UNIVERSIDADE FEDERAL DE ALAGOAS

INSTITUTO DA COMPUTAÇÃO

CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

DEIVINSON SEVERINO DA SILVA

Aprendizagem de máquina aplicada a detecção de ocupação em salas de escritórios com base em dados de sensores.

Maceió - AL

2019

DEIVINSON SEVERINO DA SILVA

Aprendizagem de máquina aplicada a detecção de ocupação em salas de escritórios com base em dados de sensores.

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Alagoas – UFAL, para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador(a): Professor Lucas Benevides Viana de Amorim.

Maceió - AL
2019

FOLHA DE APROVAÇÃO

DEIVINSON SEVERINO DA SILVA

Aprendizagem de máquina aplicada a detecção de ocupação em salas de escritórios com base em dados de sensores.

Trabalho de Conclusão de Curso submetido ao corpo docente do curso de Bacharelado em Sistemas de Informação da Universidade Federal de Alagoas, aprovado em 09 de outubro de 2019

BANCA EXAMINADORA:

_____

Professor, Me. Lucas Benevides Viana de Amorim
Universidade Federal de Alagoas – Campus A. C. Simões (Orientador)

_____

Professor, Dr. Bruno Almeida Pimentel
Universidade Federal de Alagoas – Campus A. C. Simões (Examinador interno)

_____

Professor, Dr. Evandro de Barros Costa
Universidade Federal de Alagoas – Campus A. C. Simões (Examinador interno)

RESUMO

Muitos esforços tem sido empregado em diferentes linhas de pesquisa com o intuito de se aperfeiçoar os índices de precisão na detecção de ocupação por pessoas em um determinado recinto, entre os estudos relatados na literatura, constam técnicas com o auxílio de dispositivos Wi-Fi, Bluetooth, Sensores Passivos de Infravermelho (PIR), câmeras de videomonitoramento, análise de cluster, entre outras. Algumas pesquisas apontam para a fragilidade dos tradicionais detectores de movimento, uma vez que o emprego em locais onde as pessoas permanecem por muito tempo paradas se torna inviável. A redução do desperdício, economia financeira e a preocupação com o meio ambiente são os maiores motivos para realização de estudos dessa natureza. Existem outras abordagens focadas nesta temática, porém dedicadas aos ambientes residências, todavia esse estudo tem como desafio, a realização de ensaios na expectativa da obtenção de resultados iguais ou superiores aos relatados na literatura, aplicando técnicas de Aprendizagem de Máquina supervisionado a fim de se obter respostas precisas para detecção de ocupação em local de trabalho, mais especificamente em uma sala de escritório, combinando dois elementos, o algoritmo de classificação gerador de árvore de decisão C5.0 (RULEQUEST, 2019) e três conjuntos de dados fornecidos por sensores de luz, umidade, CO_2 e temperatura provenientes de um estudo realizado empregando um tratamento semelhante com diferentes modelos de classificação estatística proposto por (CANDANEDO e FELDHEIM, 2015). No atual estudo, em uma ambientação montada para o cenário 3, que serviu de referência para a análise dos resultados, uma taxa de exatidão de 99,22% foi alcançada segundo a medida estatística Accuracy, no mesmo cenário, um desempenho de 99,49% foi anotado com a métrica F1 Score (empregada como referência). O melhor resultado retornado foi na casa 99,83% no cenário 2 aplicando a equação Precision sobre os valores retornados.

Palavras-chave: Detecção de ocupação, Aprendizagem de máquina, Tarefa de classificação, Árvore de decisão, C5.0, Métricas estatísticas.

ABSTRACT

Many efforts have been made in different lines of research in order to improve occupancy detection accuracy in a given room. Among the studies reported in the literature, there are techniques with the aid of Wi-Fi, Bluetooth devices, Passive Infrared Sensors (PIR), video surveillance cameras, cluster analysis, among others. Some research points to the weakness of traditional motion detectors, as employment in places where people have been standing for a long time becomes unviable. The reduction of waste, financial savings and concern for the environment are the main reasons for conducting studies of this nature. There are other approaches focused on this theme, but dedicated to home environments, however this study has as a challenge the performance of tests in the expectation of obtaining results equal or superior to those reported in the literature, applying supervised Machine Learning techniques in order to obtain Accurate responses to workplace occupancy detection, specifically in an office room, combined two elements, the decision tree generator classification algorithm C5.0 (RULEQUEST, 2019) and three datasets provided by light sensors , humidity, CO2 and temperature from a study conducted using a similar treatment with different statistical classification models proposed by (CANDANEDO and FELDHEIM, 2015). In the current study, in a scenario 3 settings, which served as a reference for the analysis of the results, an accuracy rate of 99,22% was achieved according to the Accuracy statistical measure, in the same scenario, a performance of 99,49% was noted with the F1 Score metric (used as a reference). The best result returned was in the 99,83% box in scenario 2 applying the Precision equation on the returned values.

Keywords: Occupation detection, Machine learning, Classification task, Decision tree, C5.0, Statistical metrics.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIAÇÕES E SIGLAS

AM	Aprendizagem de Máquina
ANN	Rede Neural Artificial
CART	Classification and Regression Trees
DT	Decision Tree
DS	Dataseet (Conjunto de dados)
FP	False Positive
FN	False Negative
GBM	Gradient Boosting Machines
GPL	General Public License (Licença Pública Geral)
KNN	K-Vizinho Mais Próximo
LDA	Linear Discriminant Analysis (Análise Discriminante Linear)
ML	Machine Learning
NB	Bayesiana ingênua
PIR	Passive infrared
RF	Random Forest
SVM	Máquina de Vetores de Suporte
TAN	Rede Bayes ingênua aumentada em árvore
WEKA	Waikato Environment for Knowledge Analysis

SUMÁRIO

1 INTRODUÇÃO

A qualidade de vida das pessoas está diretamente relacionada com o uso da energia elétrica, que é empregada em diferentes atividades socioeconômicas e no desenvolvimento tecnológico, e a sua utilização de forma consciente é importante tanto por motivos financeiros, quanto por motivos ambientais, uma vez que a sua produção depende da exploração de recursos naturais. O impacto da atividade humana sobre o meio ambiente, acompanhado do desenvolvimento econômico, tornou-se expressivo e o crescimento populacional implicou em aumento do consumo, originando problemas ambientais cuja solução tornou-se o grande desafio (PUCRS, 2010).

Os edifícios consomem cerca de 40% da energia total no mundo para fornecer um ambiente interno confortável segundo (D'OCA, HONG e LANGEVIN, 2018), e por causa das crises energéticas, cada vez mais atenção tem sido dada aos edifícios dotados de sistemas que potencializam a eficiência para diminuição do consumo. Devido ao aumento progressivo da demanda de eletricidade e a possibilidade de o sistema energético não atender futuramente todos os setores consumidores, se torna indispensável a otimização no uso da luz elétrica, e nesse contexto torna-se perfeitamente viável economizar energia sem reduzir o conforto, o bem-estar e a segurança na parte interna dos edifícios, porém para atingir esse objetivo, a informação de ocupação de suas dependências é um fator decisivo, segundo (YAN, 2015). A detecção da presença de pessoas dentro de uma sala de escritório, por exemplo, poderia determinar o funcionamento de um sistema de ar-condicionado ou de iluminação.

Existem diversos estudos tentando reduzir o uso de energia em salas de edifícios, muitos deles mostrando que é possível a obtenção de uma economia que representa cerca de 33,33% como em (BROOKS, et al., 2015), passando dos métodos tradicionais de controle até os métodos mais eficientes, mas para o alcance dessa redução faz-se necessário que esses estudos sejam colocados em prática, e que as novas construções se adequem aos novos padrões de eficiência energética. Entre as principais formas possíveis de redução do consumo energético, três são abordadas por Ahmad et al., (2018): “construções realizadas com materiais mais eficientes em termos energéticos, implantação de sistemas energéticos mais eficientes, e ajustes às condições internas em proporção ao número de pessoas em um prédio e seu comportamento”. Essa temática tem sido alvo de discussão, e vários esforços tem sido empregados na tentativa de encontrar uma forma precisa de determinar a ocupação do ambiente interno de um prédio.

A redução do uso de energia nos edifícios é um fator importante, para lidar com a emissão de CO_2 e o aquecimento global, devido ao fato de que o uso de energia nesses lugares representa um terço das emissões de gases de efeito estufa (MIRAKHORLI e DONG, 2016). Os prédios comerciais representam a maior área útil da maioria dos países desenvolvidos e utilizam uma quantidade substancial de energia na prestação de serviços para satisfazer as necessidades de conforto dos ocupantes (TIMILEHIN, 2015), neste sentido a necessidade da adoção de posturas ecologicamente corretas tem impulsionado diversos estudos voltados para essa finalidade, entre eles a detecção de pessoas tanto em locais de trabalho quanto em ambientes residências. Entretanto, reconhecer a presenças de pessoas não é uma tarefa fácil, um dos maiores entraves envolve o uso de câmeras de videomonitoramento, pois com o uso desse equipamento a preservação da privacidade fica comprometida conforme estudo de (PETERSEN, 2016). Dados de Sensor Infravermelho Passivo (PIR), comumente usados para detecção de ocupação, instalados principalmente para operação eficiente de energia da iluminação, serviram de base para a pesquisa de (ANSANAY, 2013), no entanto confiar apenas nos dados do sensor PIR para detecção de ocupação não é muito seguro, pois os sensores não capturam ocupantes imóveis ou fora do campo de visão do sensor, portanto não se aplicaria a uma sala de escritório, onde os usuários passam a maior parte do tempo parados. Uma pesquisa para quantificar a presença de pessoas utilizado interruptor mecânico como sensor de cadeira foi implementada em uma sala de conferências com o objetivo de controlar a ocupação visando melhorar a ventilação do local (TIMILEHIN, 2015). Os métodos acima mencionados, baseados em modelos físicos, necessitam de informações previamente delimitadas sobre as condições físicas de cada sala ou ambiente, o que configura uma grande desvantagem da mesma forma que as abordagens existentes para detecção de ocupação baseada em dados, que precisam de informações prévias para funcionar na prática.

Uma possível solução para os problemas apontados foi testada no estudo de Candanedo e Feldheim (2016) que abordou a aprendizagem de máquina, combinando o uso de sensores com algoritmos de classificação, bons resultados foram obtidos nessa pesquisa, onde uma acurácia de 99,33% foi alcançada. Nesse cenário surge a pergunta: de que forma a aprendizagem de máquina (AM) pode contribuir para melhorar a precisão da detecção de pessoas em uma sala de escritório? Tendo como base esse contexto, o atual estudo objetiva fazer uso da AM em uma experiência na tentativa de superar os resultados propostos na literatura, impulsionado principalmente pelos motivos que a se seguem.

A Minimização do impacto ambiental está entre os principais motivos que impulsionam a realização de estudos dessa natureza, muitos deles preocupados com as elevadas taxas de consumo dos edifícios comerciais que consomem aproximadamente 40% da energia global e são responsáveis por quase um terço das emissões mundiais de gases de efeito estufa. Na última década, normas rigorosas para a construção de edifícios verdes levaram a melhorias significativas na qualidade das características térmicas, além da forma como muitos deles são planejados (AHMAD et al., 2018), pois a diminuição do impacto negativo das atividades humanas sobre o meio ambiente tornou-se uma prioridade universal, assim como a procura por soluções sustentáveis para a redução do consumo elétrico. O futuro depende de atitudes ambientalmente corretas, socialmente justas e economicamente viáveis, da mesma forma que a utilização eficiente dos recursos naturais integrando novas soluções energéticas (PUCRS, 2010). Evitar o desperdício é sem dúvida outra justificativa importante para a realização de estudos com essa finalidade uma vez que trabalhar o consumo consciente de energia em qualquer organização é um desafio, pois é necessário saber engajar todos para que os maus hábitos relacionados ao desperdício sejam desencorajados e deem lugar às boas práticas. Atitudes simples, como desligar o que não esteja em uso, não deixar carregadores plugados na tomada sem utilização, aproveitar períodos do dia para diminuir o uso dos aparelhos com maior consumo, todos esses hábitos podem ajudar na economia de energia elétrica corporativa e fomentar o uso consciente (ALSOL, 2019). Neste sentido os aspectos acima relacionados impulsionaram a elaboração deste estudo.

Outra razão para realização desta pesquisa está diretamente relacionada com a economia financeira e a redução de custos com a execução de projetos de sistemas para estipular a ocupação de um determinado local, uma vez que o preço de um sensor de ocupação autônomo e com fio de alta qualidade pode custar R$ 200 ou mais. Os dispositivos sem fio podem oferecer custos mais baixos de instalação, apesar de não serem totalmente confiáveis em relação à comunicação de dados ou por serem alimentados com baterias ou sistemas de economia de energia que aumentam o custo e têm seus próprios problemas de confiabilidade. Finalmente, a segurança, que também pode figurar entre os propulsores para estudos de detecção da presença, a fim de que os ocupantes possam usufruir na plenitude dos edifícios, estes devem satisfazer requisitos arquitetônicos, funcionais, ecológicos, e de proteção. A utilização e o fim a que se destina cada edifício determinam a instalação e implementação de diferentes medidas de proteção e segurança.

Uma breve revisão da literatura científica, permitiu a identificação de diferentes abordagens nos trabalhos relacionados a temática deste estudo. É importante salientar que embora seja um levantamento parcial, ficou claro que o interesse em determinar a ocupação do ambiente de trabalho, ou até mesmo ambientes residenciais, é muito recente, pois os levantamentos mostraram que as pesquisas começaram a ser desenvolvidas a partir das últimas duas décadas, provavelmente por causa das questões ambientais que receberam um maior foco durante esse período.

O aumento na quantidade de dispositivos de uso pessoal com recursos de comunicação sem fio, foi determinante para a realização de um levantamento com o objetivo de mensurar a quantidade de pessoas em um mesmo ambiente, com base na captura de quadros de Wi-Fi e Bluetooth ou Bluetooth Low Energy transmitidos pelos dispositivos dos usuários. Uma taxa de 97% de acerto na precisão da ocupação foi relatada por (LONGO, REDONDI e CESANA, 2019).

Duas abordagens heurísticas foram aplicadas: análise de cluster e modelos baseados em fluxogramas lógicos. Numa pesquisa construída a partir de um escritório em uma universidade, com o objetivo de obter informações sobre padrões de ocupação, monitorando, estado da ocupação, temperatura do ar, umidade relativa, CO_2, VOC, abertura de portas e janelas e uso da eletricidade, em (MORA et al., 2019).

Uma metodologia de aprendizado interativo foi proposta por (Amayri et al., 2019) “para evitar a rotulagem manual da ocupação real de uma câmera de vídeo em uma sala fazendo uso de uma abordagem de aprendizado supervisionado”. Para esse experimento foi empregado um algoritmo puro da árvore de decisão C4.5 e um classificador baseado em regras parametrizado, em conjunto com o processo de aprendizado interativo. A média de erro retornada foi igual a 0,032 por pessoa.

No método empregado por (Lam et al., 2009) “diferentes recursos de dados foram estudados e ordenados usando o conceito de ganho relativo de informação (RIG). A correlação entre o número de ocupantes foi classificada em 77,65% para umidade, 73,42% para acústica, 67,14% para CO_2 e 37,39% para temperatura”.

Um estudo da detecção de ocupação de salas de escritórios, baseado no uso de três conjuntos de dados gerados por sensores de luz, umidade, CO_2 e temperatura combinados com os algoritmos Random Forest (RF), Gradient Boosting Machines (GBM), Linear Discriminant Analysis (LDA) e Classification and Regression Trees (CART), foi proposto por (CANDANEDO e FELDHEIM, 2016). O ensaio com o LDA aplicado ao segundo conjunto de dados foi o que retornou a melhor precisão, 99,33%.

1.1 Objetivo Geral

Avaliar a performance de um algoritmo de aprendizagem de máquina sobre um conjunto de dados obtido a partir de sensores de luz, temperatura, CO_2 e umidade.

1.2 Objetivos Específicos

- Realizar a detecção de pessoas com base nos dados de sensores de luz, temperatura, CO_2 e umidade, utilizando um algoritmo de classificação.

- Encontrar um conjunto de hiperparâmetros do modelo de classificação que otimizem seu funcionamento para a aplicação específica.

- Comparar os resultados alcançados com os resultados relacionados na literatura.

1.3 Perguntas de Pesquisa

Com a execução deste trabalho, esperamos responder à seguinte pergunta de pesquisa:

Q1: É possível realizar a detecção de ocupação de escritórios utilizando um algoritmo de classificação baseado em árvores de decisão com desempenho melhor ou igual ao documentado na literatura?

Para efetiva solução do problema em tela, propõe-se a utilização do algoritmo de classificação C5.0 (RULEQUEST, 2019), utilizando dados reais coletados a partir de sensores de luz, umidade, CO_2 e temperatura instalados em uma sala de escritório fixados em pontos estratégicos, para geração de um modelo de classificação baseado em árvores de decisão. A partir da configuração dos atributos da aplicação (conjunto de dados) e da combinação de alguns argumentos nativos do classificador C5.0, possibilitando a efetivação de vários ensaios em diferentes cenários. Com os resultados obtidos das saídas geradas pelo algoritmo classificador uma análise será realizada com o auxílio de algumas métricas estatísticas para determinar a porcentagem da precisão dos resultados alcançados, permitindo assim um confrontamento com resultados estabelecidos na literatura e listados neste estudo,

melhorando a confiabilidade dos índices de acerto da detecção de pessoas em um ambiente de trabalho. Este trabalho trata da tarefa de detecção de ocupação supervisionada, no qual dado um conjunto de observações ou exemplares rotulados, ou seja, conjunto de observações em que a classe, também chamada de atributo alvo de cada exemplo é conhecida, onde a finalidade é encontrar uma hipótese capaz de classificar novas observações entre as classes já existentes (PRATI, 2006).

Pesquisas recentes apontam que o uso do C5.0 para o estudo do desempenho energético dos edifícios é bastante confiável e pode fornecer uma solução mais rápida e adequada nessas situações. A ocupação em um edifício é um fator importante no consumo de energia, e a detecção precisa da ocupação é, portanto, necessária e útil para gerenciar os requisitos de energia com mais eficiência. A aplicação dessa pesquisa é perfeitamente viável em prédios comerciais, uma vez que combinações desses sensores já podem ser encontradas em muitos prédios. Medições experimentais dessa natureza relataram que a economia de energia foi de 37% em (BROOKS et al., 2014) e entre 29% e 80% quando os dados de ocupação foram usados como entrada para os algoritmos de controle de um sistema de ventilação e ar-condicionado HVAC (BROOKS et al., 2015).

1.4 Estrutura do Trabalho

A seguir, esta monografia está dividida da seguinte forma: O segundo capítulo, a fundamentação teórica, apresenta o embasamento teórico necessário a este trabalho, podendo servir de base para a análise e interpretação dos dados coletados. A primeira seção do segundo capítulo discorre sobre aprendizagem de máquina e suas diversas tarefas de uma forma generalizada, enquanto a segunda seção do mesmo capítulo aborda a preparação dos conjuntos de dados, que é o primeiro passo para adentrar no processo de mineração de dados. A terceira seção abrangerá a tarefa de classificação. Na quarta seção nomeada, Avaliação do modelo de classificação, uma abordagem das diversas métricas de medidas de variabilidade estatística, e a validação cruzada será apresentada.

No terceiro capítulo, intitulado materiais e métodos, será descrito de forma clara e precisa como o estudo foi executado. No quarto capítulo serão apresentados e discutidos os resultados obtidos, além de comparações com as soluções propostas na literatura. Finalmente, o quinto capítulo conclui este trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Aprendizagem de Máquina

A ciência da computação possui diferentes áreas de estudo, uma delas é a inteligência artificial, que da mesma forma se divide em diferentes áreas de aplicação, entre elas a aprendizagem de máquina (AM). Machine Learning (ML), termo em inglês, tem como objetivo o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Uma das primeiras abordagens sobre AM foi a de Arthur Samuel em 1959 afirmando se tratar do "campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados" (SIMON, 2013). Outra definição importante é a de Mitchell (1997) "Diz-se que um programa de computador aprende com a experiência, com relação a alguma classe de tarefas, e com a medida de desempenho, se o desempenho nas tarefas, medido pela [medida de desempenho], melhora com a experiência". Em uma abordagem mais recente ML tem suas origens em três disciplinas: inteligência artificial, estatística computacional e reconhecimento de padrões, sendo definida como um conjunto de técnicas

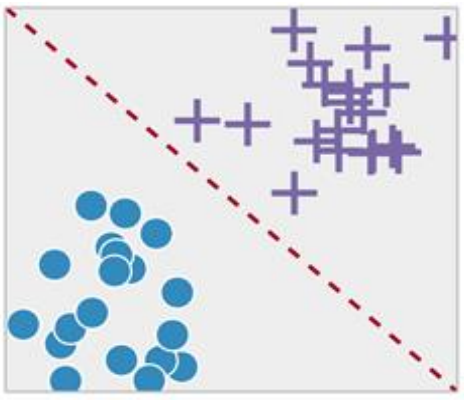
computacionais de análise de dados baseadas no aprendizado para fins de previsão, descrição ou explicação orientada por dados (CAMACHO et al., 2018).

A produção de um modelo de aprendizado de máquina, exige inicialmente a determinação do que se deseja encontrar no conjunto de dados, desta forma a tarefa correta de aprendizado de máquina poderá ser empregada para a finalidade desejada. Outro ponto importante após a escolha da tarefa mais adequada para o cenário proposto, é a escolha de um algoritmo para o treinamento do modelo. Um algoritmo de aprendizado de máquina é um processo usado para ajustar um modelo a um conjunto de dados, por meio de treinamento ou aprendizado (WILLCOCK et al., 2018). Uma listagem descrevendo as diferentes tarefas de aprendizado de máquina assim como alguns exemplos de aplicação para cada tarefa, pode ser conferido a seguir.

2.1.1 Classificação

Uma tarefa de classificação é um processo que permite a descoberta de conhecimento a partir de um conjunto de dados, por meio da categorização de suas características, criando um classificador a partir de um conjunto de exemplos rotulados. Este conjunto de exemplos consiste em grupos de instâncias já pré-classificadas, ou seja, com o atributo alvo (às vezes chamado de rótulo) da classificação já valorado. O classificador é treinado para classificar novos grupos de instâncias com pontuações desconhecidas para cada instância. No contexto das tarefas de aprendizado de máquina, o processo de classificação se dá por meio de um algoritmo indutor, cujo objetivo é predizer o rótulo de novas entradas com base nos exemplos de entrada já rotulados (FERREIRA, 2016). A Figura 1 ilustra quando um conjunto de instâncias assume valores discretos, de acordo com o tipo a qual pertence o atributo classe.

Figura 1 - Classificação de instâncias



Fonte: SCORECARDSTREET (2019)

Classificação Binária

Essa é talvez a mais comum entre as tarefas, a Classificação binária, tem a finalidade de identificar a qual classe pertence um determinado registro, onde o modelo analisa o conjunto de registros fornecidos, com cada registro já contendo a indicação à qual classe pertence, a fim de “aprender” como classificar um novo registro (aprendizado supervisionado) (CAMILO, 2009). Por exemplo, categorizamos cada registro de um conjunto de dados contendo as informações sobre os funcionários de uma empresa: Perfil Técnico, Perfil Negocial e Perfil Gerencial. O modelo analisa os registros e então é capaz de dizer em qual categoria um novo colaborador se encaixa (CAMILO, 2009). Essa é a tarefa de aprendizado que será aplicada na produção do modelo de aprendizado de máquina neste estudo. Exemplos de cenários de classificação binária incluem:

- Reconhecer algo como "positivo" ou "negativo".
- Diagnosticar se um paciente tem uma determinada doença ou não.
- Tomar a decisão de marcar um e-mail como "spam" ou não.

Classificação de Multiclasse

É o nome dado a tarefa de classificação onde existem mais de duas classes a serem inferidas. Diferente do processo dos problemas de classificação binária, não se torna necessário a escolha de um limite de pontuação para fazer previsões. A resposta prevista é a classe (rótulo) com a maior pontuação prevista (CAMILO, 2019). Exemplos de classificação multiclasse:

- Determinar a raça de um cão como um "Husky Siberiano", "Golden Retriever", "Poodle", etc.
- Entender as resenhas de filmes como "positivas", "neutras" ou "negativas".
- Categorizar as avaliações de hotel como "local", "preço", "limpeza", etc.

2.1.2 Agrupamento ou Clustering

A tarefa de agrupamento visa identificar e aproximar os registros similares. Um agrupamento (ou cluster) é uma coleção de registros similares entre si, porém diferentes dos outros registros nos demais agrupamentos. Esta tarefa difere da classificação pois não

necessita que os registros sejam previamente categorizados (aprendizado não-supervisionado). Além disso, ela não tem a pretensão de classificar, estimar ou predizer o valor de uma variável, ela apenas identifica os grupos de dados similares (CAMILO, 2009). Exemplos:

- Compreender os segmentos de hóspedes do hotel com base nos hábitos e características das opções de hotel.
- Identificar segmentos de clientes e dados demográficos para ajudar a criar campanhas de publicidade segmentadas.
- Categorizar o inventário com base nas métricas de fabricação.

2.1.3 Detecção de Anomalias

Esta tarefa cria um modelo de detecção de anomalias usando a Análise de Componente Principal (PCA). A Detecção de Anomalias Baseada em PCA ajuda a criar um modelo em cenários onde é fácil obter dados de treinamento de uma classe, como transações válidas, mas é difícil obter exemplos suficientes das anomalias direcionadas (CAMILO, 2009). A detecção de anomalias abrange várias tarefas importantes no aprendizado de máquina, como por exemplo:

- Identificar transações que são potencialmente fraudulentas.
- Reconhecer padrões que apontam para a ocorrência de uma invasão da rede.
- Localizar clusters anormais de pacientes.

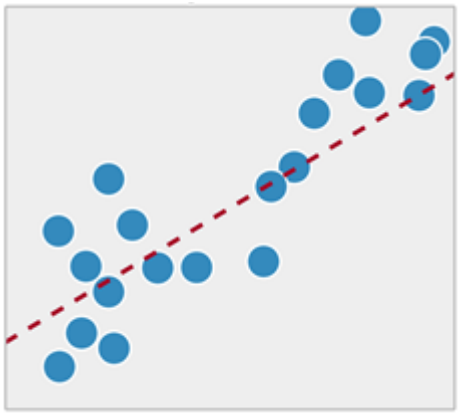
2.1.4 Regressão

Essa tarefa é usada quando o registro é identificado por um valor numérico e não um categórico. Assim, pode-se estimar o valor de uma determinada variável analisando-se os valores das demais conforme ilustrado na Figura 2. Por exemplo, um conjunto de registros contendo os valores mensais gastos por diversos tipos de consumidores e de acordo com os hábitos de cada um. Após ter analisado os dados, o modelo é capaz de dizer qual será o valor gasto por um novo consumidor (CAMILO, 2009). A tarefa de estimação pode ser usada por exemplo para:

- Previsão de preços de casas com base nos atributos da casa, como número de quartos, localização ou tamanho.

- Previsão de preços futuros de ações com base em dados históricos e tendências atuais do mercado.
- Previsão de vendas de um produto com base em orçamentos de publicidade.

Figura 2 - Tarefa de regressão



Fonte: SCORECARDSTREET (2019)

2.1.5 Recomendação

Uma tarefa de recomendação permite produzir uma lista de produtos ou serviços recomendados, usando a Fatoração Matricial (MF), um algoritmo de filtragem colaborativa para as recomendações quando se tem dados históricos de classificação do produto em seu catálogo. Por exemplo, a existência de dados históricos de classificação de filmes para usuários caso se deseje recomendar outros filmes que eles provavelmente assistirão (MICROSOFT, 2019).

2.2 Preparação de Dados

O Data Mining, do termo em inglês, é um conjunto de procedimentos que permitem examinar grandes bases de dados, automática ou semi-automaticamente, com o objetivo de encontrar padrões repetidos que explicam o comportamento desses dados. Mas os conceitos de Mineração de Dados podem ser considerados multidisciplinar, pois as definições acerca desse termo variam com o campo de atuação dos autores. Na perspectiva de Hand et al. (2006), a definição é dada a partir de uma visão estatística: "Mineração de Dados é a análise de grandes conjuntos de dados a fim de encontrar relacionamentos inesperados e de resumir os dados de uma forma que eles sejam tanto úteis quanto compreensíveis ao dono dos dados". Ao processo de preparação dos dados para a mineração, que por sinal é seu primeiro passo,

também dar-se o nome de pré-processamento, segundo (Han et al., 2006) e suas principais etapas são:

Seleção de Atributos

Também conhecida como redução de dimensionalidade, essa etapa é muito importante quando se trabalha com conjuntos de dados com muitos atributos, dado que o custo computacional de várias técnicas de mineração de dados aumenta com a quantidade de atributos.

Redução dos Dados

O volume de dados usado na mineração costuma ser alto. Em alguns casos, este volume é tão grande que torna o processo de análise dos dados e da própria mineração impraticável. Nestes casos, as técnicas de redução de dados podem ser aplicadas para que a massa de dados original seja convertida em uma massa de dados menor, ou seja, com uma quantidade menor de instâncias, porém, sem perder a representatividade dos dados originais. Isto permite que os algoritmos de mineração sejam executados com mais eficiência, mantendo a qualidade do resultado (CAMILO, 2006).

Limpeza de Dados

É muito comum encontrar inconsistências nos conjuntos de dados, tais como: registros incompletos, valores errados e dados inconsistentes (CAMILO, 2009). A limpeza dos dados é um procedimento que visa preencher dados ausentes, “alisar” ruído, identificar e/ou remover valores aberrantes, resolver inconsistências. As técnicas usadas nesta etapa vão desde a remoção da instância com problemas, passando pela atribuição de valores padrões (como a média dos valores do atributo em questão), até a aplicação de técnicas de agrupamento para auxiliar na descoberta dos melhores valores. Devido ao grande esforço exigido nesta etapa, (Han et al., 2006) propõem o uso de um processo específico para a limpeza dos dados.

Integração dos Dados

É comum obter-se os dados a serem minerados de diversas fontes: banco de dados, arquivos textos, planilhas, data warehouses, vídeos, imagens, entre outras. Surge então, a necessidade da integração destes dados de forma a termos um repositório único e consistente. Para isto, é necessária uma análise aprofundada dos dados observando redundâncias, dependências entre as variáveis e valores conflitantes (categorias diferentes para os mesmos valores, chaves divergentes, regras diferentes para os mesmos dados, entre outros) (CAMILO, 2009).

Transformação dos Dados

Nesta etapa alguns algoritmos trabalham apenas com valores numéricos e outros apenas com valores categóricos. Nestes casos, é necessário converter os valores numéricos para categóricos ou os categóricos em valores numéricos. Não existe um critério único para transformação dos dados e diversas técnicas podem ser usadas de acordo com os objetivos pretendidos. Algumas das técnicas empregadas nesta etapa são: suavização (remove valores errados dos dados), agrupamento (agrupa valores em faixas sumarizadas), generalização (converte valores muito específicos para valores mais genéricos), normalização (colocar as variáveis em uma mesma escala) e a criação de novos atributos, também chamados de atributos derivados (gerados a partir de outros já existentes) (CAMILO, 2019).

2.3 Classificação

Algoritmos de Classificação

Sem dúvida o subcampo da Inteligência Artificial, Machine Learning, é a tendência mais atraente nos setores de tecnologia atualmente, a ML é suficientemente poderosa para fazer previsões ou sugestões calculadas com base em grandes quantidades de dados. No aprendizado de máquina, a classificação é uma abordagem de aprendizado supervisionado, na qual o programa de computador aprende com a entrada de dados fornecida e, em seguida, usa esse aprendizado para classificar novas observações. Esse conjunto de dados pode ser simplesmente de duas classes (como identificar se a pessoa é homem ou mulher ou se o e-mail é spam ou não spam) ou também pode ser de várias classes (MEDIUM, 2019). Alguns

exemplos de problemas de classificação são: reconhecimento de fala, reconhecimento de manuscrito, identificação biométrica, classificação de documentos etc., e os algoritmos de classificação se tornam uma peça chave na evolução do ML. Alguns exemplos de algoritmos de classificação serão listados a seguir.

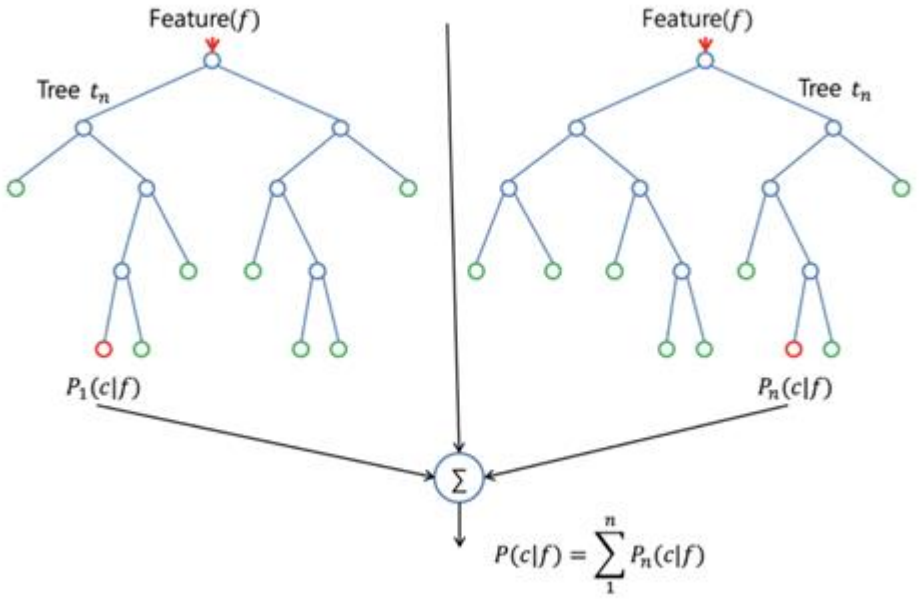
Árvore de Decisão

Árvores de Decisão são métodos de aprendizado de máquina supervisionados não-paramétricos, muito utilizados em tarefas de classificação e regressão, é certamente, o algoritmo de Aprendizagem de Máquina mais estudado para aplicações em Mineração de Dados (WITTEN e FRANK, 2000). Por fazer parte do assunto de pesquisa e, consequentemente, da proposta deste trabalho, uma interpretação mais detalhada do algoritmo é necessária, sendo assim o seção 2.3.1 contém um esboço do funcionamento de um algoritmo padrão de Árvores de Decisão, da mesma forma a seção 2.3.2 apresenta um detalhamento do algoritmo C5.0, derivado do C4.5 que é o modelo mais conhecido de árvores de decisão.

Floresta Aleatória

Floresta Aleatória (random forest) é um algoritmo de aprendizagem supervisionada. A “floresta” a que se refere o nome do algoritmo, é uma combinação (ensemble) de árvores de decisão, na maioria dos casos, treinados com o método de bagging. A ideia principal do método de bagging é que a combinação dos modelos de aprendizado aumenta o resultado geral. De modo simples, o algoritmo de florestas aleatórias cria várias árvores de decisão e as combina para obter uma predição com maior acurácia e mais estável. A Figura 3, mostra uma floresta aleatória com duas árvores (MEDIUM, 2019).

Figura 3 - Floresta Aleatória com duas árvores

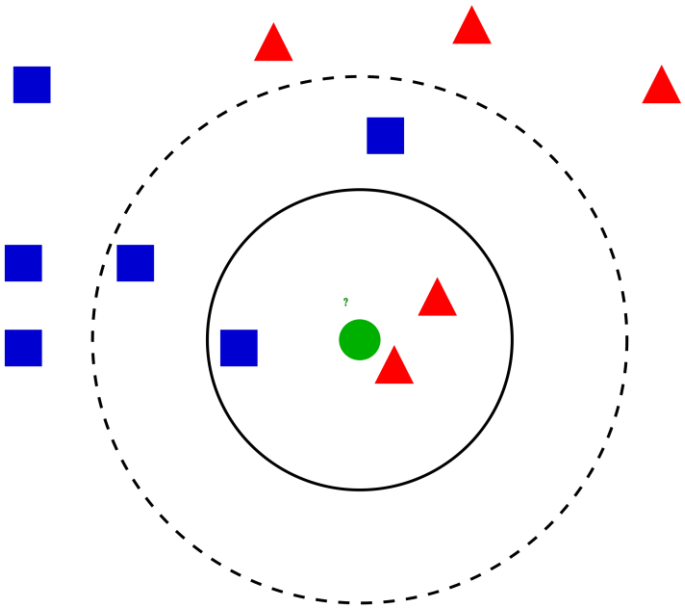


Fonte: MEDIUM (2019)

K-Vizinho Mais Próximo

O algoritmo de K-Vizinho Mais Próximo é um eficaz algoritmo não-paramétrico de classificação/regressão, sendo utilizado desde de 1950 na área de Estatística. É um algoritmo demorado, porém eficaz, sendo recomendado para bases de dados que contenham muitas instâncias, já que dependendo do valor escolhido para k (geralmente um número ímpar), não é seguro utilizar o algoritmo em conjuntos de dados menores, tornando o algoritmo propício a ruído no conjunto de dados e afetando de forma significativa a precisão do mesmo. Seu funcionamento é relativamente simples: ao se classificar uma nova instância, o algoritmo busca as k instâncias que possuem a menor distância em relação à nova instância. Outra grande desvantagem deste algoritmo, e que deve ser analisada com cuidado ao se considerar a sua utilização, é que todos os aspectos de uma instância possuem o mesmo peso ao se calcular a distância. Ou seja, caso uma característica tenha uma importância maior do que as outras na hora de se classificar uma instância, esta importância acaba sendo descartada por este algoritmo. (WITTEN e FRANK, 2000). A Figura 4 é uma representação simplificada desse algoritmo.

Figura 4 - Ilustração simplificada do algoritmo K-Vizinho Mais Próximo



Fonte: ANALYTICS VIDHYA (2019)

Naive Bayes

Um classificador Naive Bayes é um algoritmo de aprendizado de máquina supervisionado que usa o Teorema de Bayes, que se baseia na "ingênua" suposição de que as variáveis de entrada são independentes uma da outra, ou seja, não há como saber nada sobre outras variáveis quando recebe uma variável adicional. Independentemente dessa suposição, ele prova ser um classificador com bons resultados. Os classificadores Naive Bayes contam com o teorema de Bayes, que é baseado em probabilidade condicional ou em termos simples, a probabilidade de um evento A acontecer, dado que outro evento B já aconteceu. Essencialmente, o teorema permite que uma hipótese seja atualizada sempre que novas evidências são introduzidas. A equação 1 expressa o Teorema de Bayes, em que P(A|B) é a probabilidade de A ocorrer dado que ocorreu B e P(B|A) é a probabilidade de ocorrer B dado que ocorreu A. A seguir é apresentado o coeficiente de Bayes, no qual o classificador é baseado (MEDIUM, 2019).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

Redes Neurais Artificiais

O estudo de Redes Neurais Artificiais foi inspirado, em grande parte, pela observação de que os sistemas de aprendizado biológicos são formados por complexas redes de neurônios interligados (MITCHELL, 1997). Portanto, os algoritmos de Redes Neurais Artificiais muitas vezes possuem como objetivo tentar simular as capacidades de processamento de um cérebro humano, por meio de unidades (ou nós) de processamento simples que modelam um neurônio biológico. Na área de AM, Redes Neurais Artificiais são consideradas algoritmos poderosos, sendo utilizadas para resolução de problemas lineares ou não-lineares, tendo suporte tanto a aprendizado supervisionado quanto não supervisionado. Sua utilização mais frequente é na resolução de problemas não-lineares de Classificação, por meio de uma estrutura de rede chamada de Perceptron Multi-Camada (Multi-Layer Perceptron, ou MLP). O algoritmo mais conhecido de aprendizado para Redes Neurais Artificiais é o algoritmo de Retropropagação (backpropagation), sendo que a modelagem de Redes Neurais Artificiais voltadas para a área de AM são implementadas utilizando estes dois algoritmos em conjunto.

Regressão Linear

Esse algoritmo consiste em representar a saída esperada com o aspecto de uma função linear, onde cada instância é relacionada com um peso. Ele é um algoritmo estatístico e paramétrico no qual, sua principal vantagem é a sua simplicidade, de forma resumida, sendo reconhecidamente o algoritmo mais simples e utilizado para a criação de modelos de regressão. Por outro lado, o fato de trabalhar apenas com dados numéricos constitui uma desvantagem no uso desse algoritmo, além de sua eficiência em problemas não-lineares ser baixíssima, devido a sua limitação de tentar transformar uma função não-linear em um modelo linear simples. Ou seja, sua utilização é muito limitada. (WITTEN e FRANK, 2000).

Regressão Logística

Apesar do seu nome, este é um algoritmo de classificação e não um método de regressão. A regressão logística realiza classificação binária, portanto os rótulos de saída são dicotômicos. Ao ser definido P (y=1 | x) como a probabilidade condicionada de que a saída y é 1 quando dado como entrada um vetor de características x. Os coeficientes w são os pesos do modelo que serão encontrados pelo algoritmo.

$$P(y = 1|x) = \frac{1}{1 + e^{-w^t x}} \quad (2)$$

É uma técnica recomendada para situações em que a variável dependente é de natureza dicotômica ou binária. Quanto às independentes, tanto podem ser categóricas ou não. A regressão logística é um recurso que nos permite estimar a probabilidade associada à ocorrência de determinado evento em face de um conjunto de variáveis explanatórias. (MEDIUM, 2019).

2.3.1 Árvores de Decisão

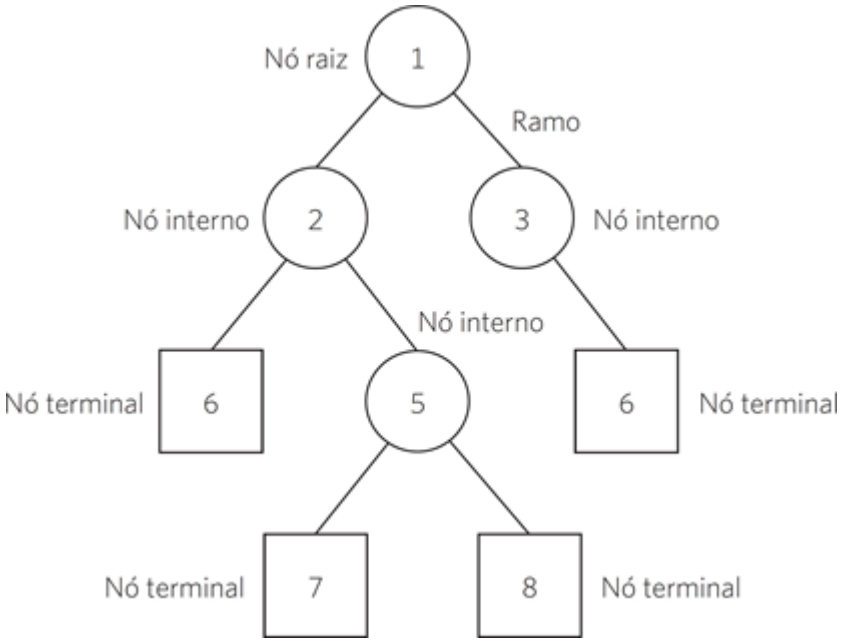
As árvores de decisão estão entre os mais populares algoritmos de inferência e tem sido aplicado em várias áreas como, por exemplo, diagnóstico médico e risco de crédito (MITCHELL, 1997), e deles pode-se extrair regras do tipo "se - então" que são facilmente compreendidas. Muitas definições sobre árvore de decisão são encontradas na literatura, mas de uma forma genérica pode-se definir como um classificador que utiliza diversos sistemas de aprendizado de máquina. Uma árvore de decisão é induzida a partir de um conjunto de dados de treinamento onde as classes são previamente conhecidas. Em Maxwell (2019) é encontrada a seguinte interpretação "Árvores de decisão são modelos estatísticos que utilizam um treinamento supervisionado para a classificação e previsão de dados. Em outras palavras, em sua construção é utilizado um conjunto de treinamento formado por entradas e saídas, estas últimas são as classes". Para Gama (2004) "estes modelos utilizam a estratégia de dividir para conquistar: um problema complexo é decomposto em subproblemas mais simples e recursivamente esta técnica é aplicada a cada sub - problema".

Interpretação de uma Árvore de Decisão

Uma árvore de decisão pode ser retratada da seguinte forma, cada nó interno (não-folha) é rotulado com o nome de um dos atributos previsores, em seguida os ramos (ou arestas) saindo de um nó interno são rotulados com valores do atributo naquele nó, a partir daí cada folha é rotulada com uma classe, a qual é a classe prevista para exemplos que pertençam àquele nó folha, no espaço definido pelos atributos, cada folha corresponde a um hiper-retângulo onde a interseção destes é vazia e a união é todo o espaço (GAMA, 2004). O processo de classificação de um exemplo ocorre fazendo aquele exemplo "caminhar" pela

árvore, a partir do nó raiz, procurando percorrer os arcos que unem os nós, de acordo com as condições que estes mesmos arcos representam. Ao atingir um nó folha, a classe que rotula aquele nó folha é atribuída àquele exemplo. Nos casos em que a árvore é usada para classificação, os critérios de divisão mais conhecidos são baseados na entropia e índice Gini. (Onoda, 2001) A Figura 5 ilustra um modelo de árvore de decisão.

Figura 5 - Representação de uma Árvore de Decisão



Fonte: SCIELO (2019)

Entropia

A Entropia é definida como o cálculo do ganho de informação baseado em uma medida utilizada na teoria da informação, ela atua na caracterização da impureza dos dados. Em um conjunto de dados, é uma medida da falta de homogeneidade dos dados de entrada em relação a sua classificação (COIMBRA, 2004). A entropia é utilizada para calcular o Ganho de Informação. O algoritmo ID3 foi um dos pioneiros na indução de árvores de decisão, utilizando essa medida. Para determinar a qualidade da condição de teste realizada, é necessário comparar o grau de entropia do nó-pai (antes da divisão) com o grau de entropia dos nós-filhos (após a divisão). O atributo que gerar uma maior diferença é escolhido.

$$I_H = -\sum_{j=1}^{C} p_j \, log_2(p_j) \qquad (3)$$

Índice Gini

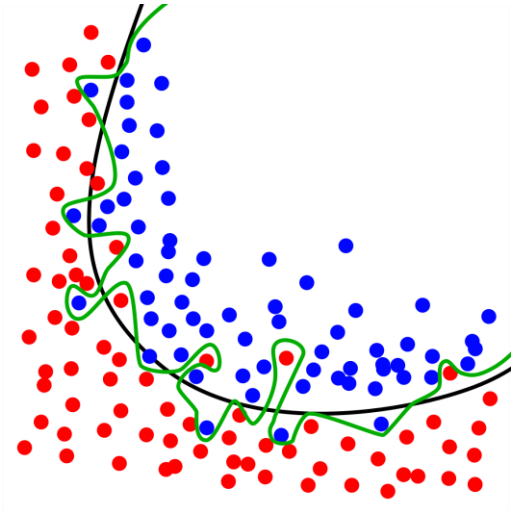
Outra medida bastante conhecida é o Gini, proposto em 1912 pelo estatístico italiano Corrado Gini, que mede o grau de heterogeneidade dos dados, podendo ser utilizado para medir a impureza de um nó, que representa uma única decisão em uma árvore de decisão. O índice Gini é uma maneira de medir o quão "impuro" é um nó. fazendo uso de um índice de dispersão estatístico. Ele é muito utilizado em análises econômicas e sociais, por exemplo, para quantificar a distribuição de renda em um certo país.

$$I_G = 1 - \sum_{j=1}^{c} p_j^2 \qquad (4)$$

Overfitting

Overfitting é o ajuste exagerado dos dados de treinamento. Gama (2004), define esse ajuste como “uma árvore de decisão (d) que faz sobre-ajustamento aos dados se existir uma árvore (d) de tal modo que: (d) tem menor erro que (d) no conjunto de treino mas (d) tem menor erro na população”. Isso significa dizer que a adaptação excessiva acontece quando um modelo aprende os detalhes e o ruído nos dados de treinamento na medida em que afeta negativamente o desempenho do modelo em novos dados, ou seja, o ruído nos dados de treinamento é captado e aprendido como conceito pelo modelo. O problema é que esses conceitos não se aplicam a novos dados e afetam negativamente a capacidade de generalização dos modelos. No algoritmo de partição recursiva, a árvore estende a sua profundidade até o ponto de classificar perfeitamente os elementos do conjunto de treinamento. Quando o conjunto de treinamento não possui ruído, o número de erros no treinamento pode ser zero. Quando este conjunto, entretanto, possui ruído, ou quando o conjunto de treinamento não é representativo, este algoritmo pode produzir árvores em que há *overfitting*. A Figura 6 apresenta um esquema simplificado de *overfiting.*

Figura 6 - Overfitting (Excesso de ajuste)



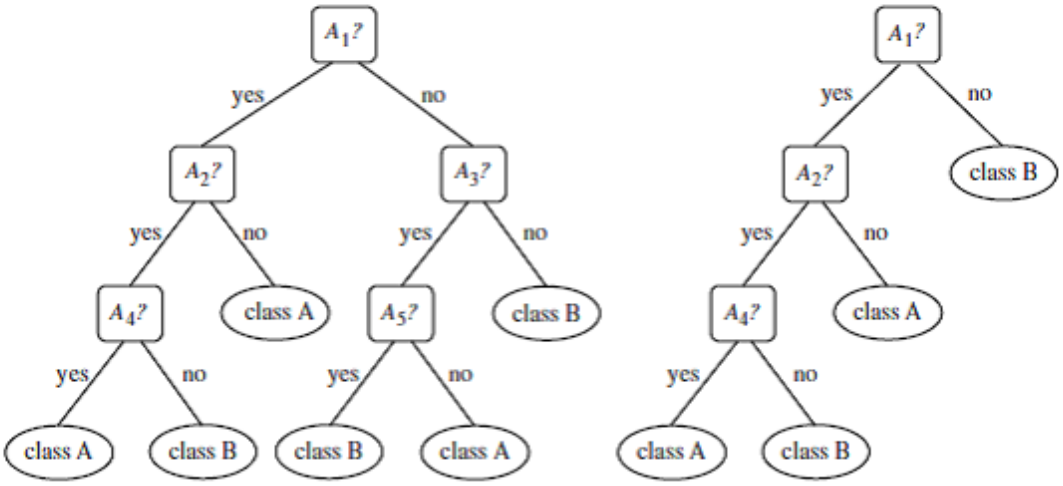
Fonte: WIKIMEDIA (2019)

Poda

No aprendizado de máquina e na mineração de dados, a poda ou pruning (termo em inglês) é uma técnica associada às árvores de decisão. A remoção reduz o tamanho das árvores de decisão removendo partes da árvore que não fornecem dados para classificar instâncias. As árvores de decisão são as mais suscetíveis de todos os algoritmos de aprendizado de máquina à adaptação excessiva e a poda eficaz, que podem reduzir essa probabilidade O processo de podagem pode ser feito por meio dos seguintes passos, a árvore é percorrida em sua profundidade, e para cada nó de decisão é calculado o erro no nó e a soma dos erros nos nós descendentes, se o erro do nó é menor ou igual à soma dos erros dos nós descendentes então o nó é transformado em folha. A Figura 7 mostra como exemplo a mesma árvore antes e depois do processo de poda (SHAREENGINEER, 2019). Duas técnicas diferentes de poda podem ser aplicadas para ajudar na adaptação excessiva, a pré - poda ou parada precoce e pós-poda.

Na abordagem de pré - poda, uma árvore é “podada” interrompendo sua construção mais cedo. A segunda e mais comum abordagem é a pós - poda, que remove as sub - árvores de uma árvore “totalmente crescida”. Uma sub - árvore e seus ramos em um determinado nó é removida, e substituída por uma folha. A folha é rotulada com a classe mais frequente na sub - árvore sendo substituída (SHAREENGINEER, 2019).

Figura 7 - Árvore de Decisão antes e depois do processo de podagem



Fonte: BLOGSPOT (2019)

Algoritmos

Existem vários algoritmos de classificação que utilizam a árvore de decisão, conforme citado anteriormente, não se pode afirmar que um é melhor do que o outro, pois isso depende da finalidade para qual cada um será utilizado, nesse caso um algoritmo pode ser mais eficiente que outro. A Figura 8 ilustra um algoritmo de árvore de decisão.

Figura 8 - Algoritmo de Indução em Árvore de Decisão.

INPUT: S, **where** S = *set of classified instances*
OUTPUT: *Decision Tree*
Require: $S \neq \varnothing$, $num\_attributes > 0$

```
1:  procedure BuildTree
2:      repeat
3:          maxGain ← 0
4:          splitA ← null
5:          e ← Entropy(Attributes)
6:          for all Attributes a in S do
7:              gain ← InformationGain(a, e)
8:              if gain > maxGain then
9:                  maxGain ← gain
10:                 splitA ← a
11:             end if
12:         end for
13:         Partition(S, splitA)
14:     until all partitions processed
15: end procedure
```

Fonte: kdnuggets (2019)

Entre os algoritmos mais conhecidos pode ser destaca: ID3 (QUINLAN, 1979), CART – Classification and Regression Trees (BREIMAN, et al., 1984), Assistant (CESTNIK, et al., 1987), C4.5 (QUINLAN, 1993), C5.0 (See5), CHAID (Chi Square Automatic Interaction Detection). Neste trabalho será utilizado o algoritmo C5.0.

CART

O algoritmo CART (Classification and Regression Trees) foi apresentado por Breiman et al., (1984) e traduz-se em uma técnica não-paramétrica que induz tanto árvores de classificação quanto árvores de regressão, dependendo se o atributo é nominal (classificação) ou contínuo (regressão). Este algoritmo possui muitas vantagens dentre as quais podem ser destacadas, a grande eficiência em pesquisa de relações entre os dados, mesmo quando elas não são claras, bem como a produção de resultados sob a forma de árvores de decisão de grande simplicidade e legibilidade (FONSECA, 1994). As árvores geradas pelo algoritmo CART são geralmente do tipo binárias, podendo ser percorridas da sua raiz até as folhas respondendo apenas a questões elementares do tipo “sim” ou “não”. Os nós que correspondem a atributos contínuos são representados por grupos de valores em dois conjuntos. Da mesma forma como ocorre no algoritmo C 5.0, o CART utiliza a métodos de pesquisa exaustiva para definir os limiares a serem utilizados nos nós para dividir os atributos contínuos. Adicionalmente, o CART dispõe de um tratamento especial para atributos ordenados e também permite a utilização de combinações lineares entre atributos (agrupamento de valores em vários conjuntos). Diferente das abordagens adotadas por outros algoritmos, os quais utilizam pré-poda, o CART expande a árvore exaustivamente, realizando pós-poda por meio da redução do fator custo-complexidade (BREIMAN et al., 1984). Segundo alguns os autores, a técnica de poda utilizada é muito eficiente e produz árvores mais simples, precisas e com boa capacidade de generalização.

ID3

ID3 é um algoritmo simples que construí uma árvore de decisão sob os seguintes argumentos: Cada vértice (nó) corresponde a um atributo, e cada aresta da árvore a um valor possível do atributo. Uma folha da árvore corresponde ao valor esperado da decisão segundo os dados de treino utilizados. A explicação de uma determinada decisão está na trajetória da raiz a folha representativa desta decisão. Cada vértice é associado ao atributo mais

informativo que ainda não tenha sido considerado. Para medir o nível de informação de um atributo se utiliza o conceito de entropia da Teoria da Informação. Quanto menor o valor da entropia, menor a incerteza e mais utilidade tem o atributo para a classificação. A principal limitação do ID3 é que ele só lida com atributos categóricos não-ordinais, não sendo possível apresentar a ele conjuntos de dados com atributos contínuos, por exemplo. Nesse caso, os atributos contínuos devem ser previamente discretizados. Além dessa limitação, o ID3 também não apresenta nenhuma forma para tratar valores desconhecidos, ou seja, todos os exemplos do conjunto de treinamento devem ter valores conhecidos para todos os seus atributos.

C4.5

O C4.5 é um classificador estatístico baseado no algoritmo ID3 usado no aprendizado de máquina. Ele trabalha com o conceito de entropia da informação. Os dados de treinamento são um conjunto de amostras classificadas com vetores p-dimensionais que definem os atributos da amostra para gerar uma árvore de decisão em que cada nó divide as classes com base no ganho de informações. O atributo com o maior ganho normalizado de informações é usado como critério de divisão. A publicação do algoritmo C4.5 foi realizada em 1987, tendo como desenvolvedor John Ross Quinlan. O algoritmo tem como objetivo gerar um modelo classificador na forma de uma árvore de decisão, apresentando dois estados durante o processo, os quais são: folha que indica um ponto no final da classificação, sendo atribuída a uma classe e nó de decisão, onde baseando-se no atributo em análise, poderá conter uma ramificação seguida de uma folha ou uma sub-árvore para cada possível valor encontrado na base (QUINLAN, 1993).

2.3.2 C5.0

Conforme mencionando anteriormente, o classificador C5.0 baseia-se no algoritmo C4.5 e atua dividindo um modelo com base no campo que fornece o ganho máximo de informações. Cada subamostra definida pela primeira divisão é então dividida novamente, normalmente baseado em um campo diferente, e o processo se repete até que as subamostras não possam mais ser divididas, no final, as divisões de nível mais baixo são reexaminadas e as que não contribuem significativamente para o valor do modelo são removidas. Como resultado o C5.0 prevê apenas um destino categórico para cada instância do modelo. Nesta

tarefa o C5.0 retorna uma melhor relação de exatidão, precisão e especificidade em relação a outros algoritmos que realizam a mesma função, conforme observado no comparativo de (AGAOGLU, 2016). Mas essa é apenas uma das funcionalidades do C5.0, outra não menos importante, e que em alguns estudos apresenta um melhor resultado, é a geração de um conjunto de regras para realizar previsões para registros individuais. Os conjuntos de regras são derivados de árvores de decisão e, de certa forma, representam uma versão simplificada das informações encontradas na árvore de decisão. Os conjuntos de regras costumam reter a maioria das informações importantes de uma árvore de decisão completa, mas com um modelo menos complexo. Devido à maneira como os conjuntos de regras funcionam, eles não têm as mesmas propriedades que as árvores de decisão. A diferença mais importante é que, com um conjunto de regras, mais de uma regra pode ser aplicada a qualquer registro específico ou nenhuma regra pode ser aplicada. Nos algoritmos C4.5 e C5.0, existem algumas melhorias em termos de manipulação do viés em relação a testes com muitos resultados, desempenho e remoção (AGAOGLU, 2016).

O ID3 desenvolvido em 1986 por Ross Quinlan, cria uma árvore de múltiplas vias, encontrando para cada nó, onde o recurso categórico é que produzirá o maior ganho de informações para destinos categóricos. As árvores crescem no tamanho máximo e, em seguida, geralmente é aplicada uma etapa de poda para melhorar a capacidade da árvore de generalizar para dados não vistos, porém uma desvantagem no uso do algoritmo ID3 é o fato de que, ele utiliza apenas dados categóricos. O C4.5, sucessor do ID3, removeu a restrição de que os recursos devem ser categóricos, definindo dinamicamente um atributo discreto (com base em variáveis numéricas) que particiona o valor do atributo contínuo em um conjunto discreto de intervalos. O C5.0 sucessor do C4.5 por sua vez, incorporou algumas melhorias e converte as árvores treinadas em conjuntos de regras “se - então”, onde a precisão de cada regra é avaliada para determinar a ordem em que elas devem ser aplicadas, e realiza a remoção a partir da pré-condição de uma regra, se a precisão melhorar sem ela. CART (Árvores de classificação e regressão) é muito semelhante, porém o C5.0 o supera, devido ao fato de suportar variáveis de destino numéricas (regressão) e computar conjuntos de regras. O CART constrói árvores binárias usando o atributo e o limiar que produz o maior ganho de informações em cada nó.

O C5.0 incorpora várias melhorias em relação aos demais algoritmos, isso foi o que motivou a sua aplicação nesse estudo, pois na versão anterior, C4.5, todos os erros são tratados como iguais, mas em aplicações práticas alguns erros de classificação são mais graves que outros. C5.0 permite que um custo separado seja definido para cada par de classes

previsto; se essa opção for usada, o C5.0 criará classificadores para minimizar os custos esperados de classificação incorreta, em vez de taxas de erro (RULEQUEST, 2019). Os casos em si também podem ter uma importância desigual. O C5.0 possui um atributo de ponderação de caso que quantifica a importância de cada caso; se isso aparecer, o C5.0 tentará minimizar a taxa de erro preditivo ponderada. O C5.0 possui vários novos tipos de dados, além dos disponíveis no C4.5, incluindo datas, horas, atributos discretos ordenados e rótulos de maiúsculas e minúsculas. Além dos valores ausentes, o C5.0 permite que os valores sejam marcados como não aplicáveis. Além disso, o C5.0 fornece recursos para definir novos atributos como funções de outros atributos (RULEQUEST, 2019), conforme observado no exemplo de uma saída do classificador na Figura 9.

Figura 9 - Árvore de decisão gerada pelo classificador C5.0

```
Decision tree:

CO2 <= 640.75:
:...CO2 <= 470.75: 0 (770/3)
:   CO2 > 470.75:
:   :...HumidityRatio <= 0.003694578:
:       :...CO2 > 570.25: 1 (12)
:       :   CO2 <= 570.25:
:       :   :...CO2 > 535.5:
:       :       :...HumidityRatio <= 0.003613967: 1 (5)
:       :       :   HumidityRatio > 0.003613967: 0 (33/10)
:       :       CO2 <= 535.5:
:       :       :...HumidityRatio > 0.003579683: 0 (40)
:       :           HumidityRatio <= 0.003579683:
:       :           :...HumidityRatio <= 0.003452951:
:       :               :...CO2 <= 472.25: 0 (4/1)
:       :               :   CO2 > 472.25: 1 (4)
:       :               HumidityRatio > 0.003452951:
:       :               :...HumidityRatio <= 0.003548027:
:       :                   :...CO2 <= 498.25: 0 (72/4)
:       :                   :   CO2 > 498.25:
:       :                   :   :...HumidityRatio <= 0.003512921: 1 (4)
:       :                   :       HumidityRatio > 0.003512921: 0 (8/1)
:       :                   HumidityRatio > 0.003548027:
:       :                   :...HumidityRatio > 0.00357688: 1 (4)
:       :                       HumidityRatio <= 0.00357688:
:       :                       :...HumidityRatio <= 0.003557659: 1 (6/1)
:       :                           HumidityRatio > 0.003557659: 0 (19/1)
```

Fonte: AUTOR (2019)

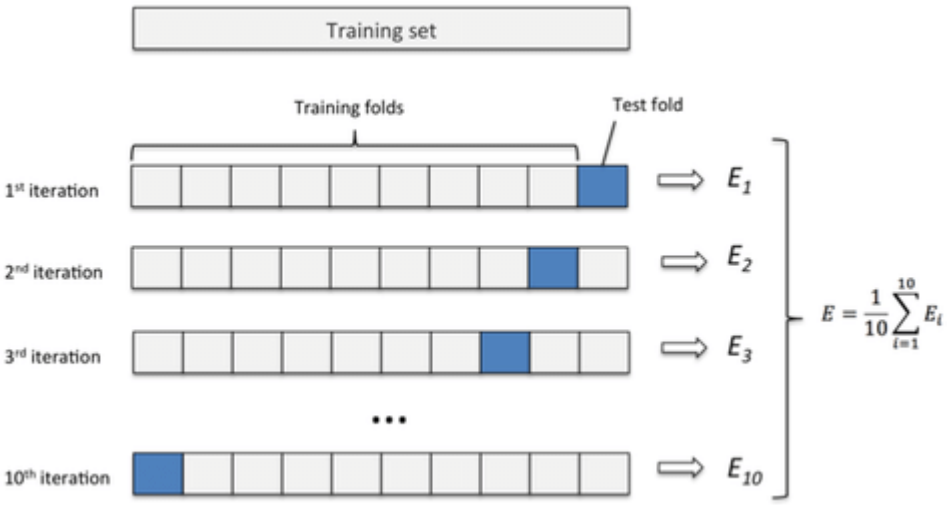
2.4 Avaliação do Modelo de Classificação

Após a determinação da base de dados que será utilizada, seguido pela preparação dos dados e escolha do algoritmo de classificação que se aplica da melhor forma para a tarefa pretendida, inicia-se a avaliação do modelo de classificação da AM, para tanto, o modelo será gerado por meio do algoritmo C5.0, devido às vantagens supracitadas.

Validação Cruzada

Segundo Carvalho (2014), “a técnica de validação cruzada é utilizada para verificar se o conjunto de dados de treinamento está representativo o suficiente em relação à base de dados na qual se deseja predizer certas instâncias”, ou seja, com esse procedimento o modelo generaliza como o classificador se comporta quando vai prever a classe de uma instância de dado que nunca viu. Essa é, portanto, uma das melhores técnicas para saber se o seu modelo generaliza bem e para criar diferentes conjuntos de treino e teste, com a finalidade de treinar o modelo e ter certeza de que ele está com uma boa performance. Nesse caso, ao invés de usarmos apenas um conjunto de teste para validar o modelo, utilizaremos N outros a partir dos mesmos dados. Existem diferentes métodos de validação cruzada, entre eles pode ser destacado o método denominado k-*fold*, que consiste em dividir o conjunto total de dados em k subconjuntos mutuamente exclusivos do mesmo tamanho e, a partir daí, um subconjunto é utilizado para teste e os k-1 restantes são utilizados para estimação dos parâmetros, fazendo-se o cálculo da acurácia do modelo. Este processo é realizado k vezes alternando de forma circular o subconjunto de teste. A Figura 10 mostra o esquema realizado pelo k-*fold*. Ao final das k iterações calcula-se a acurácia sobre os erros encontrados, através da equação descrita anteriormente, obtendo assim uma medida mais confiável sobre a capacidade do modelo de representar o processo gerador dos dados.

Figura 10 - Validação Cruzada k-fold



Fonte: RESEARCHGATE (2019)

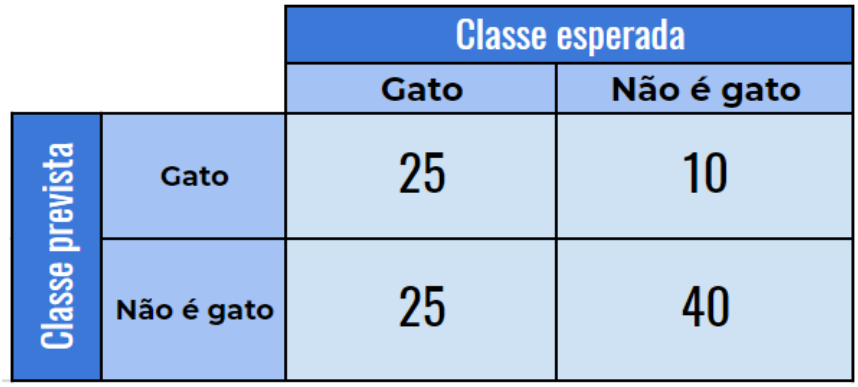
O método hold-out assemelha-se com o K-*fold* onde o k=2, porém com uma particularidade, a base de dados é dividida em duas partes, com isso uma das partes é usada para treino e a outra parte para teste, sem a alternação que ocorre com o k-*fold*. Este processo é realizado uma vez apenas, diferente do processo de K-*fold* em que os dados são divididos em K partes, e cada parte é usada tanto para treino como para teste, de tal forma que todas as partes passem por ambos os lados. Uma vantagem do modelo hold-out é que o tempo necessário para aprender o modelo é relativamente menor do que o tempo necessário para a aprendizagem do modelo usando a validação cruzada k-*fold* (YADAV E SHUKLA, 2016).

Matriz de Confusão

A técnica de Matriz de Confusão será um dos métodos utilizados neste trabalho para avaliar a precisão do algoritmo. A tabela de confusão, como também é conhecida, é uma representação voltada para modelos de classificação e tem como objetivo calcular a quantidade de falso positivo e falso negativo, verdadeiro positivo e verdadeiro negativo, A Matriz de Confusão fornece uma estimativa mais detalhada, em relação à classificação incorreta entre classes. A Figura 11 mostra como é formada uma matriz de confusão para um conjunto de classificação contendo duas classes-alvo, gato e não gato.

É com base na matriz de confusão, que se torna possível a especificação de quantas instâncias foram incorretamente classificadas e quantas foram classificadas sem erros. Uma matriz de confusão para N classes-alvo é tamanho N x N: como o exemplo mostrado na Figura 10 contém apenas duas classes-alvo, a matriz possui tamanho 2 x 2, sendo quatro valores possíveis, quantas instâncias da classe gato foram corretamente classificadas (chamados verdadeiros positivos, ou VP), quantas instâncias da classe gato foram incorretamente classificadas na classe não gato (falso positivo, ou FP), quantas instâncias da classe não gato foram corretamente classificadas (verdadeiro negativo, ou VN) e quantas instâncias foram incorretamente classificadas na classe gato (falso negativo, ou FN). Neste sentido, se torna compreensível que a matriz de confusão é uma excelente ferramenta para verificar a exatidão de um algoritmo, fornecendo maior detalhamento da estimativa de erro obtida pela Validação Cruzada. Quando usadas em conjunto, oferecem um excelente material para que se possa trabalhar na otimização do modelo de AM (CARVALHO, 2014).

Figura 11 - Matriz de Confusão



Fonte: PAULO VASCONCELLOS (2019)

Accuracy

A accuracy (acurácia) é uma das tantas métricas disponíveis atualmente para avaliação de um modelo de classificação. De forma genérica a acurácia de um modelo é obtida da razão entre as previsões que o modelo acertou e todas as previsões feitas. Para a classificação binária, a accuracy pode ser calculada em termos de positivos e negativos da seguinte maneira:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (5)$$

Onde TP = Verdadeiros Positivos, TN = Verdadeiros Negativos, FP = Falsos Positivos e FN = Falsos Negativos. Com o uso dessa métrica precisão pode chegar a 0,91, ou 91% (91 previsões corretas de um total de 100 exemplos). Isso significa que o classificador está fazendo um ótimo trabalho na identificação, porém, em uma análise mais próxima dos pontos positivos e negativos para obter mais informações sobre o desempenho do modelo, suponha-se que dos 100 exemplos de tumores, 91 são benignos (90 TNs e 1 FP) e 9 são malignos (1 TP e 8 FNs). Dos 91 tumores benignos, o modelo identifica corretamente 90 como benignos, isso é bom, porém, dos 9 tumores malignos, o modelo apenas identifica corretamente 1 como maligno - um resultado terrível, pois 8 em cada 9 malignidades não são diagnosticadas, embora 91% de precisão possa parecer boa à primeira vista. Em outras palavras, essa métrica não é a mais indicada para a tarefa de detecção de ocupação (GOOGLE, 2019). A accuracy por si só não é suficientemente boa quando se trabalha com um conjunto de dados

desequilibrado de classe, como este, onde há uma disparidade significativa entre o número de rótulos positivos e negativos. portanto são propostas outras métricas como as destacadas a seguir.

Precision

No reconhecimento de padrões, recuperação de informações e classificação binária, precision (precisão), que também é conhecida como valor preditivo positivo, é a fração de instâncias relevantes entre as instâncias recuperadas, enquanto recall, também conhecido como sensibilidade, é a fração de instâncias relevantes que foram recuperadas sobre o total da quantidade de instâncias relevantes. Precision e recall são, portanto, baseados em uma compreensão e medida de relevância. Em uma tarefa de classificação, a precisão de uma classe é o número de verdadeiros positivos (ou seja, o número de itens corretamente rotulados como pertencentes à classe positiva) dividido pelo número total de elementos rotulados como pertencentes à classe positiva (ou seja, a soma de positivos verdadeiros e falsos positivos, que são itens incorretamente rotulados como pertencentes à classe). Nesse contexto, é definido como o número de verdadeiros positivos dividido pelo número total de elementos que realmente pertencem à classe positiva (ou seja, a soma dos verdadeiros positivos e falsos negativos, que são itens que não foram identificados como pertencentes à classe positiva, mas deveria ter sido).

$$Precision = \frac{TP}{(TP + FP)} \tag{6}$$

Recall

A definição precisa de recall é o número de verdadeiros positivos dividido pelo número de verdadeiros positivos mais o número de falsos negativos. Verdadeiros positivos são pontos de dados classificados como positivos pelo modelo que são realmente positivos (o que significa que estão corretos) e falsos negativos são pontos de dados que o modelo identifica como negativos que são realmente positivos (incorretos).

$$Recall = \frac{TP}{(TP + FN)} \tag{7}$$

F1 Score

Essa é uma medida geral da precisão de um modelo que combina Precision e Recall, um bom resultado para F1 Score significa que se tem baixos falsos positivos e baixos falsos negativos. Portanto o resultado para a métrica F1 Score é considerada perfeita quando é 1, enquanto o modelo é considerado ruim quando é igual ou próximo de 0. F1 Score é a média ponderada de precisão e recall, por isso essa pontuação leva em consideração tanto os falsos positivos quanto os falsos negativos. Intuitivamente, essa medida é geralmente mais útil que Accuracy, especialmente quando se tem uma distribuição de classe desigual. A accuracy funciona melhor se os falsos positivos e falsos negativos tiverem um custo semelhante. Se o custo de falsos positivos e falsos negativos for muito diferente, é melhor olhar para Precision e Recall. Quando a F1 Score é aplicada como métrica, se seu resultado for alto, a precision e o recall do classificador indicarão bons resultados. Essa característica da métrica nos permite comparar o desempenho de dois classificadores usando apenas uma métrica e ainda assim ter certeza de que os classificadores não estão cometendo erros que são despercebidos pelo código que pontua sua saída.

$$F1\ Score = \frac{2 * (Recall\ * Precision)}{(Recall + Precision)} \tag{8}$$

Kappa

A medida estatística Kappa é uma métrica que compara uma precisão observada com uma precisão esperada (chance aleatória). A estatística Kappa é usada não apenas para avaliar um único classificador, mas também para avaliar os classificadores entre si. Além disso, leva em consideração a chance aleatória (concordância com um classificador aleatório), o que geralmente significa que é menos enganoso do que simplesmente usar a precisão como métrica.

$$\widehat{K} = \frac{\hat{P}0 - \hat{P}e}{1 - \hat{P}e} \tag{9}$$

O cálculo da precisão observada e da precisão esperada é essencial para a compreensão da estatística Kappa e é mais facilmente ilustrada pelo uso de uma matriz de

confusão. Outro ponto importante que diferencia essa medida das demais é fato de que os resultados obtidos podem ser comparados com os de uma tabela que pode ser utilizada para interpretar os valores, como observado na Tabela 1.

Tabela 1 - Qualidade de classificação Kappa

Valor kappa	Grau de concordância
<0,00	Ruim
0,00 - 0,20	Fraca
0,21 - 0,40	Razoável
0,40 - 0,60	Boa
0,60 - 0,80	Muito boa
0,80 - 1,00	Excelente

Fonte: Desenvolvido pelo autor.

3 MATERIAIS E MÉTODOS

3.1 Descrição do Objeto de Estudo

Conjunto de Dados

Três conjuntos de dados foram usados nesta pesquisa para treinar e testar o modelo de classificação, os conjuntos de dados foram gerados originalmente pela pesquisa de Candanedo e Feldheim (2016), eles estão resumidos na Tabela 2. Para todos os conjuntos de dados, são definidos sete atributos, a saber: umidade, taxa de umidade relativa (atributo derivado gerado pela divisão umidade/temperatura), luz, CO_2 e temperatura, além do status da ocupação (0 para não ocupado, 1 para ocupado), marcada como variável alvo, e a marcação de data/hora. As distribuições de classe também estão expostas na mesma tabela.

As bases de dados empregadas no estudo foram extraídas de um diretório disponibilizado na internet, em github.com (um sistema de controle de versão de arquivos através do qual se pode desenvolver projetos em que diversas pessoas podem contribuir simultaneamente). Os dados contidos nas bases foram obtidos em uma sala de escritório com dimensões aproximadas de 5,85m (largura) x 3,5m (comprimento) x 3,53m (altura). Na pesquisa que deu origem as respectivas bases de dados, uma câmera digital foi usada para

determinar se a sala estava realmente ocupada ou não, todavia dados obtidos a partir do uso da câmera não compõem os atributos preditores, estes por sua vez são gerados a partir dos dados de outros sensores apenas. Nesta abordagem pretende-se conhecer o comportamento de um determinado algoritmo classificador em relação aos conjuntos de dados propostos. No estudo do qual as bases de dados tiveram origem, para estimar a diferença na precisão da detecção de ocupação fornecida pelos modelos, eles foram testados quando a porta do escritório estava aberta e fechada, no estudo atual, essa abordagem não será aplicada. As leituras foram registradas em intervalos de 14 s ou 3 a 4 vezes por minuto e depois calculadas a média do minuto correspondente. Os sensores foram colocados em uma mesa, como mostra a Figura 12. A distância para o ocupante mais próximo foi de 1,1 m para o segundo ocupante, cerca de 2,9 m (CANDANEDO e FELDHEIM, 2016).

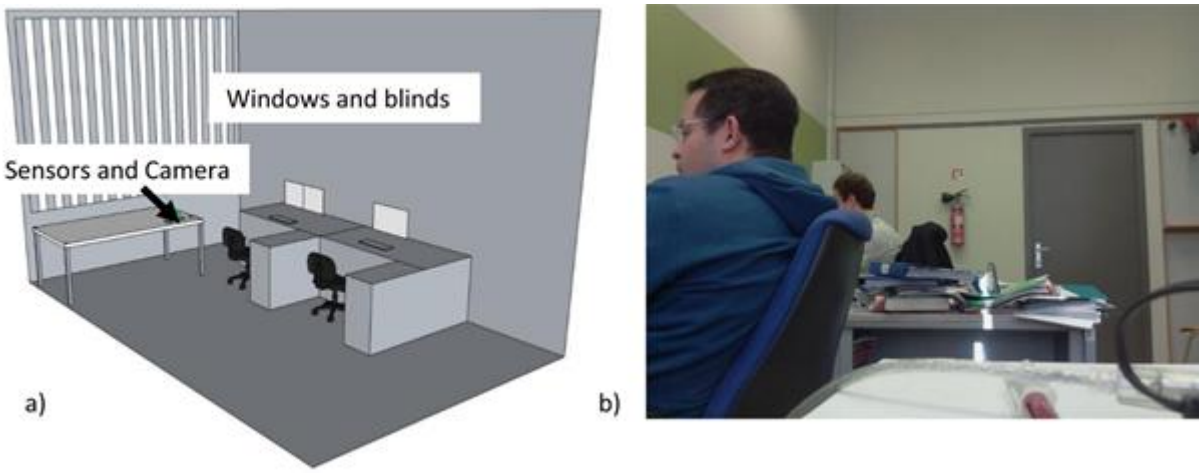
Tabela 2 - Descrição do conjunto de dados

Conjunto de dados	**Número de casos**	**Atributos**	**Distribuição da classe de dados (%)**	
			0 (não ocupado)	**1 (ocupado)**
Treinamento	8143	7	79%	21%
Teste 1	2665	7	64%	36%
Teste 2	9752	7	79%	21%

Fonte: Desenvolvido pelo autor.

Nos ensaios realizados neste estudo, os conjuntos de dados oriundos de fontes secundárias, foram divididos em três diferentes cenários, em que os dois primeiros foram realizados apenas para permitir a comparação dos resultados com um dos trabalhos relacionados e o último foi delineado com o objetivo de obter a maior confiabilidade nos resultados gerados. Em resumo, descrevemos os conjuntos de dados destes cenários, que emprega a mesma proporção das classes dos conjuntos de dados originais: cenário 1 (um) contendo um conjunto de dados para teste com 2.665 caos e um conjunto de treinamento com 8.145 casos, tendo sido realizado uma validação cruzada do tipo holdout (da mesma forma que ocorreu no estudo que deu origem as bases de dados). O cenário 2 (dois) com um total de 9.754 casos na base de teste, confrontando com o mesmo conjunto de treinamento de 8.145 casos, tendo sido realizado uma validação cruzada do tipo holdout. Já no cenário 3 (três) foi realizada uma junção de todos os dados coletados totalizando (2.665 + 8.145 + 9754 = 20.564) casos em que se aplicou uma validação cruzada 10-*fold*.

Figura 12 - (a) Esboço da sala mostrando a posição dos sensores e a posição dos ocupantes, (b) Exemplo de uma das imagens da câmera digital usada para estabelecer a ocupação da sala



Fonte: CANDANEDO e FELDHEIM (2016)

O algoritmo Classificador

Para finalidade proposta são empregados os seguintes parâmetros: O boosting (onde o classificador melhora os resultados ‘observando’ os erros da árvores anteriores) para treinar com 3, 15, 30 e 50 tentativas ou ciclos, A profundidade da poda foi marcada com os valores 5, 15, 25 (padrão) e 36, o parâmetro -g que desativa a poda global também foi utilizado no ensaio, além dos padrões para definição de caso mínimo de criação de uma nova folha que foi marcada com os valores 2, 4 e 8 e no final de cada rodada de testes foi proposto o ensaio sinalizando a quantidade de ciclos de boosting para 50, a profundidade da poda para 5 e os padrões para criar uma nova folha sinalizado com o valor 8, para os três cenários, formando a combinação de parâmetros -t 50 -c 5 -m 8. A tabela 3 exibe uma relação de parâmetros utilizados neste estudo, que foram selecionados em uma lista disponibilizada na documentação do classificador disponível em (RULEQUEST, 2019).

Para o correto funcionamento do sistema, faz-se necessário a configuração dos arquivos de entrada do algoritmo classificador. Nesse estudo, dois arquivos foram essenciais para o funcionamento do C5.0, “nomeDaAplicação”.data (recebe os dados) e “nomeDaAplicação”.names (contém os nomes dos atributos, seus respectivos tipos, assim como a determinação de qual deles é o atributo alvo), nesta etapa foi necessário a exclusão do atributo id em todos os arquivos “nomeDaAplicação”.names, nos três cenários por meio da adição da seguinte linha no arquivo "nomeDaAplicação".names, attributes excluded: id.

Tabela 3 - Lista de parâmetros utilizados

Parâmetro	Descrição	Valores
-t	Define o boosting com uma quantidade específica de tentativas	3, 15, 30, 50
-c	Define a profundidade da poda	5, 15, 25 (padrão), 36
-g	Desativa a poda global	-
-m	Define a quantidade de ramificações de uma árvore	2 (padrão), 4, 8

Fonte: Desenvolvido pelo autor.

3.2. Delineamento da Pesquisa

O sentido empírico desse estudo tem como propósito a busca de informações relevantes e convenientemente obtidos através da experiência, e dos ensaios de realizados em outros estudos. Desta forma este estudo objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos, envolve verdades e interesses locais se consolidando com uma pesquisa de natureza aplicada, com uma forma de abordagem quantitativa. Quanto aos objetivos é uma pesquisa exploratória, e enquanto análise tem como finalidade proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito constituído com auxílio de levantamento bibliográfico, análise de exemplos que estimulem a compreensão (GIL, 2008).

3.3 Procedimentos Específicos

O levantamento foi ambientado em um computador do tipo laptop configurado com os seguintes dispositivos: processador Intel Core i5-4200U CPU @ 1.60 GHz x 4, memória ram de 8,00 GB, armazenamento de 1 TB, sistema operacional Linux Ubuntu 18.04.1 LTS, ambiente de trabalho GNOME versão 3.28.2 e arquitetura 64 bits. O algoritmo empregado neste estudo para a construção de árvores de decisão é o C5.0, edição 2.07, GPL Edition, disponível para download em (RULEQUEST, 2019), que nesse estudo é aplicado na construção de árvores de decisão. A versão descrita do algoritmo foi compilada no terminal Linux utilizando o compilador GCC versão 7.4.0. Os resultados obtidos da classificação são disponibilizados em uma matriz de confusão, conforme exemplo exposto na Tabela 4.

Tabela 4 - Matriz de confusão gerada pelo C5.0

(a)	(b)	<-classified as
6047	367	(a): class 0
521	1208	(b): class 1

Fonte: Desenvolvido pelo autor.

Através dos resultados obtidos pelo classificador, e da matriz de confusão, os resultados gerados foram agrupados em um software gerador de planilhas eletrônicas onde foram empregadas as seguintes métricas que foram utilizadas a fim de validar os resultados obtidos: Accuracy, Precision, Recall, F1 Score e Kappa. Os valores retornados por essas métricas ficam entre 0 e 1, onde valores menores indicam um pior desempenho, e menor exatidão em determinar se o ambiente está ocupado por pessoas. Conforme visto na seção 2.4, valores entre 0,80 e 1,00 do coeficiente Kappa indicam grau de concordância excelente entre o modelo gerado e os valores reais (aqueles anotados na base por meio do uso da câmera). Os valores de referência para essa métrica podem ser observados na Tabela 1.

3.4 Configurações do Classificador

Nos cenários 1 e 2 foram utilizados os valores para os parâmetros conforme descritos na tabela 6 e no cenário 3 foram utilizados os mesmos parâmetros, porém combinados com a validação cruzada em 10-*folds* (-X 10). O método de Validação Cruzada que foi utilizado no cenário 3 foi o método de k-partições. Como o objetivo do trabalho consiste em analisar os resultados obtidos principalmente na base de dados do cenário 3, cenário montado para ensaios deste estudo, todas as instâncias já estarão classificadas, como nas bases de dados candidatas analisadas anteriormente. Assim sendo, a base de dados será dividida em 10 partições diferentes, cada partição com o mesmo tamanho (caso o número de instâncias seja divisível por 10. Caso contrário, as instâncias restantes serão distribuídas pelas partições). Estas partições então serão rotuladas com valores de 1 a 10. Então por 10 iterações, cada uma das partições irá ser o conjunto de casos de teste enquanto as 9 restantes irão ser o conjunto de casos de treinamento. A cada iteração realizada, a estimativa de erro do modelo, que consiste em quantas classificações incorretas foi feita no conjunto de casos de teste, será calculada. A estimativa total do erro do modelo será a soma das estimativas obtidas em cada iteração dividida por 10. O parâmetro k=10 foi escolhido porque segundo Witten et al.,

(2005), testes extensivos em vários conjuntos de dados, com diferentes técnicas de aprendizagem, mostraram que 10 é o número certo para obter a melhor estimativa de erro, e também há algumas evidências teóricas que apoiam isso. Todavia ainda há bastante discussão quanto ao melhor parâmetro k, porém, na prática, o k=10 se tornou o método padrão (WITTEN et al., 2005).

Tabela 5 - Parâmetros e valores

Parâmetros	**Valores**
-f	-
-t	3, 15, 30, 50
-c	5, 15, 25, 36
-g	-
-m	2, 4, 8

Fonte: Desenvolvido pelo autor.

Tabela 6 - Sumarização das configurações

Cenário	**Configurações utilizadas**	**Número de casos**	**Tipo de validação**
1	-t 3, -t 15, -t 30, -t 50	2.665	holdout
	-c 5, -c 15, -c 25, -c 36		
	-g		
	-m 2, -m 4, -m 8		
	- t 50, -c 5, -m 8		
2	-t 3, -t 15, -t 30, -t 50	9.754	holdout
	-c 5, -c 15, -c 25, -c 36		
	-g		
	-m 2, -m 4, -m 8		
	- t 50, -c 5, -m 8		
3	-t 3, -t 15, -t 30, -t 50	20.564	10-*folds* (-X 10)
	-c 5, -c 15, -c 25, -c 36		
	-g		
	-m 2, -m 4, -m 8		
	- t 50, -c 5, -m 8		

Fonte: Desenvolvido pelo autor.

4 RESULTADOS E DISCUSSÕES

4.1 Cenário 1

Para análise dos resultados deste trabalho, foi escolhida a métrica F1-score, descrita na seção 2.4, por ser uma das métricas mais utilizadas na literatura e por considerarmos que realiza uma boa síntese do desempenho do classificador. Neste cenário 1, observamos que as taxas de acerto obtidas não foram satisfatórias o bastante, uma vez que a média de erros chegou a 6,5%. Uma das possíveis causas do desempenho desse cenário ter sido inferior às dos outros dois cenários, é o fato de quanto menos dados o conjunto tiver, os resultados serão menos expressivos, pois o desempenho do conjunto de treinamento com 8.143 no mesmo cenário obteve precisão de 100%, e no conjunto de testes com apenas 2.665, só conseguiu alcançar resultados aceitáveis quando foi empregado o hiperparâmetro -t, que define o boosting com uma quantidade específica de tentativas. O hiperparâmetro que retornou o melhor F1 Score foi -t 3, com um valor que corresponde a 97,10%, com apenas três tentativas de boosting, que tem como finalidade gerar vários modelos, também chamados de ensaios, ao invés de apenas um modelo. O parâmetro trial (-t) controla o número de vezes que serão gerados os ensaios, sendo que para cada construção do modelo é dada mais atenção às regras de classificação com maiores taxas de erros, tentando melhorá-las no próximo ensaio (Quinlan, 2013).

Neste cenário foi empregada a validação cruzada holdout, portanto o fato dessa forma de validação não ser a mais segura para comprovar a eficiência desse tipo de modelo, faz com que esse cenário não seja o mais apropriado para a validação dos resultados dessa pesquisa, apesar de serem encontradas respostas satisfatórias conforme disponível na Tabela 7. Entre as vantagens do uso desse tipo de validação cruzada está o fato de ela ser muito simples e que se torna necessário para treinar apenas um modelo. Se o desempenho for bom o suficiente, podemos prosseguir e usá-lo em qualquer aplicativo que pretendamos. No entanto para o conjunto de dados alvo desse estudo isso é não é muito adequado uma vez que o conjunto de dados não está relativamente uniforme em termos de distribuição das classes, conforme observado na Tabela 2. De acordo com o que pode ser observado na Tabela 7, os hiperparâmetros que não podem ser aprendidos diretamente do processo regular de treinamento e expressam propriedades de nível superior do modelo, como sua complexidade ou a rapidez com que ele deve aprender, e que neste cenário apresentaram os melhores resultados como o hiperparâmetro -t 3 citado anteriormente, e a combinação de parâmetros

com -t 50 -c 5 -m 8 com uma precisão igual a 97,10% e 97%. Observa-se também que neste cenário os melhores resultados foram obtidos com o auxílio do processo de boosting.

Tabela 7 - Resultados obtidos pelo modelo no cenário 1

Parâmetros	TP	FN	FP	TN	Erros	Accuracy	Precision	Recall	F1 score	Kappa
- f	1.645	48	165	807	8,0%	0,920	0,909	0,972	0,939	0,823
-t 3	1.640	53	45	927	3,7%	0,963	0,973	0,969	0,971	0,921
-t 15	1.642	51	60	912	4,2%	0,958	0,965	0,970	0,967	0,910
-t 30	1.641	52	60	912	4,2%	0,958	0,965	0,969	0,967	0,909
-t 50	1.641	52	62	910	4,3%	0,957	0,964	0,969	0,966	0,907
-c 25 (padrão)	1.645	48	165	807	8,0%	0,920	0,909	0,972	0,939	0,823
-c 5	1.645	48	165	807	8,0%	0,920	0,909	0,972	0,939	0,823
-c 15	1.645	48	165	807	8,0%	0,920	0,909	0,972	0,939	0,823
-c 36	1.645	48	165	807	8,0%	0,920	0,909	0,972	0,939	0,823
-g	1.645	48	165	807	8,0%	0,920	0,909	0,972	0,939	0,823
-m 2 (padrão)	1.645	48	165	807	8,0%	0,920	0,909	0,972	0,939	0,823
-m 4	1.644	49	137	835	7,0%	0,930	0,923	0,971	0,946	0,846
-m 8	1.643	50	147	825	7,4%	0,926	0,918	0,970	0,943	0,837
-t 50 -c 5 -m 8	1.642	51	52	920	3,9%	0,961	0,969	0,970	0,970	0,917
Menor resultado alcançado					**3,7%**	**0,920**	**0,909**	**0,969**	**0,939**	**0,823**
Maior resultado alcançado					**8,0%**	**0,963**	**0,973**	**0,972**	**0,971**	**0,921**

Fonte: Desenvolvido pelo autor.

4.2 Cenário 2

No ambiente de ensaios do cenário 2, que foi preparado com 9.752 casos de testes e 8.145 de treinamento e com todos os atributos dos conjuntos de dados, o classificador conseguiu obter bons resultados em relação ao conjunto de dados proposto. No atual cenário a métrica F1 Score retornou um valor correspondente a 99,51% com o parâmetro de boosting o hiperparâmetro -t 3 sendo considerada a melhor resposta do cenário com o cálculo realizado por meio da referida métrica, uma hipótese para esse resultado pode ser o fato de que a validação holdout é a mais adequada para utilização em combinação com o parâmetro de boosting. É importante ressaltar que, quando os resultados obtidos a partir da matriz de confusão gerada pelo classificador foram calculados usando a métrica Precision, neste cenário foi retornado uma precisão elevada equivalente a 99,83% também com o parâmetro de entrada -t 3, a baixa quantidade de classificações errôneas favoreceu esse resultado, conforme listado na Tabela 8, que por sua vez pode ser considerado a melhor resposta na precisão de detecção de ocupação usando como base os dados utilizados nesta pesquisa. O

parâmetro de entrada -t 50 -c 5 - m 8 retornou à segunda melhor resposta para este cenário, equivalente a 98,68% de precisão quando utilizada a métrica F1 Score, uma possível causa nessa taxa de acerto elevada se dá devido ao fato dessa combinação de hiperparâmetros gerar uma árvore menor e por conseguinte menos propensa a erros, conforme pode ser observado na Tabela 8 a taxa de erros média chegou a apenas 2,1% com essa combinação de parâmetros. De forma geral a taxa de erros para esse cenário variou em torno dos 5,5%, sendo considerada elevada se levada em consideração todos os parâmetros utilizados neste estudo.

Tabela 8 - Resultados obtidos pelo modelo no cenário 2

Parâmetros	**TP**	**FN**	**FP**	**TN**	**Erros**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**Kappa**
- f	7.330	373	347	1.702	7,4%	0,926	0,955	0,952	0,953	0,779
-t 3	7.640	63	13	2.036	0,8%	0,992	0,998	0,992	0,995	0,977
-t 15	7.497	206	27	2.022	2,4%	0,976	0,996	0,973	0,985	0,930
-t 30	7.442	261	29	2.020	3,0%	0,970	0,996	0,966	0,981	0,914
-t 50	7.410	293	39	2.010	3,4%	0,966	0,995	0,962	0,978	0,902
-c 25 (padrão)	7.330	373	347	1.702	7,4%	0,926	0,955	0,952	0,953	0,779
-c 5	7.330	373	347	1.702	7,4%	0,926	0,955	0,952	0,953	0,779
-c 15	7.330	373	347	1.702	7,4%	0,926	0,955	0,952	0,953	0,779
-c 36	7.330	373	347	1.702	7,4%	0,926	0,955	0,952	0,953	0,779
-g	7.330	373	347	1.702	7,4%	0,926	0,955	0,952	0,953	0,779
-m 2 (padrão)	7.330	373	347	1.702	7,4%	0,926	0,955	0,952	0,953	0,779
-m 4	7.326	377	304	1.745	7,0%	0,930	0,960	0,951	0,956	0,792
-m 8	7.330	373	318	1.731	7,1%	0,929	0,958	0,952	0,955	0,789
-t 50 -c 5 -m 8	7.539	164	37	2.012	2,1%	0,979	0,995	0,979	0,987	0,939
Menor resultado alcançado					**0,8%**	**0,926**	**0,955**	**0,951**	**0,953**	**0,779**
Maior resultado alcançado					**7,4%**	**0,992**	**0,998**	**0,992**	**0,995**	**0,977**

Fonte: Desenvolvido pelo autor.

O menor valor obtido neste cenário foi de 95,32% de acordo com o que pode ser visto na coluna F1 Score com o hiperparâmetro especificado na entrada -c, que define a profundidade da poda, seguidos pelos respectivos argumentos. Os mesmos não retornaram uma boa resposta nesse conjunto de dados, mesmo com a alteração dos argumentos para esse parâmetro os resultados alcançados permaneceram inalterados. Talvez pelo tipo de validação cruzada empregado, a validação cruzada holdout. Da mesma forma que ocorre no cenário 1 a tabela do cenário 2, na primeira coluna estão os parâmetros utilizados na classificação, na segunda coluna está a média de erros geradas pelo classificador, na terceira coluna estão os resultados encontrados com a métrica precision que para tanto utiliza a equação 6 do atual estudo. Na coluna intitulada Recall estão disponíveis os resultados desta métrica obtidos por

meio do uso da equação 7, e a última coluna representando os valores alcançados com a métrica F1 Score.

4.3 Cenário 3

No cenário 3, em que realiza-se uma validação cruzada 10-*fold*, considerada mais adequada no que diz respeito à confiabilidade da avaliação dos modelos de classificação gerados, a métrica F1 Score, que combina precisão e recall de modo a trazer um número único que indique a qualidade geral do modelo, trabalhando bem até com conjuntos de dados que possuem classes desbalanceadas, apresentou excelentes resultados neste ambiente experimental que combina as instâncias dos três conjuntos de dados, treinamento, teste 1 e teste 2 totalizando 20.560 instâncias. No ensaio representado na Tabela 9 onde foi empregada a validação cruzada com 10 interações, sendo a melhor quantidade que deve ser empregada para obtenção de uma estimativa de erro menor (WITTEN et. al., 2005), a métrica F1 Score teve uma pequena oscilação com a inserção de diferentes parâmetros com respectivos argumentos, a variação ficou entre 0,990 e 0,995, possivelmente por ser uma métrica que não tem relação direta com o desbalanceamento dos dados, retornou resultados considerados muito bom para essa medida. De forma geral o classificador C5.0 apresentou um bom desempenho em relação a base de dados usada o cenário 3.

Uma média de erro de apenas 0,9% foi alcançada nesse cenário, acredita-se que uma das causas do bom desempenho seja devido ao uso da validação cruzada k-*fold*, técnica estatística aplicada para testar o desempenho de um modelo de Machine Learning. Em particular, um bom método de validação cruzada fornece uma medida abrangente do desempenho do modelo em todo o conjunto de dados. Além do viés de seleção, a validação cruzada também ajuda a evitar o ajuste excessivo. A divisão de uma base de dados em um conjunto de treinamento e testes, conhecido como validação holdout, conforme foi proposto nos cenários 1 e 2 auxilia na correta verificação do modelo, se está tendo um bom desempenho nos dados vistos durante o treinamento ou não, entretanto entre as desvantagens no uso desse tipo de validação surgem quando o conjunto de dados não é totalmente uniforme. Ao dividir o conjunto de dados, podemos acabar dividindo-o de tal maneira que o conjunto de treinamento seja muito diferente do conjunto de teste, ou mais fácil ou mais difícil. Portanto, o teste único que é realizado com holdout não é abrangente o suficiente para avaliar adequadamente o modelo, acabando com classificações ruins, como super ajuste ou medições imprecisas do desempenho projetado no modelo. No cenário atual o valor máximo

alcançado para F1 Score foi 99,49%, obtido com o emprego dos parâmetros -t 30 – X 10, -t 50 -X 10 e -m 2 -X 10. Os conjuntos de dados foram divididos em dez conjuntos. 9 séries foram usadas para treinamento e 1 série para teste.

A grande vantagem que vem com a validação cruzada K-*fold* é que é muito menos propensa ao viés de seleção, pois o treinamento e o teste são realizados em várias partes diferentes. Em particular, se aumentarmos o valor de K, podemos ter ainda mais certeza da robustez do modelo, pois é treinado e testado em tantos sub-conjuntos de dados diferentes, entretanto o custo computacional torna-se excessivo. Outra característica importante observada na abordagem proposta neste cenário é que a validação cruzada contribui para diminuição do ajuste excessivo ou insuficiente de um modelo de decisão, a esse tipo de validação é usada como uma ferramenta de ajuste fino. No processo de validação cruzada com dobra 10, todos os dados são divididos em 10 dobras iguais (ou quase iguais). Os dados da primeira dobra são tratados como o conjunto de dados de validação e os outros, ou seja, dobras k-1, são considerados o conjunto de dados de treinamento para gerar a primeira árvore de decisão. Posteriormente, 10 iterações de treinamento e o processo de validação, por sua vez, são executados.

Tabela 9 - Resultados obtidos pelo modelo no cenário 3

Parâmetros	**TP**	**FN**	**FP**	**TN**	**Erros**	**Accuracy**	**Precision**	**Recall**	**F1 score**	**Kappa**
-X 10	15.675	135	53	4.697	0,9%	0,991	0,997	0,991	99,40%	0,974
-t 3 -x10	15.695	115	193	4.557	1,5%	0,985	0,988	0,993	99,03%	0,958
-t 15 - X10	15.688	122	56	4.694	0,9%	0,991	0,996	0,992	99,44%	0,976
-t 30 - X10	15.690	120	42	4.708	0,8%	0,992	0,997	0,992	99,49%	0,978
-t 50 - X10	15.690	120	40	4.710	0,8%	0,992	0,997	0,992	99,49%	0,978
-c 25 (padrão) -X10	15.686	124	50	4.700	0,8%	0,992	0,997	0,992	99,45%	0,976
-c 5 - X10	15.653	157	49	4.701	1,0%	0,990	0,997	0,990	99,35%	0,972
-c 15 -X10	15.668	142	50	4.700	0,9%	0,991	0,997	0,991	99,39%	0,974
-c 36 -X10	15.682	128	54	4.696	0,9%	0,991	0,997	0,992	99,42%	0,975
-g -X10	15.680	130	59	4.691	0,9%	0,991	0,996	0,992	99,40%	0,974
-m 2 -X10 (padrão)	15.693	117	52	4.698	0,8%	0,992	0,997	0,993	99,46%	0,977
-m 4 -X10	15.678	132	46	4.704	0,9%	0,991	0,997	0,992	99,44%	0,976
-m 8 -X10	15.676	134	62	4.688	1,0%	0,990	0,996	0,992	99,38%	0,973
-t 50 -c 5 -m 8 -X10	15.636	174	33	4.717	1,0%	0,990	0,998	0,989	99,34%	0,972
Menor resultado alcançado					**0,8%**	**0,985**	**0,988**	**0,989**	**0,990**	**0,958**
Maior resultado alcançado					**1,5%**	**0,992**	**0,998**	**0,993**	**0,995**	**0,978**

Fonte: Desenvolvido pelo autor.

4.4 Discussões

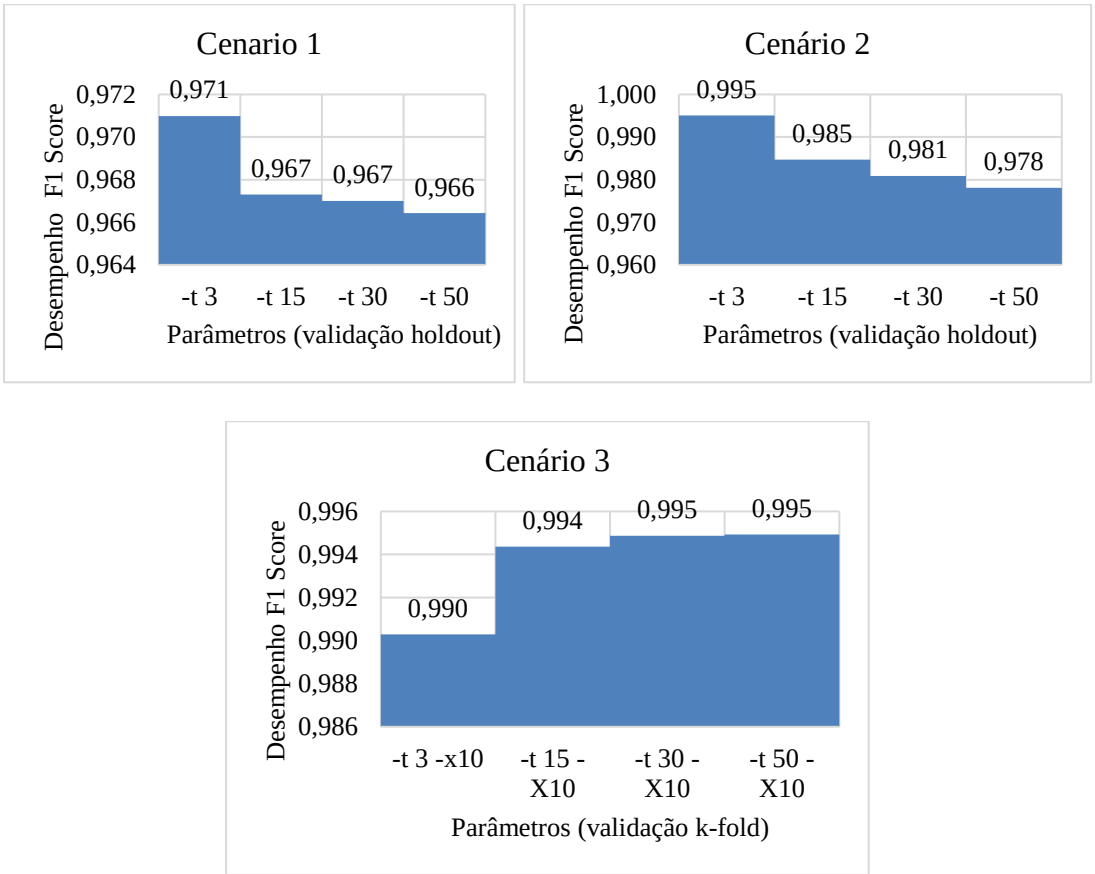
O estudo teve como objetivo fornecer regras de classificação com a finalidade de determinar a ocupação de uma sala de escritório de acordo com dados de sensores e obteve resultados compatíveis com os apresentados na literatura. Índices de precisão maiores ou menores que os relatados na literatura não representam respectivamente sucesso ou fracasso, mas sim o quão próximo do estado da arte o modelo pode determinar a presença de ocupantes dentro do recinto. O estudo não limitou o número de camadas e nós da árvore de decisão, de modo a representar o modelo completo e aumentar a exatidão dos resultados.

O C5.0 foi adotado como a ferramenta de classificação de dados para analisar 20.560 instâncias da base de dados do cenário 3, dos quais os três indicadores de desempenho foram utilizados para validar os resultados alcançados, entre eles, precision com 99,79% no cenário 3, e 99,83% (o maior resultado retornado nos três cenários) no cenário 2. Curiosamente, nos três cenários o parâmetro de entrada -c, que determina a profundidade da poda, com os valores dos argumentos definidos como 5, 15, 25 e 36 não retornou alterações expressivas no resultado da precisão medido pelo medidor F1 Score, pois as saídas permaneceram idênticas, o que pode indicar que o problema de sobreajustamento não foi tão relevante nos modelos gerados, uma vez que a poda visa a sua amenização; no cenário 3 por exemplo onde foi empregada a validação em 10 iterações, o valor retornado foi de 0,994 com uma pequena variação para 0,993 com argumento definido com valor 5.

Nos cenários 1 e 2 é possível detectar uma diminuição da precisão na proporção que o valor do argumento de entrada inserido no parâmetro -t, que determina a quantidade de tentativas de boosting, aumenta, isso pode ser notado através dos histogramas representados na Figura 12, uma das possíveis causas para esses resultados é divisão das instâncias dentro dos conjuntos de dados, onde instâncias que deveriam estar no conjunto de treinamento estão no conjunto de testes, favorecendo resultados menos concisos. A figura 12 aponta também que ocorre o contrário no cenário 3 onde o hiperparâmetro -t 50 -X 10 é o que retorna a melhor precisão, correspondente a 99,51%, essa melhora nas respostas do cenário 3 é ocasionada provavelmente devido ao emprego do método de validação cruzada k-*fold* que elimina a possibilidade de instâncias alocadas no conjunto de dados onde não possui representatividade, ocasionando diminuição da exatidão, como pode ter ocorrido nos cenários 1 e 2.

Para responder à pergunta de pesquisa Q1: É possível realizar a detecção de ocupação de escritórios utilizando um algoritmo de classificação baseado em árvores de decisão com desempenho melhor ou igual ao documentado na literatura? Nesta seção é realizada, uma comparação entre os resultados do algoritmo de aprendizado C5.0 e um conjunto de algoritmos de aprendizado supervisionado que foi utilizado no estudo de Candanedo e Feldheim (2016) e representada na Tabela 11 os algoritmos são Floresta Aleatória (RF) Máquinas de Reforço de Gradiente (GBM), Análise Discriminante Linear (LDA) e Árvores de Classificação e Regressão (CART). A Tabela 10 foi utilizada para relacionar os resultados obtidos pela métrica accuracy, utilizado como indicador de desempenho no estudo que deu origem aos respectivos conjuntos de dados, neste experimento foram utilizados os atributos umidade, luz, CO_2, umidade relativa e temperatura, conforme experimentos realizados no estudo de (CANDANEDO e FELDHEIM, 2016), para que a título de comparação o C5.0 seja colocado entre os algoritmos citados na pesquisa.

Figura 13 - Histogramas comparativo de desempenho do parâmetro - t (Boosting) em três cenários



Fonte: Desenvolvido pelo autor.

Na Tabela 11 é apresentado um comparativo dos resultados alcançados com os algoritmos utilizados no estudo de (CANDANEDO e FELDHEIM, 2016), e o C5.0, algoritmo utilizado na linha de pesquisa deste trabalho e numerado com o id 05 na tabela, que consegue resultado final igual ao algoritmo Floresta Aleatória (RF) para o conjunto de treinamento com 100% de precisão, superando classificadores tradicionais como GBM, CART e LDA. Com o conjunto de teste 1 obtido por meio do cenário 1, o C5.0 ficou em segundo lugar, com uma precisão de 96,32% ficando atrás do classificador LDA com uma precisão anotada de 97,90%. Empregando o conjunto de teste 2 o C5.0 superou os outros algoritmos com uma precisão relatada de 99,22% obtida a partir do cenário 2.

Tabela 10 - Resultados utilizando a métrica Accuracy

Parâmetros	**Métrica accuracy**		
	Treinamento	**Teste 1**	**Teste 2**
- f	0,993	0,920	0,926
-t 3	0,995	0,963	0,992
-t 15	1,000	0,958	0,976
-t 30	1,000	0,958	0,970
-t 50	1,000	0,957	0,966
-c 25 (padrão)	0,993	0,920	0,926
-c 5	0,991	0,920	0,926
-c 15	0,991	0,920	0,926
-c 36	0,993	0,920	0,926
-g	0,993	0,920	0,926
-m 2 (padrão)	0,993	0,920	0,926
-m 4	0,990	0,930	0,930
-m 8	0,985	0,926	0,929
-t 50 -c 5 -m 8	1,000	0,961	0,979
Menor	**0,985**	**0,920**	**0,926**
Maior	**1,000**	**0,963**	**0,992**

Fonte: Desenvolvido pelo autor.

O modelo classificador C5.0 obteve um excelente resultado, de forma geral, em relação ao desemprenho de outros classificadores utilizando os mesmos conjuntos de dados e relatados na pesquisa de (CANDANEDO e FELDHEIM, 2016), conforme pode ser observado na Tabela 11, o C5.0 alcançou uma accuracy equivalente 100% em relação ao conjunto de treinamento da mesma forma que ocorreu com o algoritmo Floresta Aleatória (RF). Em relação ao conjunto de teste 1, o C5.0 superou os algoritmos RF, GBM e CART,

ficando com uma accuracy abaixo apenas do algoritmo LDA, provavelmente devido ao fato de o algoritmo objeto de estudo nessa pesquisa retornar melhores respostas com grandes conjuntos de dados. No panorama do conjunto de dados empregado para testar o classificador, o C5.0 superou todos os outros sistemas relatados no estudo de Candanedo e Feldheim (2016), com uma resposta equivalente a 99,22%. Em palavras simples, ao testar um modelo de aprendizado de máquina gerado em dados previamente calibrados com saída, é medido qual porcentagem da saída do modelo é igual à saída original nos dados de teste. A precisão do modelo de aprendizado de máquina é a medida usada para determinar qual modelo é melhor na identificação de relacionamentos e padrões entre variáveis em um conjunto de dados com base nos dados de entrada ou treinamento. Quanto melhor um modelo pode generalizar para dados 'invisíveis', melhores previsões e insights que produz que proporcionam mais valor comercial.

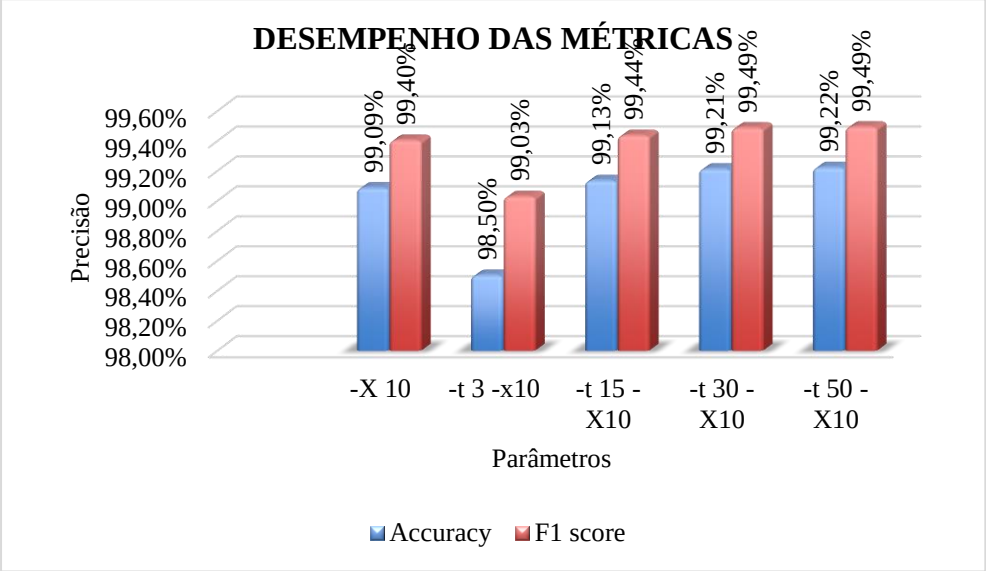
Tabela 11 - Comparativo entre os algoritmos

MÉTRICA: ACCURACY (resultados em %)					
ID	**MODELO**	**PARÂMENTROS**	**TREINAMENTO**	**TESTE 1**	**TESTE 2**
01	RF	umidade, luz, CO_2, temperatura, umidade relativa	100,00	95,05	97,16
02	GBM		99,98	93,06	95,14
03	CART		99,30	95,57	96,47
04	LDA		98,78	97,90	98,76
05	C5.0		100,00	96,32	99,22

Fonte: Desenvolvido pelo autor.

O resultado alcançado com a métrica accuracy pode ser enganoso. Por exemplo, em um problema em que há um grande desequilíbrio de classe, um modelo pode prever o valor da classe majoritária para todas as previsões e alcançar uma alta precisão de classificação. Portanto, este estudo faz uso de indicadores de desempenho mais adequado pare esse tipo de tarefa, uma vez que se torna necessárias medidas de desempenho adicionais, como a F1 Score. A métrica accuracy tem como principal desvantagem o chamado problema de classificação não balanceada. A Figura 14 mostra o desempenho das métricas accuracy e F1 Score. Recapitulando o que foi sobre a tarefa de classificação que envolve a atribuição de quais elementos um conjunto de categorias ou rótulos devem ser atribuídos a alguns dados, de acordo com algumas propriedades dos dados.

Figura 14 - Comparativo entre as métricas



Fonte: Desenvolvido pelo autor.

5 CONCLUSÃO

A detecção de ocupação do ambiente de trabalho é um campo de estudos promissor e definido por condutas que visam a preservação do meio ambiente, redução do desperdício, conforto e bem-estar dos ocupantes. O modelo proposto visa colaborar para o tratamento dos dados gerando informações precisas para determinar se o ambiente está ou não ocupado por pessoas. Esse é um dos requisitos essenciais que pode ser aplicado na construção de edifícios ecologicamente corretos, entre tantas outras utilidades dessa área de estudo. Os resultados experimentais mostram que, a partir da alteração de alguns hiperparâmetros o algoritmo C5.0 (RULEQUEST, 2019) consegue alcançar índices de precisão equivalente, e em alguns casos chegando até superar os resultados anotados no estudo que deu origem às bases de dados empregadas neste estudo (CANDANEDO e FELDHEIM, 2016), aplicando uma metodologia semelhante para determinar a ocupação. Além disso, o modelo proposto superou os modelos existentes na literatura analisada com uma diferença que pode chegar a 0,67% na acurácia. A melhoria da acurácia foi alcançada usando o método boosting com especificação de 50 tentativas para ajustar sequencialmente vários modelos simples, chamados aprendizes fracos, para que cada modelo aprenda com os erros do modelo anterior, possibilitando assim a diminuição da curva dos dados classificados com erro, aliado com técnicas de seleção de recursos dos dados a fim de encontrar os parâmetros com os quais o classificador busca encontrar os melhores resultados.

Este trabalho mostrou que é possível obter alto desempenho na determinação de ocupação com dados de sensores de umidade, CO_2, luz e temperatura e o algoritmo de classificação C5.0. As respostas menos significativas foram obtidas quando foram utilizados apenas dois atributos ou menos, nesse caso a taxa de erro chegou a 14,70 % quando a classificação utilizou apenas o atributo umidade relativa e 4,5% com as propriedades CO_2 e umidade relativa, provavelmente devido à presença de variáveis altamente correlacionadas conforme estudo de (CANDANEDO e FELDHEIM, 2016). No entanto, foram encontrados elevados índices de precisão com a métrica Precision (cerca de 99,79%) com a combinação dos parâmetros -t 50 -c 5 -m 8 -X 10 para o cenário 3, e 99,83% com o hiperparâmetro -t 3 no cenário 2 com o conjunto de testes. Esta pesquisa também mostrou que, em geral, é uma boa prática incluir informações relacionadas à hora do dia ao criar os modelos de classificação, no ambiente de ensaios nomeado cenário 1, todas as métricas utilizadas retornaram resultados igual a 1,00 combinadas com os diferentes parâmetros de entrada para o conjunto de treinamento. Existe a suspeita de que possa ter ocorrido um ajuste excessivo

das árvores do cenário 1. Para garantir a reprodutibilidade dos resultados pela comunidade de pesquisa e uma eventual melhora na detecção de precisão ou na comparação de modelos, a descrição dos conjuntos de dados, juntamente com os scripts de processamento de dados, serão fornecidos neste trabalho. Trabalhos futuros também podem se concentrar na determinação do grau de *overfitting*, que é um grande problema árvores de decisão. Isso é especialmente verdadeiro em redes modernas, que geralmente têm um grande número de pesos e vieses. Para treinar de forma eficaz, precisamos de uma maneira de detectar quando o está acontecendo. E precisamos aplicar técnicas para reduzir os efeitos do *overfiting*. Outra opção para melhorar a precisão da detecção de ocupação poderia ser o uso de modelagem probabilística.

REFERÊNCIAS

AGAOGLU, M. Predicting Instructor Performance Using Data Mining Techniques in Higher Education. IEEE Access, volume 4, páginas 2379 - 2387, maio de 2016.

ALSOL. Dez formas de reduzir o consumo de energia elétrica corporativa. Disponível em: http://blog.alsolenergia.com.br/2017/08/energia-eletrica-corporativa-economia. Acesso em: setembro de 2019.

AMAYRI, M. et al. Estimating Occupancy Using Interactive Learning With a Sensor Environment: Real-Time Experiments. IEEE Access, Volume 7, abril/2019.

ANSANAY, G. Estimating Occupancy Using Indoor Carbon Dioxide Concentrations Only in an Office Building: a Method and Qualitative assessment. in 11th REHVA world congresso. Praque, Czech Republic, 2013.

AHMAD, J. et al. Occupancy detection in non-residential buildings – A survey and novel privacy preserved occupancy monitoring solution.

BREIMAN, L. et al. Classification and Regression Trees. Wadsworth. (1984).

BROOKS, J. et al. Energy efficient control of poorly performing HVAC zones in commercial buildings. Energy building. 93 (2015), pp. 160 – 168.

CANDANEDO L. M.; FELDHEIM V. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models University. Energy and Buildings, volume 112, janeiro 2016, páginas 28-3915.

CANDANEDO L. M.; FELDHEIM V.; DERAMAIX. D. A methodology based on Hidden Markov Models for occupancy detection and a case study in a low energy residential building. Energy and Buildings, Volume 148, agosto de 2017, páginas 327-34.

CAMILO, C. O.; SILVA, J. C. Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas. Technical Report - RT-INF_001-09 - Relatório Técnico. 2009.

CAMACHO, D. et al. Next-generation machine learning for biological networks. Cell, 173 (2018), pp. 1581 - 1592.

CARVALHO, H. M. Aprendizado de Máquina voltado para Mineração de Dados: Árvores de Decisão. Brasília, DF, 2014.

DEVELOPERS.GOOGLE. Classificação: Accuracy. Disponível em: https://developers.google.com/machine-learning/crash-course/classification. Acesso em: setembro de 2019.

D'OCA, S.; HONG T.; LANGEVIN, J. The human dimensions of energy use in buildings: a review. Renovar. Sustentar. Energia Rev., 81 (2018), pp. 731 – 742.

FERREIRA, L. D. Técnicas de aprendizado de máquina aplicadas à identificação de perfis de aprendizado em um ambiente real de ensino. Universidade de São Paulo, São Carlos – SP, 2016.

FONSECA, J. Indução de árvores de decisão. Tese de Mestrado, Lisboa. (1994).

GIL, A. C. Métodos e técnicas de pesquisa social. 4. ed. São Paulo: Atlas, 1994. Como elaborar projetos de pesquisa. 4. ed. São Paulo: Atlas, 2008.

HAN, J; KAMBER, M. Data Mining: Concepts and Techniques. Elsevier, 2006.

IBM, KNOWLEDGECENTER. Nó C.5.0. Disponível em: <https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.modeler.help/c50node_general.htm> Acesso em: setembro de 2019.

LAM, K. et al. Information-theoretic environmental features selection for occupancy detection in open offices. P.A. Strachan, N.J. Kelly, M. Kummert (Eds.), Proceedings of the Eleventh International IBPSA Conference, Citeseer (2009), pp. 1460-1467

LONGO, E.; REDONDI, A.; CESANA, M. Accurate occupancy estimation with WiFi and bluetooth/BLE packet capture. Computer Networks, volume 163, novembro/2019.

MEDIUN.COM. Árvore de Decisão. Disponível em: <https://medium.com/machine-learning-beyond-deep-learning/%C3%A1rvores-de-decis%C3%A3o-3f52f6420b69> Acesso em: setembro de 2019.

MICROSOFT.COM. Tarefas de aprendizado de máquina no ML.NET. Disponível em: https://docs.microsoft.com/pt-br/dotnet/machine-learning/resources/tasks. Acesso em: setembro de 2019.

MITCHELL, T. Machine Learning. New York, NY. McGraw-Hill Science, Engineering, março/1997.

MORA, D. et. al. Occupancy patterns obtained by heuristic approaches: Cluster analysis and logical flowcharts. A case study in a university office. Energy and Buildings, Volume 186, março/2019, Páginas 147-168.

PETERSEN, S. et. al. Establishing an image-based ground truth for validation of sensor data-based room occupancy detection. Energy Build., 130 (2016), pp. 787-793.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL. FACULDADE DE ENGENHARIA. Grupo de Eficiência Energética. USE - Uso Sustentável da Energia [recurso eletrônico]: guia de orientações / PUCRS, FENG, GEE, PU; coord. PROAF. – Dados eletrônicos. - Porto Alegre: PUCRS, 2010.

QUINLAN, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

SHAREENGINEER.COM. Classificação por indução de árvore de decisão. Disponível em: http://shareengineer.blogspot.com/2012/09/classification-by-decision-tree.html. Acesso em: setembro/2019.

SIMON, P. Too Big to Ignore: The Business Case for Big Data. Wiley (2013). 89 páginas.

PRATI, R. C. Novas Abordagens em aprendizado de máquina para a geração de regras, classes desbalanceadas e ordenação de casos. ICMC/USP, São Carlos – SP, julho de 2006.

RULEQUEST. C5.0: um tutorial informal. Disponível em: https://www.rulequest.com/see5-unix.html#DATA. Acesso em: setembro de 2019.

TIMILEHIN, L. et, al. Occupancy Measurement In Commercial Office Buildings For Demand-Driven Control Applications - A Survey And Detection System Evaluation.

WITTEN, I. H.; FRANK, E. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. 2nd. ed. San Francisco, CA: Morgan Kaufmann, 2000.

WITTEN, Ian H. et al. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.

YADAV, S.; SHUKLA, S. Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. In: Advanced Computing (IACC), 2016 IEEE 6th International Conference on. IEEE, 2016. p. 78-83.

YANG, Z. et. al. A systematic approach to occupancy modeling in ambient sensor-rich buildings. Simulation, 90 (8) (2014), pp. 960-977.

YANG, Z. et. al. A multi-sensor based occupancy estimation model for supporting demand driven HVAC operations. Proceedings of the 2012 Symposium on Simulation for Architecture and Urban Design, Society for Computer Simulation International, San Diego, CA, USA (2012), pp. 49-56.