidatos foi construído através do preenchimento de um currículo online disponível na área "Trabalhe Conosco" do site da empresa Sol Saúde on Life. Os campos deste currículo que foram preenchidos pelos candidatos neste formulário estão descritos no quadro 4.

Quadro 4: Campos do currículo online da Sol Saúde on Life

Nome do Campo	Obrigatório?
Nome	Sim
Idade	Sim
Sexo	Sim
Telefone	Sim
Email	Não
Endereço Completo	Sim
Estado Civil	Sim
Formação Acadêmica	Sim
Curso Superior	Sim *
Turno	Sim *
Previsão de Conclusão	Sim *
Cursos Extracurriculares	Sim *
Experiência Profissional	Sim *
Trabalhos Voluntários	Não
Projetos de Extensão	Não
Disponibilidade de Horário	Sim
Trabalhando Atualmente?	Sim
Disponibilidade para Viagens?	Sim
Meio de Transporte	Sim
Possui Deficiência?	Sim
Tipo da Deficiência	Sim *
Cargo de Interesse	Não

Fonte: Do Autor.

Os campos que constam com * na coluna "Obrigatório?" do quadro 4, referem-se aos campos que não são obrigatórios para todos os candidatos, mas que podem vir a ser obrigatórios dependendo da informação assinalada em algum dos campos anteriores a ele. Por

exemplo: os campos “Tipo de Deficiência” terá o preenchimento obrigatório apenas para os candidatos que assinalaram no campo “Possui Deficiência” como afirmativo. Há campos, como “Experiência Profissional” ou “Cursos Extra Curriculares”, por exemplo, que são multivalorados, ou seja, o formulário permite que o candidato preencha mais de uma informação com objetivo que ele descreva de maneira mais detalhada suas competências. Como citado na sessão de Metodologia, os dados captados após o preenchimento deste formulário são armazenados em um banco de dados relacional.

Após os dados dos currículos dos candidatos serem salvos no banco de dados o algoritmo computacional proposto realiza sua primeira fase de execução. Nesta etapa são recuperados do banco de dados as informações dos currículos de candidatos para realização de um pré-processamento, que consiste em seleção de instâncias, normalização de valores e tratamento de campos com valores ausentes, além de convertê-los da representação compreendida por seres humanos para uma representação numérica a qual é compreendida pelos algoritmos de IA.

3.1 PRÉ-PROCESSAMENTO DOS DADOS

A fase de processamento dos dados que serão utilizados como entrada para o algoritmo computacional proposto inicia com a seleção de instancias, mais precisamente colunas da tabela que contém os dados dos currículos dos candidatos, que possuem maior “significância” para a realização da segmentação dos candidatos em grupos e posterior cálculo de suas similaridades com uma vaga de emprego. Em conversa com o representante do setor de RH da empresa Sol Saúde on Life, foi identificado que dentre as 22 variáveis presentes no currículo dos candidatos, as que possuíam maior representatividade e que são utilizadas pelo setor como delineadoras da escolha dos candidatos a serem recrutados para os processos seletivos das vagas disponibilizadas pela empresa são 9: Cidade em que reside, Formação Acadêmica, Curso Superior, Turno, Previsão de Conclusão, Cursos Extra Curriculares, Experiência Profissional, Disponibilidade de Horário e Cargo de Interesse.

Ciente desta informação foi realizada a segunda etapa do processamento de dados, exclusivamente sobre as 9 variáveis selecionadas na etapa anterior. O procedimento executado consistia em efetuar a normalização dessas variáveis para se adequarem a uma escala conveniente. Por exemplo: a variável Previsão de conclusão de curso superior foi adequada ao intervalo igual ou superior a data vigente e inferior a 6 anos posteriores a esta mesma data. Logo em seguida, foi realizada a mais uma etapa de pré-processamento

caracterizada pela identificação de valores ausentes, geralmente não obrigatórios, que foram substituídos pelo valor 0, com objetivo de não deixar lacunas que viessem a prejudicar a plena execução dos algoritmos de aprendizado de máquina. Por fim, a última etapa de pré-processamento executada foi a substituição do conteúdo destas 9 variáveis por valores numéricos positivos, de acordo com um dicionário de dados desenvolvido para facilitar o entendimento do significado desses valores em relação ao conteúdo a qual foi realizada a substituição.

A realização desta última etapa justifica-se pela necessidade de adequar as variáveis, que serão utilizadas como entrada para o algoritmo, em um formato compreensível pelo mesmo, que foi implementado apenas para encontrar similaridade entre valores numéricos, e que não poderia realizar sua execução de maneira adequada e satisfatória caso lhe fossem fornecidos como dados de entrada um conjunto de letras e/ou caracteres especiais. Um exemplo do funcionamento desta etapa é ilustrado no quadro 5.

Quadro 5: Exemplo de dicionário de dados

Chave do Atributo	Formação Acadêmica
0	Analfabeto
1	Ensino Fundamental Incompleto
2	Ensino Fundamental Completo
3	Ensino Médio/Técnico Incompleto
4	Ensino Médio/Técnico Completo
5	Ensino Superior Incompleto
6	Ensino Superior Completo
7	Pós Graduação
8	Mestrado
9	Doutorado
10	PhD

Fonte: Do Autor.

O quadro 5 demonstrar o uso da estratégia de criação de um dicionário de dados. Este dicionário foi criado tendo por objetivo a representação da variável “Formação Acadêmica” do currículo online anteriormente citado. A coluna “Chave do Atributo” é responsável por conter uma representação numérica para cada grau de escolaridade descrito na coluna “Formação Acadêmica”. Desta forma o dado inserido como entrada para o algoritmo será, convenientemente, o que consta na “Chave do Atributo”, de modo a evitar aumento

desnecessário de complexidade para o algoritmo e suas respectivas operações de cálculo de similaridades. Após esta etapa de pré-processamento de dados, foi executado o algoritmo de agrupamento para segmentar os diferentes perfis de currículos de candidatos de acordo com a relação de similaridade entre eles.

3.2 AGRUPAMENTO DE CANDIDATOS

Para realização desta etapa de agrupamento de candidatos, foi utilizado, dentre os algoritmos de agrupamento disponíveis, o *K-means*, cujas razões para escolha do mesmo já foram descritas no Capítulo 2. A implementação deste algoritmo foi obtida da biblioteca de aprendizado de máquina de código aberto para a linguagem de programação Python chamada *Scikit Learn* em sua versão 0.21. Este algoritmo pode receber até 11 parâmetros (*n_clusters, init, n_init, max_iter, tol, precompute_distances, verbose, random_stat, copy_x, n_jobs e algorithm*), sendo obrigatório o fornecimento de apenas 01 destes que é o *n_clusters*, o qual representa a quantidade de grupos que deseja ser obtida ao fim da execução do algoritmo.

Entretanto, mesmo possuindo apenas um parâmetro obrigatório, foi fornecido outro parâmetro (*init*) para execução deste algoritmo. O parâmetro *init* possui três valores (ou conjunto de valores) que são aceitos, os quais são: *K-means++*, *random* ou *ndarray*. O parâmetro *K-means++* foi o escolhido por ser uma estratégia que seleciona os centros de grupos iniciais de uma maneira inteligente com objetivo de acelerar a convergência para uma solução aceitável e com menor duração de execução. As outras duas possibilidades *random* ou *ndarray*, respectivamente, consistem em permitir que o algoritmo defina valores aleatórios para os centros de grupos iniciais ou que os mesmos sejam fornecidos em um *array* (conjunto de valores) ao algoritmo.

Para definição do valor a ser fornecido para variável *n-clusters* foi utilizado o "Método do Cotovelo", descrito no Capítulo 2. Para tal foi executado o algoritmo K-means fornecendo como valor para variável *n-clusters* números de 1 a 9 (que é a quantidade de variáveis selecionadas dos currículos dos candidatos). A quantidade de perfis de currículos de candidatos obtidos do banco de dados da empresa Sol Saúde on Life foi 104, dos quais foram selecionadas as 9 variáveis, consideradas como mais importantes, de cada um destes perfis,. Após as 9 execuções do algoritmo foi gerado um gráfico contendo a porcentagem de variância mínima obtida em relação ao número de grupos formados. Este gráfico é ilustrado na figura 12.

Figura 12 – Gráfico contendo a variância mínima obtida após execução do algoritmo K-means

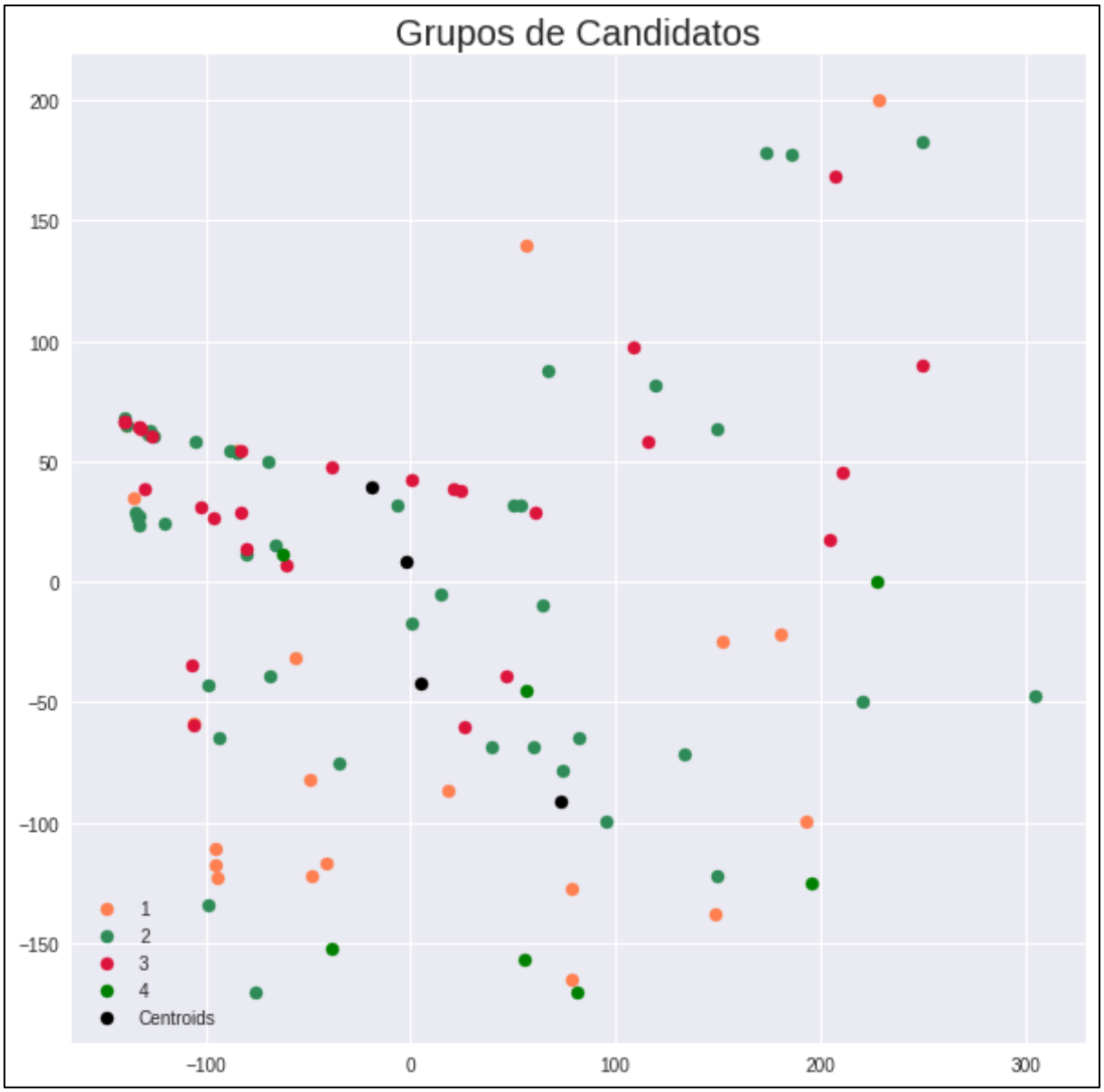
Fonte: Do Autor.

Com base na figura 12 é possível perceber que o valor mais indicado para ser utilizado no parâmetro *n_clusters* é 4, pois a partir dele a linha pontilhada do gráfico torna-se quase horizontal, ou seja, praticamente não há mais ganho de informação adicional. Após esta constatação foi realizada a execução do algoritmo *K-means* fornecendo as variáveis e "4" e "K-means++", para os parâmetros *n_clusters* e *init*, respectivamente. O resultado da execução do algoritmo é possível ser obtido através de seu atributo "*cluster_centers_*", que contém os respectivos centros dos grupos criados pelo algoritmo, e pelo atributo "*labels_*" que contém o conjunto de rótulos que foram atribuídos pelo algoritmo aos dados de entrada (perfis de currículos de candidatos) que lhe foi fornecido.

A quantidade de grupos gerados pelo algoritmo pode ser obtida utilizando dois métodos nativos da linguagem Python chamados *len* e *set*. O método *len* (abreviatura da palavra *length* do idioma inglês e que significa "comprimento" em português) retorna o número de itens contidos em um objeto que lhe foi passado como parâmetro. Já o método *set* (que significa "conjunto" em português) retorna um conjunto composto pelos elementos contidos em um objeto que lhe foi passo como parâmetro, sendo que cada elemento pode entrar no conjunto apenas uma vez. Executando, portanto, esses dois métodos em sequência e

passando o atributo "*labels__*" como parâmetro (desta maneira: len(set(labels__)), foi obtido o valor 4 que equivale a quantidade de grupos formados pelo algoritmo *K-means*. Para facilitar a identificação visual destes 4 grupos, foi gerado o gráfico de dispersão ilustrado na figura 13.

Figura 13 – Gráfico de dispersão com grupos resultantes da execução do algoritmo K-means



Fonte: Do Autor.

A figura 13 demonstra a distribuição heterogênea dos perfis de candidatos em grupos cujos participantes são mais similares entre si. As cores têm por objetivo facilitar a identificação da segmentação existente entre os pontos (perfis). Os pontos pretos representam o centro de cada um dos grupos, ou seja, a média dos atributos de todos os perfis de candidatos contidos naquele grupo e que foram obtidos através do atributo "*cluster_centers_*" resultante da execução do algoritmo *K-means*. Os dados destes centros dos grupos constitui

uma informação muito importante a qual é utilizada na próxima fase do algoritmo computacional com objetivo de identificar o grupo cujo centro é mais semelhante a uma vaga de emprego disponibilizada, evitando que seja realizada a comparação entre esta vaga e todos os 101 candidatos existentes no banco de dados da empresa.

3.3 CLASSIFICAÇÃO DE UMA VAGA DE EMPREGO

Concluído o agrupamento dos perfis de candidatos, iniciou-se o procedimento de identificação dos candidatos mais similares a uma determinada vaga de emprego. As características da vaga, elencadas pelo profissional de RH estão descritas no quadro 6.

Quadro 6: Características da vaga de emprego disponibilizada

Cargo: TÉCNICA DE ENFERMAGEM	
Característica	Resposta desejada
Formação Acadêmica	Ensino Técnico
Curso	Técnico de Enfermagem.
Turno	Não se aplica.
Previsão de Conclusão	Não se aplica.
Cidade que reside	João Pessoa
Cursos extracurriculares	Cuidados paliativos, Administração de medicamentos, Feridas e curativos, Saúde do idoso, Nutrição e dietética para enfermagem, Saúde mental, Cuidados com pacientes entubados.
Experiência Profissional	Técnica de enfermagem, Atendente domiciliar e hospitalar, Acompanhante de idosos, Tratamento de feridas, Home care.
Disponibilidade de Horário	12/24 Horas
Cargo de Interesse	Técnico de Enfermagem.

Fonte: Do Autor.

Com objetivo que estas características da vaga pudessem ser inseridas como dados de entrada para o algoritmo *K-means* foi realizado o pré-processamento do conteúdo destas características convertendo-as em valores numéricos, de acordo com o dicionário de dados desenvolvido. O quadro 7 descreve as características dessa vaga após o pré-processamento realizado.

Quadro 7: Características da vaga de emprego disponibilizada após pré-processamento

Cargo: TÉCNICA DE ENFERMAGEM	
Característica	Resposta desejada
Formação Acadêmica	5
Curso	2
Turno	0
Previsão de Conclusão	0
Cidade que reside	1
Cursos Extracurriculares	155, 102, 104, 198, 200, 49, 172
Experiência Profissional	93, 113, 116, 64, 65
Disponibilidade de Horário	4
Cargo de Interesse	9

Fonte: Do Autor.

Para realizar a classificação desta vaga em relação a algum dos grupos de candidatos existentes foi utilizado o método "*Predict*" do algoritmo *K-means*, fornecendo como parâmetro os dados pré-processados da vaga. Este método realiza o cálculo da distância entre um conjunto de dados fornecido e os centros dos grupos a qual o algoritmo realizou o agrupamento anteriormente, tendo como retorno o índice do grupo cujo centro possui menor distância em relação ao conjunto de dados fornecido. Realizado este procedimento foi verificado que o grupo cujo centro possui maior similaridade (menor distância) em relação à vaga disponibilizada é o de índice 1, o qual contém 44 perfis de candidatos. Diante disso, foi realizada a última etapa do algoritmo computacional que consiste em verificar, dentre os 44 perfis de candidatos contidos no grupo selecionado, quais possuem maior similaridade em relação à vaga de emprego disponibilizada.

3.4 ÍNDICE DE SIMILARIDADE VAGA-CANDIDATO

Com objetivo de obter um resultado mais preciso e coerente quanto à definição dos candidatos mais similares a vaga citada, foi definido, em conjunto com o profissional de RH, que seriam atribuídos "pesos" as características da vaga disponibilizada de modo a evitar que

um determinado perfil de candidato fosse considerado como mais similar a vaga apenas por possui maior quantidade de características semelhantes às requeridas pela mesma. Esses valores de “pesos” foram estabelecidos como variando no intervalo entre 0 e 1 onde, quanto mais o valor estiver próximo de 0 menos importante seria aquela característica e, quanto mais próximo de 1 maior importância seria atribuída a mesma. Definidos os “pesos” a serem atribuídos as características da vaga foi produzido o quadro 8, contendo essas características e seus respectivos “pesos”.

Quadro 8: Características da vaga de emprego após atribuição de pesos

Cargo: TÉCNICA DE ENFERMAGEM		
Característica	Resposta desejada	Peso
Formação Acadêmica	5	1
Curso	2	1
Turno	0	1
Previsão de Conclusão	0	1
Cidade que reside	1	1
Cursos Extracurriculares	155, 102, 104, 198, 200, 49, 172	0,4
Experiência Profissional	93, 113, 116, 64, 65	0,6
Disponibilidade de Horário	4	1
Cargo de Interesse	9	0,7

Fonte: Do Autor.

Para identificação dos perfis de candidatos mais semelhantes à vaga disponibilizada foram estudadas diversas fórmulas existentes na literatura, tanto relacionadas ao cálculo de distâncias, como a Distância Euclidiana, Distância de Manhattan, Distância de Minkowski, entre outras, quanto relacionadas ao cálculo de índices de similaridade, como o Coeficiente de Jaccard, Coeficiente de Sorensen e Similaridade de Cossenos. Entretanto para que o algoritmo pudesse realizar estimativas da similaridade existente entre o par vaga-candidato de maneira mais semelhante às escolhas realizadas pelo profissional de RH, foram definidas pelo autor deste trabalho 3 premissas, entretanto estas não puderam serem satisfeitas pelas fórmulas estudadas. São elas:

1. O cálculo de similaridade deve resultar em um valor percentual;
2. O cálculo de similaridade deve contemplar a possibilidade de que o perfil do candidato seja superior às características exigidas para vaga;
3. O cálculo de similaridade apenas deve resultar em um percentual de 100% de similaridade caso o perfil do candidato seja exatamente igual às características exigidas para vaga.

Algumas fórmulas estudadas, como a Distância Euclidiana, por exemplo, produzem como resultado um valor de distância que varia de zero a infinito, o que viola a premissa 1, referente a necessidade que o valor retornado seja em percentual. Outras fórmulas, como o Coeficiente de Jaccard, por exemplo, produzem um resultado percentual, porém apenas comparando a quantidade de características que o perfil do candidato possui que são iguais ao requerido pela vaga dividido pelo número de características totais da vaga, ou seja, não contempla a possibilidade de que o candidato possa ter um perfil superior a vaga disponibilizada, o que viola a premissa 2.

Em algumas das outras formulas estudadas, como o Índice de Bray-Curtis, o qual consiste no módulo da soma da diferença entre as características de um par de exemplos (vaga-candidato) dividido pelo módulo da soma dessas características, foi verificado que embora essa fórmula produza um resultado percentual e contemple a possibilidade de o perfil do candidato seja superior a vaga ela viola a premissa 3 no tocante a não garantir que um candidato seja considerado como tendo 100% de similaridade com a vaga apenas se todas as suas características sejam idênticas as requeridas pela vaga.

Tal violação ocorre quando um candidato, por exemplo, possui algumas características inferiores as requeridas pela vaga e, simultaneamente, também possui características que sejam superiores as exigidas pela mesma, o que vem a provocar uma possível "compensação" entre as características "inferiores" e "superiores" do candidato em relação aos requisitos da vaga, tornando-o em algumas situações aparentemente possuidor de maior similaridade com a vaga em detrimento a um candidato que possua todas as características exigidas pela mesma vaga.

Diante disso foram criadas 4 fórmulas que pudessem, em conjunto, garantirem a não violação das 3 premissas definidas anteriormente. Estas fórmulas são descritas a seguir:

$$S = \sum_{i=0}^{n} \frac{(k * w) - (e + d) + d}{(q * w)} \tag{3.1}$$

$$S = \sum_{i=0}^{n} \frac{(k * w)}{(q * w)} \tag{3.2}$$

$$e = \sum_{i=0}^{n} (k - q) \quad \text{Se positivo;} \tag{3.3}$$

$$d = \sum_{i=0}^{n} (k - q) \quad \text{se negativo;} \tag{3.4}$$

onde:

S = Índice de Similaridade entre o par vaga-candidato

k = Característica do Candidato

q = Característica da Vaga

w = Peso atribuído a uma característica

e = Excedente da característica do candidato em relação ao requerido pela vaga

d = Deficit da característica do candidato em relação ao requerido pela vaga

A sequencia de utilização destas 4 fórmulas pelo algoritmo ocorre da seguinte forma. Inicialmente o algoritmo calcula a diferença entre cada uma das características do candidato em relação a vaga, caso essa diferença seja negativa seu valor é incrementado a variável Deficit (*d*) que ao iniciar o algoritmo possui valor 0. Durante esta etapa, caso verificado que o valor da diferença é positivo, seu valor é incrementado a variável Excedente (*e*) que também possui valor 0 ao início do algoritmo. Caso o valor da diferença seja igual a 0 o algoritmo não realiza nenhuma ação para seu armazenamento. Descobertos os valores para as variáveis Deficit e Excedente, o algoritmo verifica se o valor da variável Deficit é maior que zero, caso sim é executada a etapa seguinte do algoritmo utilizando a fórmula 3.1. Caso não, a fórmula utilizada é a 3.2. Essa verificação se faz necessária pois a fórmula 3.1 é utilizada com objetivo de garantir a não violação da premissa 3 em casos onde poderiam ocorrer as "compensações" citadas anteriormente.

A seguir é demonstrado o funcionamento do algoritmo do algoritmo, utilizando as 4 fórmulas citadas, para uma vaga e conjunto de candidatos fictícios. O quadro 9 descreve as características da vaga fictícia, já com o pré-processamento realizado e com seus respectivos pesos definidos.

Quadro 9: Características de uma vaga fictícia para exemplo

Cargo: VENDEDOR			
Característica	Resposta	Peso	Resposta após aplicação do peso
Cidade que reside	1	1	1
Formação Acadêmica	5	1	5
Curso Superior	3	1	3
Cursos	1	0,4	0,4
Experiência	2	0,6	1,2
Disponibilidade de	2	1	2

Cargo de Interesse	7	0,7	4,9

Fonte: Do Autor.

O quadro 10 descreve as características de um conjunto de candidatos fictícios, já com o pré-processamento realizado.

Quadro 10: Características do currículo fictício de candidatos para exemplo

Nome	Cidade	Formação	Curso	Cursos Extra	Experiências	Cargo	Disponibilidade
João	1	1	4	1	2	2	1
José	0	2	3	4	3	4	3
Maria	1	4	6	3	1	3	2
Paulo	1	5	1	2	4	1	3
Lucas	1	3	2	5	2	5	1
Tiago	0	2	4	1	3	4	2
Felipe	1	0	1	2	1	1	3

Fonte: Do Autor.

A aplicação das 4 fórmulas pelo algoritmo sobre as características dos candidatos ocorre da seguinte maneira. Tendo como exemplo o currículo fictício do candidato Paulo, sobre o qual foram aplicados os mesmo pesos definidos para vaga, a tabela 1 descreve a realização da etapa inicial do uso das formulas citadas consistindo no cálculo das diferenças entre os valores das características contidas no currículo do candidato em relação as exigidas para vaga.

Tabela 1: Calculo da diferença entre as características do par Vaga-Candidato

Característica	Candidato	Vaga	Diferença	Excedente ou Deficit?
Cidade que reside	1	1	0	-
Formação Acadêmica	5	5	0	-
Curso Superior	1	3	-2	Deficit
Cursos Extracurriculares	0,8	0,4	0,4	Excedente
Experiência Profissional	2,4	1,2	1,2	Excedente
Disponibilidade de Horário	1	2	-1	Deficit
Cargo Desejado	2,1	4,9	-2,8	Deficit

Fonte: Do Autor.

Obtidos os valores resultantes das diferenças existentes entre as características da vaga e do candidato foram aplicadas as fórmulas 3.3 e 3.4 sobre esses valores, como descrito a seguir:

$e = 0,4 + 1,2 = 1,6$

$d = -2 + (-1) + (-2,8) = -5,8$

Portanto foi verificado que depois de aplicadas as fórmulas 3.3 e 3.4 sobre as características do currículo fictício do Paulo foram encontrados os valores para as variáveis Excedente (e) e Deficit (d) e verificado que o valor da variável Deficit é negativo, ou seja, a

fórmula que será utilizada, sobre os mesmos dados da vaga e do candidato utilizados no cálculo anterior, é a 3.1. Este procedimento é descrito a seguir:

$$S = \frac{(1+5+1+0{,}8+2{,}4+1+2{,}1)-(1{,}6+(-5{,}8))+(-5{,}8)}{(1+5+3+0{,}4+1{,}2+2+4{,}9)}$$

$$S = \frac{13{,}3+4{,}2-7{,}8}{(1+5+3+0{,}4+1{,}2+2+4{,}9)}$$

$$S = \frac{9{,}7}{17{,}5} = 0{,}5542 \text{ ou } 55{,}42\%$$

Após a execução desta etapa do algoritmo, utilizando os dados do currículo fictício do Paulo e a vaga fictícia de vendedor criada, foi verificado que o currículo deste candidato seria 66,80% similar a esta vaga. Esta última etapa do algoritmo foi executada então com os dados dos 44 candidatos que compõem o grupo de índice 1 conforme resultado da etapa anterior do algoritmo. Para obter um resultado mais coerente, em conversa com o profissional de RH, foi definido pelo mesmo que apenas deveriam ser retornados pelo algoritmo os candidatos que possuíssem um índice de similaridade superior dentro do intervalo de 70% a 100% em relação à vaga de Técnico de Enfermagem disponibilizada. O resultado da execução desta etapa final do algoritmo, realizada em 0,365 segundos, foi de 29 candidatos que, segundo o algoritmo, poderiam ser recrutados pelo profissional de RH para a vaga disponibilizada.

3.5 VALIDAÇÃO DO ALGORITMO E ANÁLISE DE RESULTADOS

De posse desse ranking de candidatos, em ordem decrescente, obtido como resultado da execução do algoritmo computacional desenvolvido, o mesmo foi apresentado ao profissional de RH da empresa para sua validação. E, de acordo com este profissional, o algoritmo escolheu 3 candidatos que o mesmo não teria selecionado. A razão do não recrutamento de dois destes candidatos foi devido aos mesmos possuírem ensino superior completo, o que viriam a onerar a folha salarial da empresa (um dos motivos pelos quais a vaga requisitava que o candidato possuísse ensino técnico). Quanto ao outro candidato selecionado pelo algoritmo, mas que o profissional de RH não recrutaria, a justificativa do profissional foi relacionada ao fato do candidato, embora possuidor de cursos na área de saúde, como o curso de Urgência e Emergência em Trauma Hospitalar, por exemplo, não ter realizado o curso de Técnico de Enfermagem, o qual é o um dos requisitos de maior importância para vaga.

Diante disso, o profissional de RH da empresa Sol Saúde on Life, informou ter concordado totalmente com 26 das 29 escolhas do algoritmo e discordado totalmente de 3 destas escolhas relacionadas ao recrutamento de candidatos para vaga de Técnico de Enfermagem disponibilizada pela empresa. Portanto, pode ser atribuída ao algoritmo computacional desenvolvido uma assertividade de 89,65% na realização da mesma atividade (recrutamento de candidatos) que seria feita pelo profissional humano.

4 CONSIDERAÇÕES FINAIS

O desenvolvimento deste trabalho foi muito desafiador e gratificante para o autor, pois para sua realização o mesmo teve que pesquisar diversos temas e nichos de conhecimento que não foram estudados durante a sua graduação ou que não pertenciam ao escopo de sua área de interesse de pesquisa, mas que demonstraram serem temas muito relevantes e que acrescentaram bastante conhecimento e experiência a ele.

Para realização deste estudo foram definidos alguns objetivos os quais foram todos alcançados em sua totalidade, pois o autor realizou a pesquisa sobre diversos algoritmos de aprendizado de máquina, tanto de aprendizado supervisionado, não supervisionado quanto de aprendizado por reforço e realizada a escolha de um destes algoritmos para sua utilização neste trabalho. Também foram estudadas as três principais estratégias de utilização de métodos em conjunto existentes na literatura e escolhida a que mostrou ser mais apropriada ao contexto do objetivo a ser alcançado com a implementação do algoritmo proposto. Foi criado um conjunto de dados, obtido da área trabalhe conosco do site da empresa Sol Saúde on Life, contendo dados reais de currículos de candidatos a uma vaga de emprego disponibilizada pela mesma.

Através da execução do algoritmo proposto foram criados grupos de candidatos com objetivo de reduzir a dimensionalidade do conjunto de dados criado e evitar comparações desnecessárias entre currículos com perfil totalmente incompatíveis com a vaga disponibilizada. Após isso foi obtido um ranking de candidatos, em ordem decrescente, com base nos respectivos índices de similaridade obtidos após comparação feita pelo algoritmo entre a vaga disponibilizada e os candidatos com perfil similar a mesma. A validação do trabalho foi obtida mediante submissão do resultado da execução do algoritmo desenvolvido ao profissional de RH da empresa que disponibilizou a vaga de emprego para sua subsequente verificação de concordância ou discordância quanto aos candidatos selecionados pelo algoritmo.

.Após validação do profissional de RH pode ser considerado que o índice de assertividade do algoritmo que foi de 89,65% justifica sua utilização pela empresa como auxiliar nas decisões de recrutamento de candidatos pelo setor de RH, aumento o tempo disponível dos colaboradores deste setor quanto a realização das atividades que são de sua responsabilidade, mas que por vezes eram realizadas de maneira incompleta devido ao temo gasto analisando individualmente cada currículo enviado de candidatos as vagas de emprego disponibilizadas pela empresa.

Embora o algoritmo desenvolvido tenha alcançado um alto índice de assertividade, pode ser realizado como trabalhos futuros a verificação da razão do mesmo ter considerado o recrutamento de candidatos que superavam ou que eram inferiores ao intervalo dos percentuais de similaridade definidos pelo profissional de RH da empresa para o recrutamento e subsequente participação do processo seletivo da vaga disponibilizada, de modo a melhorar ainda mais o índice de assertividade do mesmo.

REFERÊNCIAS

ALVAREZ, E. B.; SIRIANI, A. L. R.; VIDOTTI, S. A. B. G.; CARVALHO, A. M. G. Os **Sistemas de Recomendação, Arquitetura da Informação e a Encontrabilidade da Informação**. Transinformação, Dez 2016, vol.28, no.3, p.275-286.

APPOLINÁRIO, F. **Metodologia científica**. São Paulo : Cengage, 2016.

BANOV, M. R. **Recrutamento, selecão e competências**. 4. ed. São Paulo: Atlas, 2015.

BREIMAN, L. **Bias, variance, and arcing classifiers.** Technical Report 460, Statistics Department, University of California, Berkeley, CA, 1996a.

CHIAVANETO, I. **Gestão de pessoas: o novo papel dos recursos humanos nas organizações**. 4. ed. Barueri, SP: Manole, 2014.

CHIAVENATO, I. **Recursos humanos: o capital intelectual das organizações**. Rio de Janeiro: Elsevier, 2009.

COPPIN, B. **Inteligência Artificial**. Rio de Janeiro: LTC, 2017.

COSTA, M. T. C. da. **Uma Arquitetura Baseada em Agentes para Suporte ao Ensino à Distância.** Tese (Doutorado em Engenharia de Produção) – Curso de Pós-Graduação em Engenharia de Produção e Sistemas, Universidade Federal de Santa Catarina – UFSC, Florianópolis, 1999.

DANIEL, G. P. **Otimização de algoritmos de agrupamento espacial baseado em densidade aplicados em grandes conjuntos de dados.** Dissertação (mestrado) – Universidade Estadual Paulista Júlio de Mesquita Filho, Instituto de Biociências, Letras e Ciências Exatas. São José do Rio Preto, 2016.

ESTER, M.; SANDER, J.; KRIEGEL, H.; XU, X. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. **Data Mining and Knowledge Discovery**. Berlin: Springer-Verlag, 1998.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. L. F. **Inteligência Artificial: Uma abordagem de aprendizado de máquina**. Rio de Janeiro: LTC, 2011.

FERNANDES, A. M. R. da. **Inteligência Artificial: noções gerais**. 3. imp. Florianópolis: VisualBooks, 2005.

FORD, K.; HAYES, P. IJCAI'95 Proceedings of the 14th international joint conference on Artificial intelligence. **Turing Test Considered Harmful**, Quebec, Canadá, v. 1, p. 972-977, 1995.

KERN, E. **Uma Estrutura de Agentes para o Processo de Licitação**. 1998. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis.

LINDEN, R. Técnicas de Agrupamento: Tutorial. **Revista de Sistemas de Informação da FSMA**, n. 4, p. 18-36, 2009. Disponível em: <http://www.fsma.edu.br/si/sistemas.html>. Acesso em: 23 Abr. 2018.

LUGER, G. F. **Inteligência Artificial**. 6. ed. São Paulo: Pearson Education do Brasil, 2013.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of mathematical biophysics**, vol. 5 (1943), pp. 115–133.

NWANA, M. **Intelligent Agents**: A Technology and Business Application Analysis. 1995. Disponível em: <http://haas.berkeley.edu/~heilmann/agents/.> Acesso em 18 abr. 2018.

OLIVEIRA, J. A influência da área de RH na produtividade das pequenas empresas. **XIII SemeAd – Seminários em Administração**. São Paulo. 2010.

PRODANOV, C. C; FREITAS, E. C. de. **Metodologia do trabalho científico:** métodos e técnicas da pesquisa e do trabalho. 2. ed. – Novo Hamburgo: Feevale, 2013.

Reductionism: Occam's Razor, Reductionism, Monism, Reduction, Type Physicalism, Dialectical Monism, Separation of Concerns. [S.l.]: General Books. 2010. 96 páginas.

ROSA, J. L. G. **Fundamentos da inteligência artificial**. Rio de Janeiro: LTC, 2011.

ROZA, F. S. **Aprendizagem de máquina para apoio à tomada de decisão em vendas do varejo utilizando registros de vendas.** 2016. Relatório submetido à Universidade Federal de Santa Catarina como requisito para a aprovação da disciplina DAS 5511: Projeto de Fim de Curso, Universidade Federal de Santa Catarina – UFSC, Florianópolis.

RUSSELL, S.; NORVIG, P. **Inteligência artificial**. Rio de Janeiro: Elsevier, 2013.

SIMON, H. A. Why should machines learn? In: MICHALSKI, R. S.; CARBONELL, J. G.; MITCHEL, T. M. (Eds). **Machine Learning**. An Artificial Intelligence Approach, v.1. Palo Alto, CA: Tioga, 1983.

MONTEIRO, S. T.; RIBEIRO, C. H. C. Desempenho de algoritmos de aprendizagem por reforço sob condições de ambiguidade sensorial em robótica móvel. **Sba Controle & Automação**. vol.15, n. 3 Campinas, 2004. Disponível em: < http://dx.doi.org/10.1590/S0103-17592004000300008>. Acesso em: 23 Abr. 2018.

VALÊNCIO, C. R. et al. VDBSCAN+: performance optimization based on GPU parallelism. In: **international conference on parallel and distributed computing, applications and technologies (pdcat)**, 2013. p. 23-28.

VAPNIK, V. N. **The nature of statistical learning theory**. New York, NY, USA: Springer-Verlag New York, Inc., 1995. ISBN 0387945598. Disponível em: <http://portal.acm.org/citation.cfm?id=211359>.

WOOLDRIDGE, M.; JENNINGS, N. R. **Intelligent Agents**: Theory and Practice. The Knowledge Engineering Review, v. 10, n. 2, p. 115-152, 1995.

ZHOU, ZHI-HUA. **Ensemble Methods: Foundations And Algorithms.** Taylor & Francis Group, LLC, 2012.