JNT - FACIT BUSINESS AND TECHNOLOGY JOURNAL - ISSN: 2526-4281 - QUALIS B1
ANO 2024 – JULHO - FLUXO CONTÍNUO – Ed. 52. Vol. 1. Págs.185-206
DOI: 10.5281/zenodo.12820085

CAPES

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

ESTRATÉGIA *LOW-CODE* COM APRENDIZADO DE MÁQUINA PARA A ÁREA DE SAÚDE: AVALIANDO A CORRELAÇÃO DA ATIVIDADE OCUPACIONAL COM A INCIDÊNCIA DE CÂNCER NO BRASIL

LOW-CODE STRATEGY WITH MACHINE LEARNING FOR THE HEALTHCARE AREA: ASSESSING THE CORRELATION OF OCCUPATIONAL ACTIVITY WITH THE INCIDENCE OF CANCER IN BRAZIL

Rafael Luz de QUEIROZ[1]
Universidade Salvador (UNIFACS)
E-mail: rafinhalq@gmail.com
ORCID: http://orcid.org/0009-0008-8285-796X

Joberto S. B. MARTINS[2]
Universidade Salvador (UNIFACS)
E-mail: joberto@ieee.org
ORCID: http://orcid.org/0000-0003-1310-9366

RESUMO

A inteligência artificial e o aprendizado de máquina (*machine learning*) têm sido largamente utilizados com grandes benefícios em diversas áreas do conhecimento, inclusive na área de saúde. Entretanto, existe uma barreira importante na disseminação do uso do aprendizado de máquina entre os profissionais da área de saúde que consiste, principalmente, na sua não familiaridade com os conceitos de programação. A estratégia de desenvolvimento denominada 'low-code', quando aplicada ao desenvolvimento de software, inclui arcabouços e ferramentas que, em resumo, tornam o desenvolvimento de software mais acessível para comunidades de profissionais. Essa estratégia é relevante para as cidades inteligentes que visam desenvolver abordagens que possam tornar as cidades mais eficientes, humanas e sustentáveis e contribui para o alcance dos objetivos de desenvolvimento sustentável (ODS) da ONU. Esse artigo posiciona a estratégia low-code através do arcabouço PyCaret como uma inovação e contribuição para o desenvolvimento de sistemas para a saúde em cidades inteligentes. Um estudo de caso que avalia a incidência e a

[1] Acadêmico do curso de Ciência da Computação da Universidade Salvador - UNIFACS.
[2] Professor Titular da Universidade Salvador (UNIFACS) e *International Professor of the Hochschule für Technik und Wirtschaft des Saarlandes* (HTW).

correlação de atividades ocupacionais com a ocorrência de câncer no Brasil usando um algoritmo de detecção de anomalias apresenta a estratégia low-code. As contribuições do artigo incluem o posicionamento da estratégia low-code como inovação em cidades inteligentes e a apresentação de um estudo de caso como base de desenvolvimento de aplicações na área de saúde. O estudo de caso apresentado apresenta uma abordagem diferenciada de detecção de câncer usando um algoritmo de detecção de anomalia e reitera correlações da incidência de determinados tipos de câncer com relação às atividades ocupacionais.

Palavras-chave: Estratégia *Low-Code*. Aprendizado de Máquina. Área de Saúde. Cidade Inteligente. Detecção de Anomalia.

ABSTRACT

Artificial intelligence and machine learning have been widely used and have significant benefits in various areas of knowledge, including healthcare. However, an essential barrier to disseminating machine learning among healthcare professionals is their unfamiliarity with programming concepts. The development strategy called low-code, when applied to software development, includes frameworks and tools that, in short, make software development more accessible to communities of professionals. This strategy is relevant for smart cities that aim to develop approaches to make cities more efficient, humane, and sustainable and contribute to achieving the UN's sustainable development goals (SDGs). This article positions the low-code strategy through the PyCaret framework as an innovation and contribution to developing health systems in smart cities. A case study that evaluates the incidence and correlation of occupational activities with the occurrence of cancer in Brazil using an anomaly detection algorithm presents the low-code strategy. The article's contributions include positioning the low-code strategy as an innovation in smart cities and presenting a case study as a basis for developing applications in the healthcare area. The case study presented presents a differentiated approach to cancer detection using an anomaly detection algorithm and reiterates correlations between the incidence of certain types of cancer and occupational activities.

Keywords: Low-Code Strategy. Machine Learning. Health. Smart City. Anomaly Detection.

INTRODUÇÃO

Existem no Brasil e no mundo um número significativo de bases de dados para a área da saúde com levantamentos de informações sendo criadas e mantidas por diferentes órgãos municipais, estaduais e federais da área (Figueiredo et al. 2023, Gallo et al. 2022). Como consequência dessa disponibilidade, os profissionais e pesquisadores da área de saúde e de outras áreas que tratam de diferentes aspectos da sociedade tem a possibilidade de utilizar essas bases de dados como fonte para extração de diversos tipos de dados e de conhecimento de maneira geral (Morine et al. 2022).

É também fato que a inteligência artificial (IA) tem sido utilizada por diversas áreas do conhecimento (exatas, saúde e humanas) objetivando otimizar processos, detectar anomalias, extrair conhecimento e organizar ou agrupar informações relacionadas, dentre muitas outras possibilidades (Dametto et al. 2022). Em relação à área de saúde e conforme indicado em (Siwicki 2017), 86% das empresas da área num escopo global fazem uso de algum tipo de inteligência artificial.

Assim sendo, a existência de bases de dados da área de saúde constitui um potencial interessante e relevante a ser explorado pelos profissionais da área. Nesse contexto específico, os algoritmos de aprendizado de máquina (*Machine Learning* - ML) são opções versáteis da inteligência artificial que podem ser aplicados na resolução de uma ampla variedade de problemas (Zhang et al. 2023, Mduma et al. 2019).

Entretanto, existe uma barreira importante na disseminação do uso do aprendizado de máquina entre os profissionais da área de saúde que, em resumo, consiste na dificuldade de abordar a escolha e a aplicação e dos métodos e algoritmos de ML baseados em procedimentos que demandam um conhecimento mais aprofundado de matemática ou de computação.

A estratégia denominada *'low-code'*, quando aplicado ao desenvolvimento de software, inclui plataformas de desenvolvimento que tornam o desenvolvimento de aplicativos mais acessível a uma comunidade mais ampla de desenvolvedores, simplificando o processo de desenvolvimento de aplicativos (Juhás et al. 2022). As

plataformas low-code simplificam o desenvolvimento de software utilizando recursos como programação visual, desenvolvimento orientado a modelos, modelagem e gerenciamento de processos e fluxos de desenvolvimento.

Um exemplo de arcabouço no estilo *low-code* é o PyCaret (Ali 2023). Com a utilização do PyCaret, profissionais e pesquisadores da área médica podem aproveitar os benefícios dos algoritmos e métodos de aprendizado de máquina sem a necessidade de conhecimentos avançados em programação ou de aprendizado de máquina (ML). O PyCaret tem como proposta uma interface amigável e intuitiva que permite a construção e avaliação de modelos de aprendizado de máquina com algumas linhas de código, acelerando significativamente o processo de análise e interpretação de dados.

Além disso, o PyCaret oferece uma ampla gama de algoritmos de aprendizado de máquina pré-implementados, bem como ferramentas de pré-processamento de dados e avaliação de modelos, simplificando ainda mais o processo de desenvolvimento e refinamento de modelos de ML.

A motivação desse artigo consiste em cobrir uma necessidade (*gap*) existente no sentido de prover uma solução mais simples e amigável de desenvolvimento para que profissionais da área de saúde e, e eventualmente de outras áreas, possam desenvolver de maneira mais fácil e intuitiva soluções para os seus problemas com o uso do aprendizado de máquina.

As contribuições desse artigo são como segue:

- Posicionar e propor uma estratégia de desenvolvimento no estilo low-code como um elemento de contribuição para os objetivos de desenvolvimento sustentável (ODS) no contexto das cidades inteligentes;
- Desenvolver um estudo de caso no estilo *low-code* que possa servir de referência para outros desenvolvimentos; e
- Avaliar a correlação de atividades ocupacionais com a incidência de câncer no Brasil como um estudo de caso que ilustra a utilização da estratégia *low-code* para a área de saúde

Este artigo está organizado com a Seção 1 que apresenta a estratégia *low-code*. A Seção 2 introduz o arcabouço PyCaret. A Seção 3 introduz o caso de uso sobre a

incidência de câncer e apresenta um *background* sobre a detecção de anomalias. A Seção 4 apresenta o estudo de caso de detecção de câncer usando um algoritmo de detecção de anomalia no estilo *low-code*. Por último, a Seção 5 apresenta as considerações finais do artigo.

ESTRATÉGIA *LOW-CODE* PARA A ÁREA DE SAÚDE NAS CIDADE INTELIGENTES ALINHADA COM OS OBJETIVOS DE DESENVOLVIMENTO SUSTENTÁVEL (ODS)

A estratégia *low-code* com aprendizado de máquina para a área de saúde leva, de maneira geral, a uma melhoria na qualidade de vida dos cidadãos. Isso se deve ao fato que a estratégia tem o potencial de gerar novas soluções inteligentes e otimizadas para a mitigação de diversos problemas relativos a doenças que beneficiam os cidadãos como um todo.

Numa perspectiva e visão contemporâneas, a estratégia *low-code*, quando aplicada na área da saúde, contribui de forma significativa para a gestão das cidades inteligentes e se alinha com os objetivos de desenvolvimento sustentável (ODS) da ONU.

Conforme discutido em (Albino et al. 2015), o conceito de cidade inteligente (*smart city*) surgiu inicialmente como referência a uma abordagem que utiliza as Tecnologias da Informação e Comunicação (TICs) (Escorcia Guzman et al. 2022) ou, de maneira geral, utiliza tecnologia em larga escala nas cidades. Nessa perspectiva, a mera instalação ou utilização de recursos tecnológicos nas cidades tornaria a cidade uma 'cidade inteligente'.

Essa visão evoluiu. Numa visão contemporânea e holística mais atual, uma cidade inteligente é aquela na qual a inovação, a tecnologia e a governança são utilizadas de forma articulada e integrada, e estão redefinindo a governança urbana tradicional visando desenvolver abordagens que possam tornar as cidades mais eficientes, humanas e sustentáveis (Farid et al. 2021a, Martins 2018, Farid et al. 2021b).

A estratégia low-code numa cidade inteligente usa a tecnologia de aprendizado de máquina visando, a título de exemplo, facilitar a tomada de decisão em relação aos problemas de saúde de uma cidade. Isso permite um planejamento e uma gestão mais eficiente dos problemas de saúde de uma cidade.

Os objetivos de desenvolvimento sustentável (ODS) (*Sustainable Development Goals* - SDG), em resumo, são um conjunto de dezessete (17) metas globais propostas pela Organização das Nações Unidas (ONU) em 2015 que visam acabar com a pobreza, proteger o planeta e garantir prosperidade para as pessoas até 2030 (Sorooshian 2024).

Em relação aos objetivos de desenvolvimento sustentável (ODS), a estratégia low-code para a área da saúde se enquadra no ODS 3 (Boa Saúde e Bem-Estar) pois a pesquisa visa contribuir para a solução de um problema importante das cidades inteligentes que é a tomada de decisão na área de saúde suportando o planejamento e a governança de cidades.

ESTRATÉGIA *LOW-CODE* COM O ARCABOUÇO PYCARET

Neste artigo, a estratégia *low-code* faz uso do arcabouço PyCaret (Ali 2023). O arcabouço PyCaret é, de fato, uma biblioteca *low-code* e *open-source* de aprendizado de máquina em Python. Com essa biblioteca, é possível automatizar os fluxos de desenvolvimento de software utilizando algoritmos de aprendizado de máquina em todo o processo, desde a construção até a utilização do modelo. Isso resulta em um aumento considerável na velocidade do desenvolvimento dos experimentos com aprendizado de máquina, tornando-os mais produtivos.

Uma das principais características do PyCaret é a capacidade de substituir dezenas de linhas de código por algumas poucas, tornando o processo de implementação de modelos de aprendizado de máquina mais simples e acessível. Um dos objetivos do PyCaret é mitigar a lacuna entre cientistas de dados e as áreas de negócio, tornando o processo de análise de dados mais acessível e compreensível para todos os envolvidos.

Em termos práticos, o PyCaret organiza seus blocos de construção de experimentos em módulos, cada um encapsulando os algoritmos e as funções de aprendizado de máquina usados consistentemente em diferentes contextos. Isso gera uma padronização do fluxo de desenvolvimento, independentemente do modelo utilizado. Atualmente, são disponibilizados cinco desses módulos, divididos em três categorias: i) Aprendizado supervisionado (com os módulos de classificação e regressão); ii) Aprendizado não supervisionado (com os módulos de clusterização e

detecção de anomalia); e iii) Séries temporais (com um módulo específico para essa categoria).

Além disso, a biblioteca PyCaret oferece capacidades de *deploy*[3] , permitindo que os experimentos de aprendizado de máquina sejam executados em *pipelines*[4]. O PyCaret também oferece compatibilidade com diversos ambientes que suportam a linguagem Python, como *Power* BI, e interfaces gráficas que permitem uma melhor visualização e interação com o usuário. Em resumo, o PyCaret é um arcabouço que simplifica e acelera o processo de desenvolvimento de modelos de aprendizado de máquina, tornando-os acessíveis a uma gama mais ampla de usuários, desde iniciantes até profissionais experientes.

CASO DE USO - INCIDÊNCIA DE CÂNCER NO BRASIL E DETECÇÃO DE ANOMALIAS

O câncer é um desafio significativo em termos de saúde pública tanto em países desenvolvidos quanto em desenvolvimento, resultando em mais de nove milhões de mortes a cada ano, o que equivale a cerca de 12% de todas as causas de mortalidade global (*Organization* 2022). O número de casos de câncer tem experimentado um aumento considerável em todo o mundo, particularmente desde o século passado, tornando-se um dos principais problemas de saúde pública no mundo contemporâneo. A distribuição dos diferentes tipos de câncer varia de acordo com as características de cada região, destacando a importância do estudo dos padrões dessa doença para seu monitoramento e controle eficazes (M. R. Guerra 2004).

Diante desse contexto, é crucial para a área de saúde a identificação de novas formas de detecção do câncer, por exemplo, fazendo uso dos algoritmos de aprendizado de máquina. Nessa perspectiva, conduzimos um experimento no estilo low-code que faz uso de um algoritmo de detecção de anomalia das condições de saúde do paciente utilizando o arcabouço PyCaret.

[3] *Deploy* é o processo de colocar um software ou aplicação em funcionamento em um ambiente de produção ou de testes, tornando-o acessível para os usuários finais ou para outros sistemas.
[4] *Pipeline* e uma série de etapas automáticas que processam dados ou software, movendo-o através de diferentes fases, como construção, testes e implantação, para garantir qualidade e eficiência.

Estratégia *low-code* e definição do algoritmo de aprendizado de máquina

Na utilização do aprendizado de máquina na área de saúde e, também, numa estratégia *low code*, um dos problemas enfrentados pelo profissional da área de saúde é a escolha do algoritmo que deve ser utilizado para a análise de dados desejada.

Desde que os profissionais fora da área de computação não possuem necessariamente uma formação em matemática e em algoritmos que lhes permite fazer escolhas considerando as características fundamentais dos algoritmos, a solução normalmente utilizada é basear a escolha na experimentação bem-sucedida realizada por outros grupos de pesquisa da comunidade. Na estratégia proposta com low-code, essa lógica também se aplica pois, no caso, a estratégia *low-code* foca na questão da facilitação da execução dos algoritmos visando a obtenção de resultados com a experimentação. Assim sendo, a escolha do algoritmo deve ser feita de forma independente e não é coberta pela estratégia *low-code*.

Detecção se anomalia – *Background*

Com o crescente volume de dados sendo adquiridos atualmente, abordagens de detecção de anomalias se tornaram importantes para aprimorar os sistemas de tomada de decisão. Anomalias representam irregularidades nos padrões dos dados, que podem resultar de desvios, adulterações ou inconsistências. Seu estudo abrange qualquer comportamento anormal que possa sinalizar potenciais riscos (Pinto 2023).

As anomalias podem ser definidas como um conjunto de dados que se desvia dos padrões identificados como normais ou típicos, sendo classificadas como exceções ou peculiaridades. Esses pontos fora da curva nos dados merecem atenção especial, uma vez que indicam uma alteração significativa no comportamento do conjunto de dados.

Trabalhos recentes têm aplicado os algoritmos de aprendizado de máquina para a detecção de anomalias, principalmente em imagens, para a identificação de padrões nos dados que não seguem um comportamento esperado previamente definido (Tschuchnig and Gadermayr 2022). Esse processo visa encontrar amostras que se distanciam consideravelmente, com base em uma determinada métrica, do restante do conjunto de dados. Essas amostras anômalas são comumente referidas como outliers.

Essa abordagem é crucial em uma variedade de aplicações, incluindo diversos tipos de diagnósticos médicos (Bao et al. 2023).

O uso do PyCaret para a detecção de anomalias em diagnósticos médicos representa uma abordagem eficaz e acessível, que pode levar a uma identificação mais precisa de padrões anômalos nos dados, contribuindo assim para uma prática médica mais assertiva.

ATIVIDADE OCUPACIONAL E CÂNCER – EXPERIMENTAÇÃO

A utilização do arcabouço PyCaret numa estratégia *low-code* pressupõe a utilização de uma base de dados e segue a sequência de etapas conforme ilustrado na Figura 1.

O experimento de utilização de algoritmo de detecção de anomalia para identificação de casos de câncer foi, em resumo, em dividido em três etapas:

1) Na etapa 1 do experimento, realiza-se a preparação do modelo, onde é feita a tratativa inicial da base de dados para o treinamento e a inicialização (*setup*) da *pipeline* de transformação do arcabouço PyCaret;
2) Na etapa 2 do experimento, realiza-se a criação do modelo de aprendizado de máquina. Nessa etapa, é definido o algoritmo utilizado que é inicialmente treinamento, resultando em um algoritmo treinado e utilizável para o experimento; e
3) Na etapa 3 do experimento, realiza-se a execução do modelo, onde o algoritmo escolhido produz seus dados, com tabelas e gráficos sendo criados para a análise e visualização dos resultados obtidos.

Essas etapas são detalhadas nas seções subsequentes.

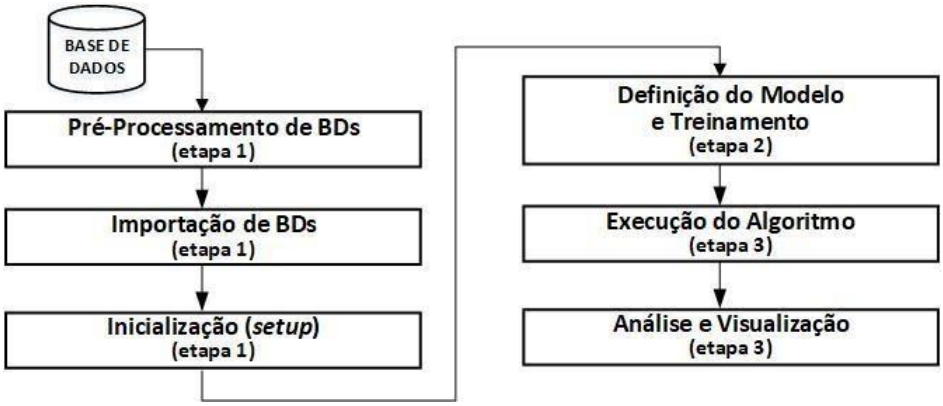

Figura 1. Sequência de Etapas – Estratégia *Low-Code.*

Bases de Dados - Correlação de Atividade Ocupacional com Ocorrência de Câncer

Para conduzir nosso experimento, utilizamos as bases de dados com informações pertinentes para o problema tratado como segue:

- Fonte: DataSUS (Dat 2024) - Sistema de Informação de Agravos de Notificação (SINAM);
- Seção: Epidemiologia e Morbidade;
- Cobertura: Abrange informações sobre doenças e agravos de notificação a partir de 2007.

O DataSUS, operado pelo Departamento de Informação e Informática do Sistema Único de Saúde (SUS), desempenha um papel crucial na coleta, processamento e custódia de dados de saúde pública no Brasil, fornecendo uma ampla gama de informações para subsidiar análises objetivas da situação sanitária, embasar tomadas de decisão e elaborar programas de ações.

Nosso foco foi analisar dados relacionados ao câncer ocupacional. Para o nosso experimento, extraímos duas tabelas de dados do DataSUS para o período de 2019 a 2023:

- Tabela com o número de notificações por CID (Classificação Internacional de Doenças) relacionadas ao câncer; e
- Tabela com o registro do número de notificações por ocupação.

O processamento das bases de dados pelos algoritmos de aprendizado de máquina, na grande maioria dos casos, requer um tratamento dos dados que será detalhado na seção detalhando o experimento (Seção 5). Esse pré-processamento é essencial para assegurar a precisão e a consistência dos dados antes de prosseguirmos com a análise.

Preparação (*Setup*) do Experimento

Na fase de preparação (*setup*) inicial, utilizamos o *dataset* de notificações por tipo de câncer. Após obter as tabelas da base de dados do DataSUS para o experimento, é necessário realizar um pré-processamento dos dados. Embora o PyCaret ofereça um

conjunto de tratamentos de forma automática como, tentar inferir se o dado é categórico, ou ordinal, numérico ou em texto, além de realizar durante essa fase, a conversão de dados categóricos para numéricos o *encoding*, para que os dados possam ser interpretados pelo modelo. Fazendo esse uma tratativa inicial dos dados é normalmente necessária antes de executar o algoritmo para garantir que os dados estejam adequados e possamos obter os resultados desejados.

A base de dados importada do DataSUS tem um conjunto de campos e formatação com a informação do CID câncer e a quantidade de notificações. As etapas de pré-processamento normalmente executadas são como segue:

- Retirada de informações não relevantes em relação aos objetivos do experimento;
- Conformação da base de dados para que contenha dados explícitos importantes para o experimento por colunas.

Assim sendo, primeiramente, removeu-se a linha contendo o número total de casos, para que não influencie no cálculo da detecção de anomalias e evite uma distorção nos resultados. Outro tratamento importante para garantir que os resultados sejam satisfatórios é a separação da coluna contendo os CID relacionados ao câncer. Para detectar anomalias considerando o tipo de notificação, é fundamental explicitar essa informação em uma coluna. Portanto, além de remover a linha contendo o número total de casos, também separamos a coluna de CID câncer das demais colunas de notificações. Esse passo assegura que a análise de anomalias se concentre exclusivamente nos dados de notificação, sem a influência dos demais códigos CID. Após essa separação, os dados estão prontos para serem analisados de forma precisa e eficiente utilizando o arcabouço PyCaret.

Com o pré-processamento da base de dados concluído fora do ambiente PyCaret, prosseguiu-se com a importação e configuração dos dados no arcabouço, preparando o ambiente para a detecção de anomalias e a análise subsequente.

A ’Amostra de Código 1’ ilustra o código (script) utilizado para importar os dados para o ambiente PyCaret. Na estratégia low-code, esse código pode ser generalizado para outros dados utilizados com outros objetivos de análise utilizando outros modelos e algoritmos de diferentes tipos.

```
import pandas as pd

data = pd.read_csv('data/tratado/notificacao_cid_cancer.csv', sep=';')

data = data[data['CID Cancer'] != "Total"]

dataForAnomalyDetection = data.drop(['CID Cancer'], axis=1)

%%%%% Amostra de Codigo 1 %%%%%
```

Após a importação dos dados para o PyCaret, a primeira função que deve ser invocada é a função *setup* ('Amostra de Código 2' - linha 3). Esta função é responsável por inicializar o ambiente PyCaret para o treinamento, além de criar a *pipeline* de transformação. O *setup* retorna variáveis globais que serão utilizadas pelo PyCaret ao longo de todo o ciclo de vida do experimento, incluindo algumas configurações e constantes, tais como o tamanho dos dados.

Definição e Execução do Algoritmo de Detecção de Anomalia

Com a base de dados devidamente processada, importada e o *setup* realizado no arcabouço PyCaret, passamos à fase de definição do algoritmo ou, conforme convenção do arcabouço PyCaret, definição do 'modelo' (*model*) utilizado.

Ainda considerando a estratégia *low-code*, o PyCaret disponibiliza uma variedade de algoritmos em sua biblioteca. Podemos analisar e optar entre esses algoritmos utilizando a função *models* ('Amostra de Código 2' - linha 5), que retorna uma lista de algoritmos disponíveis para o tipo de aprendizado de máquina específico em uso.

```
from pycaret.anomaly import *

s = setup(dataForAnomalyDetection, session_id=123, system_log=False)

models()

iforest = create_model('iforest')

%%%%% Amotra de Codigo 2 %%%%%
```

Existem diversos algoritmos de detecção de na literatura e, dentre esses, destacamos alguns algoritmos mais utilizados como o *Isolation Forest* (Liu et al. 2008), *Support Vector Machine* (SVM) (Hearst et al. 1998) e *Local Outlier Factor* (Breunig et al.

2000), cada um com suas próprias vantagens e áreas de aplicação. Ao empregar essas técnicas, os profissionais da área de saúde podem explorar e compreender melhor os dados, identificando informações valiosas que podem não ser óbvias em uma análise convencional (Carvalho et al. 2018).

Conforme mencionado na Seção 3, o PyCaret é estruturado em módulos distintos agrupando os algoritmos por tipo de aprendizado de máquina: i) Módulo de aprendizado supervisionado; ii) Módulo de aprendizado não supervisionado, incluindo a clusterização e a detecção de anomalias; e iii) Módulo de séries temporais.

Para a realização do experimento de detecção de anomalias, utilizaremos especificamente o modelo *anomaly* (Linha 1 - Amostra de Código 2). Esse modelo inclui um conjunto de algoritmos conforme ilustrado na Tabela 1.

O estudo de caso visando a detecção da correlação entre atividades ocupacionais e a ocorrência de câncer nesse experimento utilizou especificamente o algoritmo de detecção de anomalia *Isolation Forest* (Liu et al. 2008). O algoritmo, no caso, busca as maiores incidências de câncer como uma anomalia no conjunto das incidências existentes.

O algoritmo *Isolation Forest* foi escolhido devido à sua característica de detecção de anomalias baseada na premissa de que as anomalias são raras em comparação com os dados normais. Este modelo não requer dados rotulados e possui menos restrições para distinguir anomalias das observações normais, facilitando a implementação de aprendizado não supervisionado. Os autores em (Samariya et al. 2023) fazem uma apresentação e avaliação de desempenho do algoritmo *Isolation Forest* em relação a outros algoritmos de detecção de anomalias.

Tabela 1. Algoritmos de Detecção de Anomalia - Modelo *Anomaly*

Identificador	Algoritmo
abod	*Angle-base Outlier Detection*
cluster	*Clustering-Based Local Outlier*
iforest	*Isolation Forest*
knn	*K-Nearest Neighbors Detector*
histogram	*Histogram-based Outlier Detection*
lof	*Local Outlier Factor*
svm	*Support Vector Machine*
pca	*Principal Component Analysis*
mcd	*Minimum Covariance Determinant*

sod	*Subspace Outlier Detection*
sos	*Stochastic Outlier Selection*

Numa etapa prévia à execução do experimento de verificação da correlação, o algoritmo de aprendizado de máquina *Isolation Forest* deve ser treinado. Para isso, deve ser utilizada a função *create_model* disponibilizada pelo PyCaret ('Amostra de Código 2' - linha 7). Esta função aceita o modelo selecionado (*iforest*) como parâmetro e retorna o modelo já treinado, pronto para ser utilizado para a detecção de anomalias.

O método *create_model* simplifica o processo de treinamento, automatizando diversas etapas e garantindo que o modelo esteja otimizado para identificar anomalias de forma eficiente nos dados disponíveis.

Execução do Algoritmo e Visualização dos Resultados

O algoritmo *Isolation Forest* opera de maneira que os dados são atribuídos a uma árvore binária[5]. Em cada divisão, uma característica (eixo X ou Y) é selecionada aleatoriamente, e um valor de divisão é escolhido aleatoriamente dentro do intervalo dessa característica, fazendo que o valor dos eixos não tenha uma correlação fixa. Se um valor atribuído à árvore for menor que o valor de divisão, ele é posicionado à esquerda; caso contrário, à direita, criando uma ramificação. Esse processo é repetido até que todos os valores alcancem sua profundidade máxima na árvore, isolando os dados em ramificações específicas. Observa-se que outliers requerem menos divisões para se isolarem, em comparação a dados normais. O *anomaly_score* é calculado com base na facilidade de isolamento dos dados após várias iterações do treinamento. Simplificadamente, coleta-se o número de passos que cada valor leva para se isolar na árvore, calcula-se a média desses passos ao longo das repetições da árvore de treinamento e aplica-se esse valor em uma equação constante para obter o score. Os casos com scores significativamente diferentes dos demais são considerados anomalias.

A execução do algoritmo de detecção de anomalias é efetivamente realizada com a função *assign_model* ('Amostra de Código 3' - linha 3). Essa função atribui uma categoria a cada ponto de dado, classificando-os como anomalias (1) ou não anomalias

[5] Uma árvore binária é uma estrutura de dados hierárquica na qual cada nó pode ter até dois filhos. Ela é usada em várias aplicações computacionais para organizar e acessar dados de forma eficiente.

(0). Além disso, a função adiciona uma coluna denominada *Anomaly Score*, que indica o grau de anomalia de cada ponto de dado, permitindo uma comparação entre as anomalias encontradas.

A visualização dos dados processados pelo algoritmo *Isolation Forest* ajuda na interpretação dos resultados obtidos e o arcabouço PyCaret oferece algumas opções para esse fim.

```
1  plot_model(iforest, plot='umap', label=True)
2
3  result = assign_model(iforest)
4
5  cidCancerColumn = data['CID Cancer']
6
7  result.insert(0, 'CID Cancer', cidCancerColumn)
8
9  result[result['Anomaly']==1]
10
11 %%%%% Amostra de Codigo 3 %%%%%
```


Neste estudo de caso, utilizamos a função *plot model* (‘Amostra de Código 3’, linha 1), que gera um gráfico de dispersão mostrando a performance do modelo. A técnica de plotagem UMAP (*Uniform Manifold Approximation and Projection*) foi escolhida dado que estamos trabalhando com dados bidimensionais (Figura 2). Cada ponto no gráfico representa uma incidência de CID, sendo os dados normais representados em azul e os dados anômalos com incidência de câncer em amarelo. Em suma, a plotagem destaca visualmente as anomalias, tornando a avaliação dos resultados do algoritmo mais acessível e compreensível.

Incidência de Câncer por Detecção de Anomalia e Correlação da Incidência de Câncer com Atividade Ocupacional - Resultados e Discussão

Em relação ao estudo de caso, o método *assign_model* retorna um *dataset* com os dados categorizados na coluna ’Anomalia’ como anômalos (representados por 1) e não anômalos (representados por 0). Esse método também adiciona a coluna *Anomaly Score*, um valor numérico gerado pelo modelo de detecção de anomalias que indica o grau de anomalia de cada observação.

Em suma, o resultado do método *assign_model* é um *dataset* contendo o número de notificações, a classificação de anomalia e o *Anomaly_Score*. Em seguida,

reintegramos a coluna contendo o CID Câncer, que havia sido removida anteriormente. Isso nos permite exibir os CIDs correspondentes ao número de notificações e se estas são anomalias, conforme exibido na Tabela 2. Através da aplicação de técnicas de filtragem, foi possível isolar e identificar os dados anômalos no conjunto analisado. Esse procedimento expõem os códigos CID que apresentaram um número de notificações considerado anômalo no período de 2019 a 2023, conforme mostrado na Tabela 3.

Em relação às notificações de tipos de incidência de câncer no Brasil, observa-se que o câncer de pele, identificado pela CID C44, emerge como a neoplasia maligna mais comum, superando outras categorias. Os principais fatores de risco para o desenvolvimento deste tipo de câncer incluem exposição à radiação ultravioleta (UV), contato com herbicidas, formaldeído, clorofluorcarbono, uso de imunossupressores e histórico pessoal ou familiar de melanoma (Farias et al. 2021). O elevado número de casos levanta questionamentos pertinentes sobre suas causas, sendo a exposição ocupacional excessiva ao sol identificada como um fator de risco preponderante.

A eficácia da utilização de um algoritmo de detecção de anomalias na evidenciação de casos anômalos nos registros de notificação de câncer do DataSUS é corroborada por diversos fatores. Primeiramente, a capacidade desse algoritmo de identificar prontamente casos incomuns, destacando registros que se desviam do padrão esperado, contribui para uma análise mais precisa e eficiente. Além disso, a detecção de inconsistências nos dados permite melhorias na qualidade das informações coletadas.

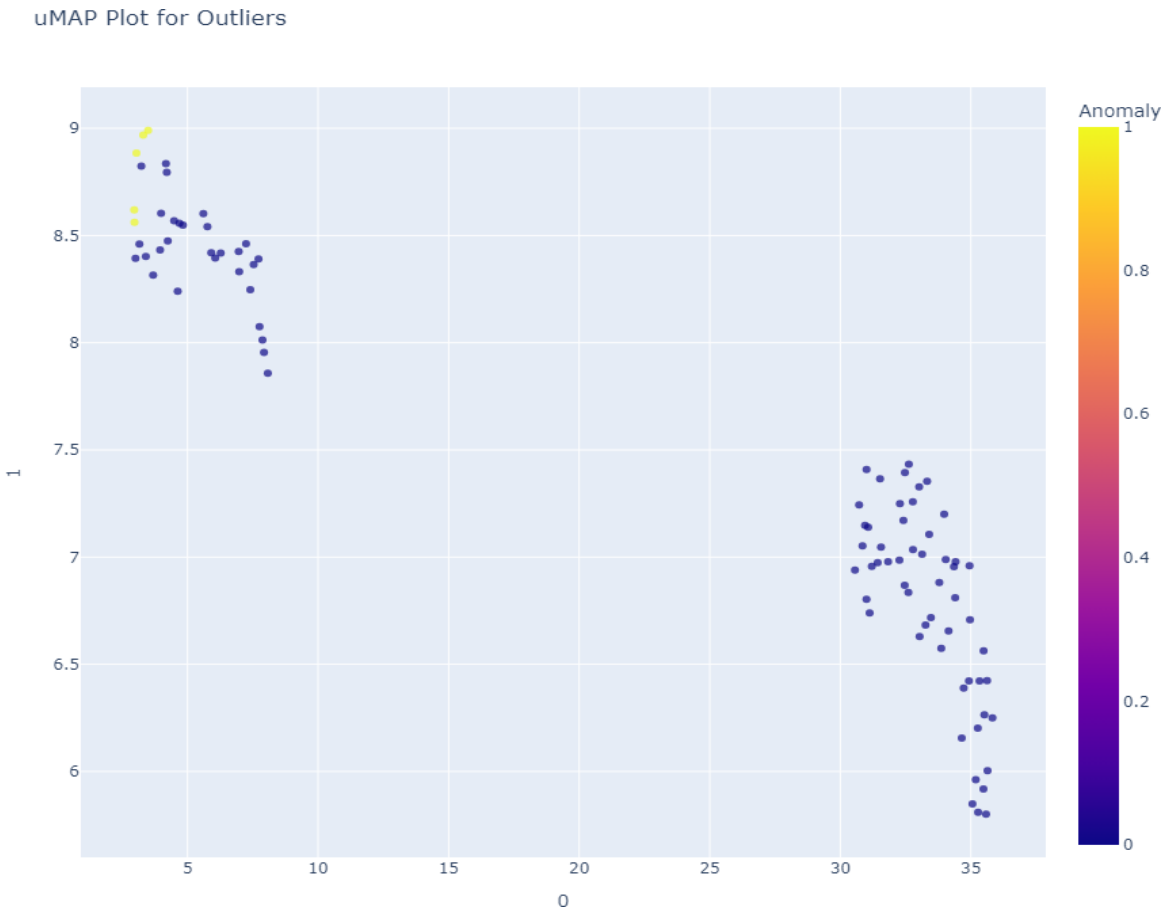

Figura 2. Detecção de Anomalias - *Outliers* - Incidência de Câncer.

Outro aspecto relevante é a agilidade proporcionada pela detecção de anomalias. Relatórios e análises podem ser gerados de forma mais rápida, possibilitando reações ágeis às situações anômalas identificadas. Essa prontidão é crucial para a saúde pública, permitindo intervenções oportunas e estratégias de prevenção mais eficazes.

Tabela 2. Notificações de Tipos de Incidência de Câncer (CID)

Tipo de Câncer (CID)	Notificação	Anomalia	*Anomaly Score*
C00 - Neopl malig do lábio	3	0	-0.226894
C01 - Neopl malig da base da língua	8	0	-0.186863
C02 - Neopl malig e NE da língua	7	0	-0.211619
C04 - Neopl malig do assoalho da boca	4	0	-0.234642
D43 - Neopl comp inc/desc encéfalo	4	0	-0.234642

Tabela 3. Notificações de Tipos de Incidência de Câncer (CID) Filtrando Apenas Anomalias

Tipo de Câncer (CID)	Notificações	Anomalia	Anomaly Score
C34 - Neoplastia maligna dos brônquios/pulmões	248	1	0.149788
C44 - Outra neoplastia maligna da pele	1261	1	0.287152
C50 - Neoplastia maligna da mama	144	1	0.033801
C61 - Neoplastia maligna da próstata	134	1	0.007476
Não preenchidos	603	1	0.215896

Adicionalmente, a aplicação contínua desse algoritmo, monitorando os registros ao longo do tempo, oferece a vantagem de identificar mudanças significativas nos padrões de notificação. Essa vigilância constante possibilita a detecção precoce de tendências preocupantes ou novos cenários, contribuindo para aprimorar as políticas de saúde.

Em suma, a utilização do algoritmo de detecção de anomalias no contexto dos registros de câncer do DataSUS representa uma ferramenta valiosa para aprimorar a qualidade dos dados, identificar casos anômalos e promover uma saúde pública mais eficiente e responsiva.

Diante dessa eficácia, o presente estudo estende essa metodologia para a análise específica das notificações de câncer em relação à atividade ocupacional (trabalho). Ou seja, investiga-se agora a correlação entre a ocorrência de câncer em relação à atividade ocupacional relatada.

Considerando a semelhança estrutural dos dados entre ambas as categorias, modificações mínimas foram aplicadas para adaptar o procedimento experimental. Os dados foram submetidos a uma etapa de pré-processamento destinada à detecção de anomalias, visando remover possíveis pontos de interferência nos resultados. O treinamento subsequente foi conduzido de forma análoga ao procedimento anteriormente descrito. A 'Amostra de Código 4' ilustra as etapas do processamento para essa nova etapa do experimento.

Os resultados da correlação entre a incidência de câncer e a ocupação profissional no Brasil é ilustrada na Tabela 4. Ao analisar as notificações por ocupação, observa-se que a grande maioria das ocupações associadas ao câncer de pele envolve

atividades ao ar livre com exposição direta ao sol (produtor agropecuário, produtor agrícola, trabalhador agropecuário, caseiro, trabalhador volante). Esta constatação sugere uma possível contribuição significativa para o alto número de casos dessa doença. Assim, ao traçar essa correlação, é plausível presumir que a exposição solar ocupacional pode ser um dos principais impulsionadores do aumento da incidência de câncer de pele.


```
import pandas as pd

data = pd.read_csv('data/tratado/notificacao_por_ocupacao.csv', sep=';')

data = data[data['Ocupacao'] != "Total"]

dataForAnomalyDetection = data.drop(['Ocupacao'], axis=1)

from pycaret.anomaly import *

s = setup(dataForAnomalyDetection, session_id=123, system_log=False)

iforest = create_model('iforest')

plot_model(iforest, plot='umap')

result = assign_model(iforest)

ocupacaoColumn = data['Ocupacao']

result.insert(0, 'Ocupacao', ocupacaoColumn)

result[result['Anomaly']==1]

%%%%% Amostra de Codigo 4 %%%%%
```

Essa dedução é reforçada pelos resultados obtidos por meio de uma análise meticulosa dos dados fornecidos por instituições de saúde pública do páis. A compreensão dessas associações entre ocupação e câncer de pele não apenas amplia nosso conhecimento sobre os fatores de risco envolvidos, mas também destaca a necessidade de medidas preventivas e intervenções direcionadas para proteger os trabalhadores expostos a esses riscos ocupacionais.

Tabela 4. Incidência de Câncer por Ocupação

Ocupação	Notificação	Anomalia	*Anomaly Score*
Faxineiro	28	1	0.015448
Cabeleireiro	32	1	0.037030
Frentista	22	1	0.007358
Produtor Agropecuário (em geral)	85	1	0.123468

Produtor Agrícola Polivalente	462	1	0.266661
Cafeicultor	36	1	0.026656
Trabalhador Agropecuário (geral)	816	1	0.300985
Caseiro (agricultura)	86	1	0.123468
Trabalhador Volante (agricultura)	170	1	0.196578
Trabalhador da Cultura de Café	37	1	0.036893

CONSIDERAÇÕES FINAIS

O estudo de caso apresentado como referência e base de desenvolvimento no estilo *low-code* com o aprendizado de máquina na área de saúde contribui com a simplificação do processo de utilização da IA para os profissionais da área de saúde. A utilização mais intensa e sistemática de algoritmos de aprendizado de máquina na área democratiza o acesso a uma tecnologia inovadora e impulsiona os desenvolvimentos da saúde para as cidades inteligentes, permitindo, concomitantemente, o atingimento dos objetivos de desenvolvimento sustentável (ODS) da ONU.

Com a capacidade de automatizar tarefas complexas e reduzir a necessidade de programação extensiva, os arcabouços *low-code*, como o PyCaret, permitem que profissionais de saúde e analistas de dados concentrem seus esforços na interpretação de resultados e na aplicação prática de suas descobertas.

Em relação à detecção de anomalias, observou-se que o algoritmo *Isolation Forest* se mostrou eficiente na aplicação com dados bidimensionais, proporcionando uma detecção eficaz de anomalias na incidência de câncer.

A avaliação da correlação da atividade ocupacional com a incidência de câncer no Brasil inovou, por um lado, por utilizar um algoritmo de detecção de anomalia fora do escopo tradicional de avaliação de anomalias utilizando os recursos de imagens. Por outro lado, a análise realizada reitera a correlação dos casos de incidência de câncer com atividades ocupacionais do campo com forte exposição à radiação solar.

REFERÊNCIAS

BRASIL. MINISTÉRIO DA SAÚDE. (2024). **DATASUS.** Tabnet. Brasília, DF: Ministério da Saúde, 2024.

ALBINO, V., BERARDI, U., AND DANGELICO, R. M. (2015). Smart Cities: Definitions, Dimensions,Performance, and Initiatives. **Journal of Urban Technology,** 22(1):3–21.

ALI, M. (2023). **Announcing PyCaret 3.0** — An open-source, low-code machine learninglibrary in Python.

BAO, J., SUN, H., DENG, H., HE, Y., ZHANG, Z., AND LI, X. (2023). **BMAD**: Benchmarks for Medical Anomaly Detection.

BREUNIG, M. M., KRIEGEL, H.-P., NG, R. T., AND SANDER, J. (2000). LOF: Identifying Density-Based Local Outliers. **SIGMOD Rec.,** 29(2):93–104.

CARVALHO, H. V. F. D., CARVALHO, E. C., ARRUDA, H., IMPERATRIZ-FONSECA, V., SOUZA, P. D., ANDPESSIN, G. (2018). Detecção de Anomalias em Comportamento de Abelhas Utilizando Redes Neurais Recorrentes. In **Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais.** ISSN: 2595-6124.

DAMETTO, M., VECHI, S. M., AND BONACIN, R. (2022). Predicting Cancer Relapse with Machine Learning from an Open Brazilian Database. In 2022. **E-Health and BioengineeringConference (EHB),** pages 1–4. ISSN: 2575-5145.

ESCORCIA GUZMAN, J. H., ZULUAGA-ORTIZ, R. A., BARRIOS-MIRANDA, D. A., AND DELAHOZ DOMINGUEZ, E. J. (2022). Information and Communication Technologies (ICT) in the Processes of Distribution and Use of Knowledge in Higher Education Institutions (HEIS).**Procedia Computer Science**, 198:644–649.

FARIAS, M. B., TOCANTINS, L. B. C., SANTOS, L. S., COSTA, T. D., GALLES, C. B., AND BRAZ, F. R. (2021). Risco de Câncer de Pele Devido à Exposição Solar Ocupacional: Uma Revisão Sistemática. **Braz. Jour. of Health Review,** 4(6):26365–26376.

FARID, A. M., ALSHAREEF, M., BADHESHA, P. S., BOCCALETTI, C., CACHO, N. A. A., CARLIER, C.-I.,CORRIVEAU, A., KHAYAL, I., LINER, B., MARTINS, J. S. B., RAHIMI, W. C. H., STILLWELL, A., AND WANG, Y. (2021a). Smart City Drivers and Challenges in Urban-Mobility, Health-Care, and Interdependent Infrastructure Systems. **IEEE Potentials**, 40(1):11–16.

FARID, A. M., ALSHAREEF, M., BADHESHA, P. S., BOCCALETTI, C., CACHO, N. A. A., CARLIER, C.-I.,CORRIVEAU, A., KHAYAL, I., LINER, B., MARTINS, J. S. B., RAHIMI, F., W. C. H., STILLWELL, A., AND WANG, Y. (2021b). Smart City Drivers and Challenges in Energy and Water Systems. **IEEE Potentials,** 40(1):6–10.

FIGUEIREDO, F. W. D. S., SILVA, B. K. R., NETO, L. S. S., QUARESMA, F. R. P., AND MACIEL, E. D. S. (2023). **Proposal for a Framework for the Use of Secondary Data in Health Education.**

GALLO, S., RIGERS, B., BOTTARO, M., AND ET. AL. (2022). Sigthings of Large Elasmobranchs fromthe Mediterranean: New Data from Medlem Database in the Last Five Years (2017–2022). **International Workshop on Metrology for the Sea**, pages 293–297.

HEARST, M., DUMAIS, S., OSUNA, E., PLATT, J., AND SCHOLKOPF, B. (1998). Support Vector Machines. **IEEE Intelligent Systems and their Applications**, 13(4):18–28.

JUHAS, G., MOLNAR, L., JUHASOVA, A., ONDRISOVÁM., MLADONICZKY, M., AND KOVACIK, T. (2022). Low-Code Platforms and Languages: The Future of Software Development. **Conf. on Emerging eLearning Technologies and Ap**., pp 286–293.

LIU, F. T., TING, K. M., AND ZHOU, Z.-H. (2008). Isolation Forest. In 2008. **Eighth IEEE International Conference on Data Mining**, pages 413–422. ISSN: 2374-8486.

M. R. GUERRA, C. V. GALLO, G. A. M. (2004**).** Risco de câncer no brasil: tendências e estudos epidemiológicos mais recentes. **Revista Brasileira Cancerologia.**

MARTINS, J. S. B. (2018). Towards Smart City Innovation Under the Perspective of Software Defined Networking, Artificial Intelligence and Big Data. **Revista de Tecnologia da Informação e Comunicação,** 8(2):1–7.

MDUMA, N., KALEGELE, K., AND MACHUVE, D. (2019). A Survey of Machine Learning Approaches and Techniques for Student Dropout Prediction. **Data Science Journal,** 18(1).

MORINE, M. J., PRIAMI, C., CORONADO, E., HABER, J., AND KAPUT, J. (2022). **A** Comprehensive and Holistic Health Database. In 2022. **IEEE International Conference on Digital Health(ICDH),** pages 202–207.

ORGANIZATION, W. H. (2022). **Cancer**.

PINTO, S. O. (2023). **Revisão de literatura**: abordagem de detecção de anomalias em sistemas financeiros. UFRGS - Escola de Engenharia.

Samariya, D., Ma, J., Aryal, S., and Zhao, X. (2023). Detection and Explanation of Anomalies in Healthcare Data. **Health Information Science and Systems,** 11(1):20.

SIWICKI, B. (2017). **86% of Healthcare Companies Use Some Form of AI**.

SOROOSHIAN, S. (2024). The Sustainable Development Goals of the United Nations: A Comparative Midterm Research Review. **Journal of Cleaner Production**, 453:142272.

TSCHUCHNIG, M. E. AND GADERMAYR, M. (2022). Anomaly Detection in Medical Imaging A Mini Review. In Haber, P., Lampoltshammer, T. J., Leopold, H., and Mayr, M., editors, Data Science – **Analytics and Applications,** pages 33–38.

ZHANG, X., GUO, F., CHEN, T., PAN, L., BELIAKOV, G., AND WU, J. (2023). A Brief Survey of Machine Learning and Deep Learning Techniques for E-Commerce Research. **Jour. of Theor. and Applied Electronic Commerce Research,** 18(4):2188–2216.