**UNIVERSIDADE FEDERAL DO PAMPA**

**Fábio Righi da Silva**

# PPGEva: Sistema para Avaliar a Relação entre Indicadores e Conceitos de PPGs

Alegrete
2023

**Fábio Righi da Silva**

# PPGEva: Sistema para Avaliar a Relação entre Indicadores e Conceitos de PPGs

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Rodrigo Brandão Mansilha

Alegrete
2023

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

S586p Silva, Fábio Righi da
PPGEva: Sistema para Avaliar a Relação entre Indicadores e Conceitos de PPGs / Fábio Righi da Silva.
45 p.

Trabalho de Conclusão de Curso(Graduação)-- Universidade Federal do Pampa, CIÊNCIA DA COMPUTAÇÃO, 2023.
"Orientação: Rodrigo Brandão Mansilha".

1. Conceitos. 2. Indicadores Bibliométricos. 3. Pós-Graduação. 4. Inteligência Artificial. I. Título.

**FÁBIO RIGHI DA SILVA**

# PPGEva: Sistema para Avaliar a Relação entre Indicadores e Conceitos de PPGs

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em: 04, 12 e 2023.

Banca examinadora:

Prof. Dr. Rodrigo Brandão Mansilha
Orientador
UNIPAMPA

Prof. Dr. Claudio Schepke
UNIPAMPA

Prof. Dr. Diego Luis Kreutz
UNIPAMPA

Kayuã Oleques Paim

UNIPAMPA

Assinado eletronicamente por **DIEGO LUIS KREUTZ**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 04/12/2023, às 19:32, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **CLAUDIO SCHEPKE**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 04/12/2023, às 19:32, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **RODRIGO BRANDAO MANSILHA**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 04/12/2023, às 19:33, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **KAYUÃ OLEQUES PAIM**, **Usuário Externo**, em 05/12/2023, às 10:12, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1312114** e o código CRC **8A1BB12C**.

## RESUMO

Diversos cursos de pós-graduação são ofertados pelas universidades. No Brasil estes cursos são avaliados quadrienalmente pela CAPES, recebendo conceitos entre três e sete. Com o intuito de alcançar conceitos maiores, e consequentemente obterem maiores investimentos, os programas de Pós-Graduação precisam buscar um constante aprimoramento. A maneira de demonstrar a produtividade à CAPES é melhorando os indicadores bibliométricos. Como não é de conhecimento público quais são os índices que mais impactam nos conceitos, este trabalho busca identificar e classificar sistematicamente os indicadores bibliométricos de acordo com sua importância. Para essa proposta, um conjunto de dados bibliométricos são obtidos através de métodos de automação computacional e, posteriormente, processados através de técnicas de inteligência artificial até gerar uma classificação decrescente de influência das métricas na obtenção de conceitos superiores. Propõe-se um sistema computacional para facilitar a execução do processo e assim, ajudar os coordenadores de programas de pós-graduação a identificarem qual plano de gestão devem utilizar para melhorar o conceito de seus programas.



**Palavras-chave**: Conceitos. Indicadores Bibliométricos. Pós-Graduação. Inteligência Artificial.

## ABSTRACT

Several graduate courses are offered by universities. In Brazil, these courses are evaluated every four years by CAPES, receiving ratings ranging from three to seven. In order to achieve higher ratings and consequently obtain greater investments, Graduate Programs need to constantly seek improvement. Demonstrating productivity to CAPES is done by improving bibliometric indicators. Since it is not publicly known which indices have the most impact on ratings, this work seeks to systematically identify and classify bibliometric indicators according to their importance. For this purpose, a set of bibliometric data is obtained through computational automation methods and subsequently processed using artificial intelligence techniques until generating a descending classification of the influence of metrics on obtaining higher concepts. A computational system is proposed to facilitate the execution of the process and, thus, assist postgraduate program coordinators in identifying which management plan to use to enhance the rating of their programs.



**Key-words**: Ratings. Bibliometric Indicators. Graduate Programs. Artificial Intelligence.

SUMÁRIO

# 1 INTRODUÇÃO

Os programas de pós-graduação precisam manter-se em constante aperfeiçoamento. Eles são avaliados pela CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) a cada quatro anos e resultam em um Conceito (entre 1 e 7). Para se aperfeiçoar é importante gerenciar através de inteligência de negócios (SHARDA; DELEN; TURBAN, 2019). A inteligência de negócios é baseada em algoritmos fundamentados em dados de conhecimento que são oriundos de informações que são obtidas a partir de dados brutos (SHARDA; DELEN; TURBAN, 2019).

Os coordenadores dos programas podem obter as informações referentes às produções científicas de diversas formas, manualmente (coleta de dados brutos e posterior processamento) ou automatizada por meio de alguma plataforma. Algumas plataformas, como aquela utilizada pela Unipampa, apenas apresentam valores de índices (informação), não realizando análises sobre eles (inteligência). Desse modo, é desejável o desenvolvimento de alguma técnica de inteligência artificial que auxilie na tomada de decisões estratégicas com base em informações (*i.e.*, dados processados).

Alguns trabalhos já abordaram o tema, como (CAPARELLI; DIGIAMPIETRI, 2018), que obtiveram os indicadores através da plataforma Sucupira[1], da análise das informações dos currículos Lattes[2] de cada docente do programa e pela pesquisa do perfil de cada membro no Google Acadêmico[3]. Após a obtenção das informações, os indicadores foram ordenados conforme o grau de relação entre o índice e o conceito CAPES, com método baseado no valor qui-quadrado. Outro trabalho semelhante foi produzido por (DIGIAMPIETRI et al., 2014), que utilizaram as mesmas técnicas de obtenção dos dados e compararam as métricas dos programas de pós-graduação em Ciência da Computação brasileiros. Contudo, esses trabalhos realizaram os processos de maneira manual, o que dificulta a atualização dos resultados e aplicação em outras áreas do conhecimento.

Nesse contexto, apresentamos a seguinte definição de problema: podemos analisar sistematicamente a relação entre métricas de desempenho e o conceito do programa de acordo com a CAPES? Para responder a esta pergunta, é necessário obter os índices dos programas de pós-graduação. Estes dados devem ser organizados e posteriormente processados, para ser possível apresentar o resultado. Para tal foi desenvolvido um software denominado PPGEva, que é um sistema automatizado para identificar as métricas de desempenho com maior correlação com Conceito alto da CAPES. Este trabalho se diferencia dos trabalhos anteriores por obter os indicadores bibliométricos de maneira sistematizada e automatizada. Além disso, apresentamos um estudo de caso que utiliza informações referentes ao quadriênio 2017 à 2020.

O restante deste trabalho está organizado como segue. O Capítulo 2 apresenta a metodologia que será utilizada neste trabalho. O Capítulo 3 discute os trabalhos relacio-

[1] <sucupira.capes.gov.br>
[2] <lattes.cnpq.br>
[3] <scholar.google.com.br>

nados. O Capítulo 4 detalha a solução proposta. O Capítulo 5 relata um estudo de caso. E o Capítulo 6 apresenta as considerações finais.

# 2 METODOLOGIA

O desenvolvimento deste trabalho emprega a metodologia de pesquisa aplicada, pois visa gerar uma solução para um problema específico (GIL, 2022). A Figura 1 apresenta os passos identificados para resolução do problema. Em seguida, cada um dos passos é discutido.

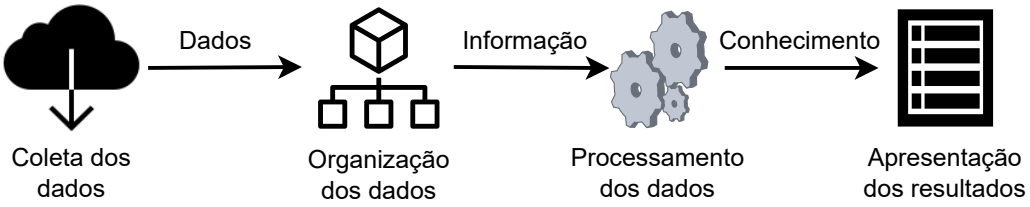


Figura 1 – Metodologia

**E1 Coleta dos Dados.** A coleta de dados corresponde à obtenção dos dados brutos, que pode ocorrer a partir de diversas fontes. Um tipo de fonte inclui as plataformas de gerenciamento de currículos, como a Plataforma Lattes, ou de acompanhamento dos programas de pós-graduação, como a Sucupira. Desse modo, a origem poderá ser a Plataforma Sucupira combinada com os currículos Lattes de cada docente vinculado aos programas de pós-graduação. Também há a alternativa de obter os dados de repositório aberto, como o sistema de dados abertos do governo federal[1]. Outra opção é obter os dados pré-processados de plataformas avançadas, onde as informações de cada programa poderão ser baixadas no formato de planilha eletrônica (p.ex., xls, odf).

**E2 Organização dos Dados.** Nesta etapa é desenvolvido um programa para ler de forma automatizada cada arquivo baixado e armazenar as informações de forma estruturada. Por exemplo, considerando a lista de publicações de todos os professores, qual a média de publicação anual. De acordo com (SILVA; PERES; BOSCARIOLI, 2016) o pré-processamento dos dados consiste em três procedimentos: limpeza de dados, integração de dados, e transformação dos dados. A limpeza de dados tem por objetivo amenizar os problemas decorrentes da existência de valores ruidosos, como dados fora do padrão, e valores ausentes, quando algum atributo não possui informação. A integração dos dados consiste em combinar os dados obtidos de fontes diferentes, pois estes valores podem estar inconsistentes, por estarem com formatação ou padronização diferentes, ou redundantes, quando há repetição de valores para um mesmo atributo. A transformação dos dados consiste em tornar os dados mais adequados para as análises, pois eles podem estar representados em

[1] <dadosabertos.capes.gov.br>

grandezas ou tipos diferentes, o que dificulta o relacionamento entre os atributos. A transformação pode ser realizada pela normalização, que consiste em deixar os dados em uma mesma escala de medida, e pela conversão, com a modificação do formato ou tipo do dado. Já (HAN; KAMBER; PEI, 2011) apresenta mais uma etapa, a redução dos dados, que objetiva diminuir o tamanho de dados muitos grandes mediante técnicas como a redução de dimensionalidade, a redução de quantidade e a compressão de dados. Caso os dados já venham pré-processados, na forma de informação, o esforço se traduz em manipulação para padronização. Nesta etapa podem ser utilizadas bibliotecas de processamento de dados, como Pandas[2], R[3].

**E3 Processamento dos Dados.** Nesta etapa os dados organizados em informação são processados para serem transformados em conhecimento. Por exemplo, dentre diversas métricas como número de alunos formados em determinado ano, ou quantidade de publicações por professor, quais são mais relevantes para a nota do programa. Para esta tarefa pode ser utilizada a aplicação Weka[4], que é uma plataforma de código aberto e gratuita, escrita em linguagem Java, e contempla uma grande coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados. A Weka oferece uma variedade de recursos para pré-processar dados, como filtragem, seleção de atributos, transformação de dados, discretização e normalização. Também possui técnicas de mineração de dados, como classificação, regressão, agrupamento, associação, e uma ampla seleção de algoritmos, incluindo árvores de decisão, redes neurais, Naïve Bayes, SVM (*Support Vector Machines*), k-means e apriori (FRANK; HALL; WITTEN, 2016). Outra opção para o processamento dos dados é o desenvolvimento de uma aplicação na linguagem Python, utilizando alguma técnica de aprendizado profundo, como redes neurais. Para isso, podemos utilizar alguma biblioteca voltada para o aprendizado de máquina, como PyTorch[5], TensorFlow[6] e Keras[7]. A seguir são abordadas diversas técnicas de inteligência artificial que podem ser exploradas nesta etapa do processo.

A classificação é um processo que prevê rótulos de classes e classifica os novos dados com base em um conjunto de treinamento (NIKAM, 2015). Uma das técnicas utilizadas neste processo são as redes neurais, que são ferramentas não lineares de modelagem estatística de dados (KESAVARAJ; SUKUMARAN, 2013), sendo um modelo computacional baseado em redes neurais biológicas (BENIWAL; ARORA, 2012). Outro algoritmo utilizado é o Naïve Bayes, que é um classificador estatístico baseado no Teorema de Thomas Bayes (SILVA; PERES; BOSCARIOLI, 2016). O

[2] <pandas.pydata.org/>
[3] <www.r-project.org>
[4] <www.cs.waikato.ac.nz/ml/weka/>
[5] <pytorch.org>
[6] <www.tensorflow.org>
[7] <keras.io>

termo Naïve (ingênuo) faz referência à independência de cada atributo em determinar à qual classe o item pertence (ARCHANA; ELANGOVAN, 2014). O SVM (*Support Vector Machines*), é uma técnica para classificar dados lineares e não lineares, usando um mapeamento não linear para transformar os dados do treinamento em uma dimensão mais alta, e nesta dimensão é buscado o hiperplano ótimo de separação linear (HAN; KAMBER; PEI, 2011). Outra técnica comumente utilizada é a árvore de decisão, que consiste em uma árvore onde decisões são tomadas a cada nó, semelhante à estrutura de dados do tipo árvore (SILVA; PERES; BOSCARIOLI, 2016). Cada nó corresponde a uma característica em uma instância a ser classificada, e cada ramo representa um valor que o nó pode assumir (PHYU, 2009). A regressão tem por objetivo estimar valores a partir de um conjunto histórico, podendo ser linear se a função utilizada representar uma reta, ou não linear se representar uma equação exponencial (SILVA; PERES; BOSCARIOLI, 2016).

O agrupamento (ou *clustering*), é um processo que permite descobrir relações entre os exemplares de um conjunto de dados por meio de suas características. Diferentemente da classificação, não é feito o uso de rótulos de classes (SILVA; PERES; BOSCARIOLI, 2016). Um dos algoritmos utilizados para esta tarefa é o k-means, que possui a estratégia de agrupamento por partição, onde k é o número de partições, e o conjunto de dados é agrupado nestas partições de acordo com suas características. Posteriormente, cada partição (ou centroide) é atualizada para representar a média dos valores a eles agrupados (SILVA; PERES; BOSCARIOLI, 2016). O k-means é eficiente no processamento de grandes conjuntos de dados, e funciona somente em valores numéricos (KAMESHWARAN; MALARVIZHI, 2014).

**E4 Apresentação de Dados.** A apresentação dos resultados corresponde ao produto deste trabalho e serve para que os profissionais da gestão possam embasar decisões estratégicas. O processamento dos dados pode gerar, por exemplo, uma tabela ordenada por prioridade, ou um gráfico de calor com os dados bibliométricos que mais influenciam no conceito de um programa de pós-graduação. Como exemplo de ferramentas para visualização de dados podemos citar o Microsoft Power BI[8], o Looker Studio[9], e o Tableau[10].

[8] <powerbi.microsoft.com>
[9] <lookerstudio.google.com>
[10] <www.tableau.com>

# 3 TRABALHOS RELACIONADOS

A obtenção de informações para apoiar decisões sofreu grande evolução nas últimas décadas, demonstrada na Figura 2. Durante a década de setenta as decisões eram tomadas com base em relatórios estruturados e periódicos, contendo apenas informações de determinado período que já aconteceu (SHARDA; DELEN; TURBAN, 2019). Com o passar das décadas, os sistemas computacionais evoluíram e passaram a fazer uso de técnicas de inteligência artificial, trazendo informações não apenas sobre o passado, mas prevendo o futuro.

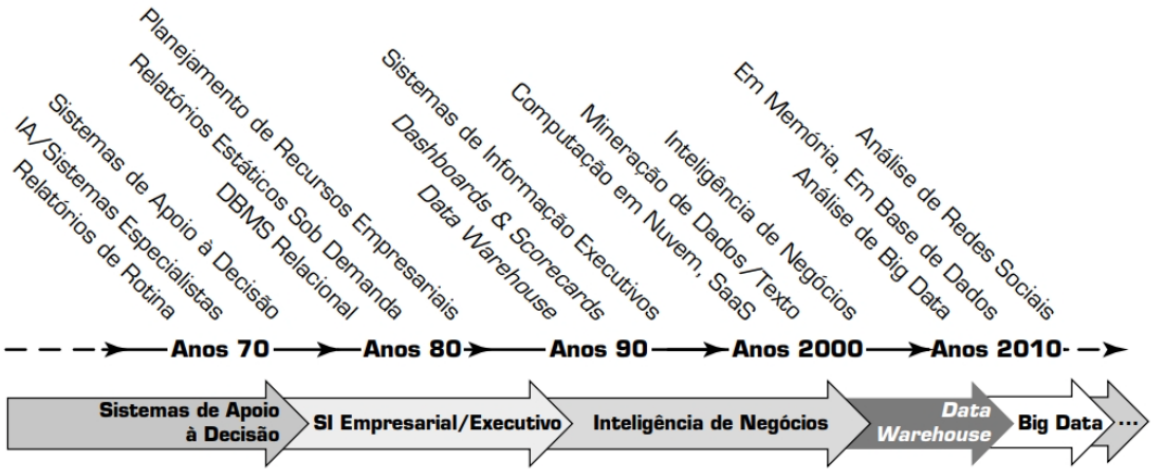


Figura 2 – Evolução da Informação (SHARDA; DELEN; TURBAN, 2019)

Muitos trabalhos abordam o uso de inteligência artificial para transformar informação em conhecimento para tomada de decisões, como (SANTOS; COSTA, 2023) que utiliza aprendizado de máquina para tomada de decisões no mercado de ações, (LOPES; PINTO; BRITO, 2020) implementou um sistema para auxiliar na triagem de pacientes com suspeita de COVID-19, e (ULMANN et al., 2021) desenvolveu um sistema de recomendação de produtos para e-commerce.

Alguns trabalhos são mais relacionados com o contexto de avaliação de grupos de pesquisa e ensino em nível de pós-graduação. A Tabela 1 resume os trabalhos considerados mais relacionados, e é seguida de uma discussão aprofundada sobre cada um deles.

Em (CAPARELLI; DIGIAMPIETRI, 2018), os autores analisaram 66 programas acadêmicos de pós-graduação em Ciência da Computação, visando analisar os dados bibliométricos juntamente com redes sociais de co-autoria em publicações a fim de verificar se os indicadores influenciam na nota obtida pela CAPES, e ainda classificar a importância destes dados. Os autores obtiveram manualmente pela Plataforma Sucupira as informações básicas de cada programa, sendo elas: nome, conceito CAPES atual e anterior, e a lista de docentes. Após, foram analisados de maneira automáticas os currículos Lattes de cada docente vinculado ao curso e também foram verificados os perfis de cada um no Google Acadêmico, de forma automática, para verificar os índices referentes a citações.

Tabela 1 – Trabalhos relacionado.

| **Título (Referência)** | **Fonte** | **Organização** | **Técnica** | **Visualização** |
|---|---|---|---|---|
| Combinando Análise Bibliométrica e Análise de Redes Sociais para a Avaliação de Grupos Acadêmicos (CAPARELLI; DIGIAMPIETRI, 2018) | Sucupira, Lattes, Google Acadêmico | SGBD | Análise de Redes Sociais, Weka | Tabela |
| Um Estudo sobre os Impactos dos Relacionamentos Sociais na Avaliação do Mérito Científico (CALISTO; NóBREGA, 2013) | Lattes | - | Análise de Redes Sociais | Tabela |
| Aplicação de Business Intelligence no Processo de Autoavaliação de Instituições de Ensino Superior (SILVA et al., 2022) | Questionário online | Base de dados do questionário | Inteligência de Negócios | Google Data Studio |
| Mineração de Dados Educacionais: uso de redes neurais artificiais na predição do Perfil Acadêmico do Aluno - IFAL Campus Maragogi (SILVA; CRUZ, 2019) | Relatórios | SGBD | Redes Neurais Artificiais | - |
| BraX-Ray: An X-Ray of the Brazilian Computer Science Graduate Programs (DIGIAMPIETRI et al., 2014) | Lattes | SGBD | Análise de Redes Sociais | Tabelas, Grafos, Gráficos de calor |

Com a obtenção dos dados bibliométricos, estes foram processados através do arcabouço Weka, utilizando o método baseado no valor qui-quadrado, que consiste em, dados dois atributos, é verificado o quanto um implica em outro (HAN; KAMBER; PEI, 2011). Para a validação dos resultados foi utilizada uma Matriz de Confusão, que retornou um resultado de 86,15% de acertos.

O trabalho de (CALISTO; NóBREGA, 2013) utilizou métricas de análise de redes sociais para identificar se há diferença nas métricas entre bolsistas de produtividade em pesquisa e os demais, e também se há diferença nas métricas entre programas de pós-graduação nível 5, 6 e 7 e programas de pós-graduação nível 3 e 4, na área de Ciência da Computação. Os dados para as análises foram obtidos através dos currículos Lattes e processados em uma ferramenta de análise de redes sociais[1]. O trabalho demonstra que as redes dos bolsistas PQs são mais ativas, e também que quanto maior o conceito CAPES, mas ativa a rede social tende a ser.

Já (SILVA et al., 2022) aborda o tema da inteligência de negócios, onde os autores obtiveram os dados através dos formulários de autoavaliação da instituição e os processaram no Google Data Studio[2]. A ferramenta gera gráficos e relatórios que poderão ser utilizados pela instituição para tomada de decisões.

No trabalho de (SILVA; CRUZ, 2019) buscou-se identificar as possíveis fragilidades de alunos, a fim obter informações que auxiliem a gestão a adotar medidas que diminuam a evasão. Os dados utilizados foram obtidos de relatórios arquivados e registros acadêmicos e posteriormente estruturados em um Sistema Gerenciador de Banco de Dados (SGBD). No processamento dos dados foi aplicada a técnica de Redes Neurais Artificiais (RNA) multicamadas, utilizando a ferramenta estatística R. O trabalho conclui que, ao utilizar aprendizado profundo com retropropagação em três camadas ocultas internas, obtêm-se

[1] <gephi.org>
[2] <lookerstudio.google.com>

uma precisão de 93% de acerto para prever o coeficiente acadêmico de novos alunos.

(DIGIAMPIETRI et al., 2014) buscaram caracterizar os programas de pós-graduação em Ciência da Computação e relacioná-los por meio da produtividade científica, através de indicadores bibliográficos. Foram avaliados 37 programas acadêmicos, com base na produção bibliográfica de cada docente vinculado aos cursos. A relação de docentes foi obtida por meio de relatórios da CAPES, e para cada um foi obtido o identificador Lattes para baixar o currículo de maneira automatizada. Os dados dos currículos foram pré-processados e armazenados em um banco de dados relacional. Posteriormente estes índices foram utilizados para classificar a produtividade científica dos programas de pós-graduação brasileiros. Também foi realizada uma análise de redes sociais, para verificar a rede de colaboração entre os docentes de diferentes programas.

# 4 SOLUÇÃO PROPOSTA

Este capítulo apresenta uma solução técnica para responder à pergunta sobre como realizar sistematicamente o processo de identificação das métricas de desempenho com maior correlação com Conceito alto da CAPES. Primeiro é detalhado como os dados são obtidos, depois como eles são organizados e normalizados, e também como é realizado o processamento das informações e obtenção da lista de indicadores mais impactantes no conceito obtido pelas CAPES. A Figura 3 demonstra de forma detalhada a metodologia utilizada no desenvolvimento do software, denominado PPGEva.

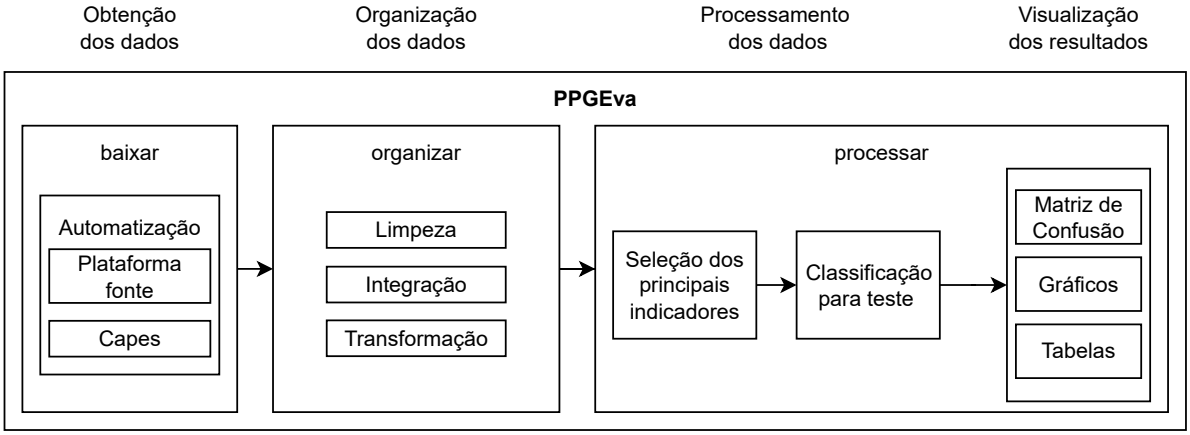


Figura 3 – Metodologia Detalhada

## 4.1 Coleta dos dados

A obtenção dos indicadores é realizada por meio de uma plataforma *web*. Esta ferramenta fornece indicadores e relatórios da produção científica dos programas de pós-graduação, por meio da extração de informações das Plataformas Sucupira e Lattes. Este ambiente permite comparar os indicadores do curso de pós-graduação ao qual o usuário está registrado com os demais cursos do país, sendo possível exportar esta comparação para uma planilha eletrônica. Para automatizar esta tarefa e fazer o *download* de todos os indicadores dos demais cursos, foi utilizada a ferramenta Selenium[1], que possui uma biblioteca que pode ser importada para várias linguagens de programação, incluindo o Python. Com ela foi possível simular ações de usuário e navegar pelo *site*. A Figura 4 demonstra como os dados são obtidos.

Para realizar o *download* das planilhas referentes a todos os cursos de pós-graduação na área de Ciência da Computação, foi desenvolvido um programa em Python chamado baixar.py. Nele, foram importadas as bibliotecas necessárias do Selenium e também o gerenciador de *driver* do Google Chrome. Este gerenciador é crucial para o programa desenvolvido utilizar o Google Chrome no acesso a plataformas *web*. Como nos primeiros

[1] <https://www.selenium.dev/>

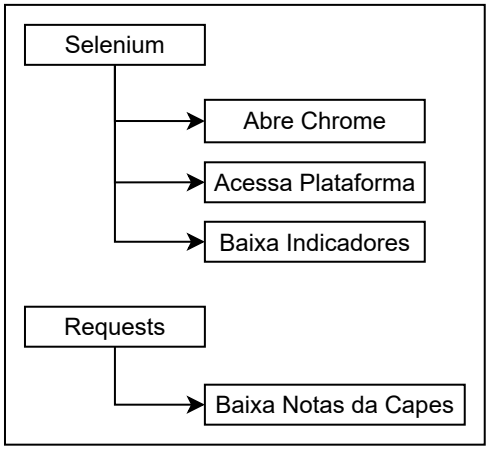


Figura 4 – Obtenção dos Dados

testes ocorreram incompatibilidade entre a versão do gerenciador instalada e do Google Chrome, e, também buscando manter o código atualizado e compatível para todos os usuários, o próprio programa desenvolvido realiza a instalação do gerenciador compatível com a versão do navegador do usuário. O código possui os endereços XPath (XML Path Language) dos elementos a serem localizados e manipulados na página, e os comandos simulando as ações do usuário. Como entrada para o programa, o usuário deve fornecer como parâmetro o e-mail e senha do usuário. Como parâmetros opcionais, podem ser fornecidos os referentes ao filtro demonstrado na Figura 5.

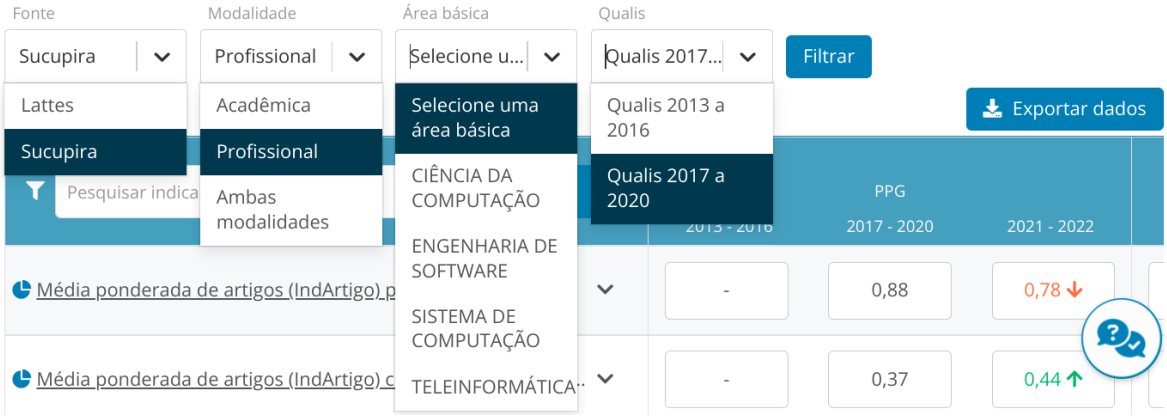


Figura 5 – Exemplo de Tela da Plataforma

Por padrão é utilizada como fonte a Sucupira, modalidade profissional e qualis 2017 a 2020. A área básica foi mantida como opção de escolha, mas ela não interfere no resultado filtrado. Para obter os indicadores de outros programas, é necessário comparar o curso ao qual o usuário tem acesso no sistema com os demais cursos. Assim escolhe-se o curso que deseja obter as informações, conforme Figura 6, e só então é possível baixar o conteúdo em uma planilha eletrônica.

Apesar de na lista suspensa constar a nota dos programas, essa informação não é

salva no arquivo. Então, o programa desenvolvido obtém e salva o conteúdo desta lista em um arquivo de texto, pois esta informação serve como rótulo da amostra que servirá para treinar o software e permitir avaliar as escolhas de métricas.

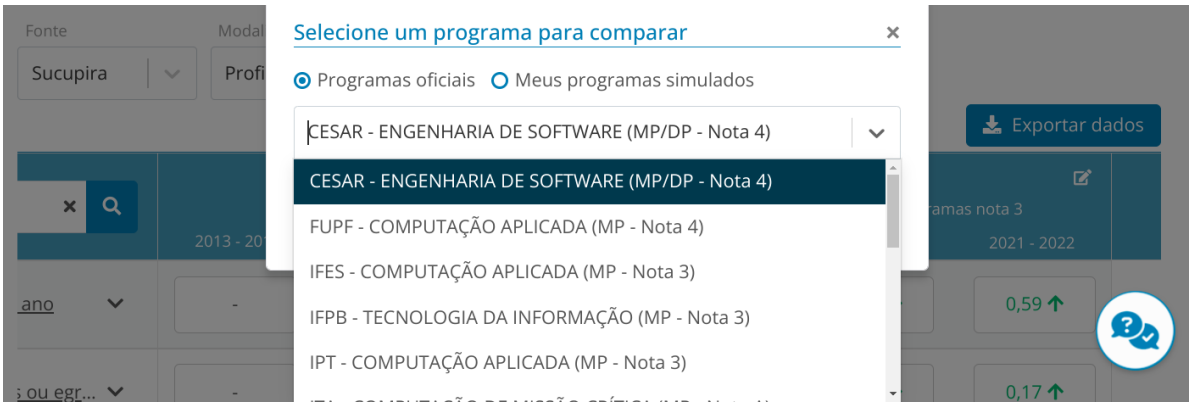


Figura 6 – Exemplo de Lista Suspensa dos Programas

Uma dificuldade encontrada neste processo foi a interação com as listas suspensas, devido à maneira que o site foi desenvolvido. O Selenium possui o objeto Select[2], que permite interagir com listas suspensas do tipo Select. Como o *dropdown* da página foi implementado como uma lista de elementos div, e não como select, foi necessário criar uma estratégia para iterar sobre todos os elementos. Sendo assim, foi utilizado o comando find_elements para capturar e armazenar todas as opções da lista suspensa em uma lista. Apesar de constar na documentação do Selenium[3] ser possível iterar sobre esta lista, no caso da página utilizada não foi possível, pois todos os elementos foram armazenados como um único item na lista. Para contornar este problema, esta lista com um único item contendo todos os elementos teve seu texto salvo em um arquivo de texto. Após, este arquivo foi lido e seu conteúdo armazenado em uma lista. Com o tamanho da lista descobrimos quantas iterações devemos fazer para acessar todos os itens da lista.

Para percorrer os elementos das listas suspensas, foi utilizado o objeto ActionChains para simular ações do teclado. Com isso, após a exibição da lista suspensa, o programa simula a pressão na tecla para baixo uma vez para selecionar o próximo item. Depois Enter é pressionado de maneira simulada. Esse loop é realizado conforme o tamanho da lista. Como na próxima iteração com a lista suspensa a escolha anterior já está selecionada, basta selecionar a próxima opção abaixo. Assim, todos os itens da lista suspensa podem ser acessados.

Por padrão, o Selenium utiliza as configurações predefinidas do navegador. Desse modo, todos os *downloads* são salvos automaticamente na pasta *Downloads* do computador. Então, após a execução do programa baixar.py, os arquivos são movidos manual-

[2] <https://www.selenium.dev/documentation/webdriver/support_features/select_lists/>
[3] <https://www.selenium.dev/documentation/webdriver/elements/finders/#get-element>

mente para uma pasta específica para eles, que será acessada posteriormente nas próximas etapas.

Para a obtenção dos conceitos de cada programa foi necessário buscar outras fontes, pois a plataforma utilizada fornece somente as notas atuais dos programas. Mas também são necessários os conceitos dos quadriênios anteriores, sendo eles de 2013 a 2016, e de 2017 a 2021. No planejamento inicial seriam utilizadas planilhas obtidas através da plataforma Dados Abertos CAPES[4]. Mas observou-se que as planilhas disponibilizadas não estão atualizadas. Como, por exemplo, a nota do PPGES da Unipampa, que na planilha referente ao período 2021–2022 apresenta o conceito A, mas na realidade possui conceito 3. Devido a esta inconsistência, optou-se por descartar a utilização da plataforma Dados Abertos. Como alternativa, foi utilizado o site da CAPES[5] para obter estas planilhas. Esses dados também são obtidos de maneira automatizada, sendo que os links das planilhas foram incluídos no código do programa. As notas da avaliação 2013–2016[6] são obtidas da página de divulgação desta quadrienal, divididas em dois arquivos, um referente aos programas profissionais e outro referente aos acadêmicos. Já o resultado da avaliação 2017–2020[7] são obtidos em um único arquivo, também disponibilizado na página referente a esta avaliação.

## 4.2 Organização dos dados

Nesta etapa, as planilhas baixadas são lidas e suas informações armazenadas em planilhas finais. É gerada uma planilha para cada quadriênio, sendo composta pela relação de programas, seus conceitos obtidos na quadrienal, e seus indicadores bibliométricos. Ocorre também o tratamento dos dados, para deixar a planilha apta a ser utilizada na etapa do processamento. A Figura 7 demonstra como é realizada a organização dos dados.

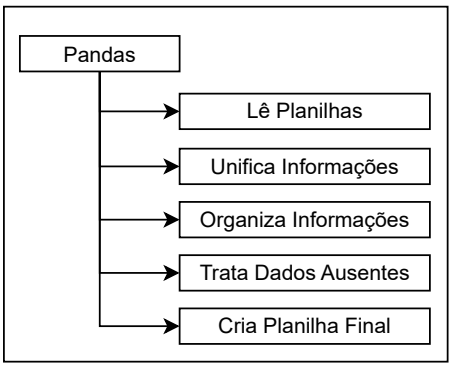


Figura 7 – Organização dos Dados

[4] <https://dadosabertos.capes.gov.br/>
[5] <https://www.gov.br/capes/pt-br>
[6] <https://www.gov.br/capes/pt-br/acesso-a-informacao/acoes-e-programas/avaliacao/avaliacao-quadrienal-2017/resultados/resultado-da-avaliacao-quadrienal-2017>
[7] <https://www.gov.br/capes/pt-br/acesso-a-informacao/acoes-e-programas/avaliacao/avaliacao-quadrienal/resultado-da-avaliacao-quadrienal-2017-2020>

Para organizar os dados foi desenvolvido um programa em Python chamado organizar.py, utilizando a biblioteca Pandas[8], que é uma ferramenta de análise e manipulação de dados, construída sobre a linguagem de programação Python (PANDAS, 2023). O programa recebe como parâmetro o endereço da pasta onde as planilhas com os indicadores de cada curso estão armazenadas. Dentro desta pasta é verificada a existência de outra pasta, chamada dados. Caso não exista, ela é criada. É nesta nova pasta onde as planilhas finais são armazenadas.

| | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| | PPG | | | CESAR - ENGENHARIA DE SOFTWARE | | |
| INDICADOR | 2013 - 2016 | 2017 - 2020 | 2021 - 2022 | 2013 - 2016 | 2017 - 2020 | 2021 - 2022 |
| Média ponderada de artigos (IndArtigo) por DPs e por ano | - | 0,88 | 0,78 | 0,04 | 0,03 | 0,04 |
| Variação percentual | - | - | -11,75% | - | 2833,33% | 1850,00% |
| Média ponderada de artigos (IndArtigo) com discentes ou egressos por DPs e por ano | - | 0,37 | 0,40 | 0,00 | 0,00 | 0,00 |
| Variação percentual | - | - | 8,60% | - | 37,00% | 40,00% |
| Indicador considerando apenas discentes | - | 0,37 | 0,40 | 0,00 | 0,00 | 0,00 |
| Indicador considerando apenas egressos registrados na Sucupira | - | 0,00 | 0,13 | 0,00 | 0,00 | 0,02 |
| Indicador considerando apenas egressos identificados pelo sistema | - | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| % do IndArtigo dos 30% dos DPs mais produtivos* | - | 72,75 | 70,51 | 100,00 | 100,00 | 100,00 |
| Variação percentual | - | - | 3,18% | - | 37,46% | 41,82% |
| % do IndArtigo dos 50% dos DPs mais produtivos* | - | 85,86 | 85,16 | 100,00 | 100,00 | 100,00 |
| Variação percentual | - | - | 0,82% | - | 16,47% | 17,43% |
| Média de artigos A1 a A4 dos DPs por ano | - | 1,45 | 1,36 | 0,04 | 0,03 | 0,06 |
| Variação percentual | - | - | -6,33% | - | 4733,33% | 2166,67% |
| Média de artigos A1 a A4 no PPG por ano (e por DPs) | - | 0,95 | 0,86 | 0,04 | 0,03 | 0,06 |
| Variação percentual | - | - | -10,00% | - | 3066,67% | 1333,33% |

Figura 8 – Exemplo de Planilha com indicadores

A Figura 8 apresenta o exemplo do conteúdo de cada arquivo, onde a primeira coluna contém os nomes dos indicadores, alinhados à esquerda, e indicadores complementares alinhados à direita. As colunas B, C e D possuem os indicadores relacionados ao PPG (Programa de Pós-Graduação) ao qual o usuário do sistema tem acesso. Já as colunas E, F e G apresentam os índices do PPG que queremos obter suas informações. Desse modo, as primeiras quatro colunas devem ser lidas somente na primeira iteração do programa, e as três restantes devem ser lidas em todas as iterações. Na primeira iteração também é realizada a verificação do alinhamento dos dados da primeira coluna, com o uso da biblioteca openpyxl[9]. E com o auxílio de um contador, os números das linhas com alinhamento à direita são armazenados em uma lista. Isso permite realizar análises considerando apenas os indicadores principais, ou considerando todos os indicadores. Para armazenar as informações lidas, são utilizadas três listas, uma para cada período da planilha. Ao final de cada *loop* os dados lidos são incluídos nessas listas.

Depois de todos os arquivos serem lidos, cada lista é concatenada lado a lado em um *data frame*. A última linha, que possui uma informação desnecessária, é excluída. E todas as ocorrências de - são substituídas por 0. Tentou-se substituir - por valores nulos, por meio da biblioteca Numpy[10], mas isso causou erros na etapa de processamento. Então optou-se por atribuir zero aos campos inexistentes.

[8] <https://pandas.pydata.org/>
[9] <https://openpyxl.readthedocs.io/en/stable/>
[10] <https://numpy.org/>

Ao final, os *data frames* gerados são duplicados, e nestas cópias são desconsiderados os indicadores auxiliares, gerando dois arquivos para cada período. Um com todos os indicadores, e outro somente com os indicadores principais.

Posteriormente é realizada a transposição entre as linhas e colunas dos *data frames*, pois, na etapa de processamento as amostras devem ficar organizadas em linhas, e suas características em colunas. Como os nomes dos PPGs são lidos somente no *data frame* referente ao período 2013–2016, é necessário copiar estas informações para os outros dois *data frames*. Esta informação também é copiada para uma lista, que será utilizada para buscar as notas dos programas.

Após ler e organizar as planilhas, é necessário armazenar os conceitos de cada PPG referentes aos três períodos. Para isso, o programa lê os dois arquivos com os conceitos do período 2013–2016, e o referente ao período 2017–2020, salvos na pasta capes do programa, e também o arquivo de texto salvo na mesma pasta das planilhas, referente ao período 2021–2022. Os dados destas planilhas são salvos em *data frames* correspondentes a cada arquivo. Após, a lista contendo os nomes dos programas é percorrida em um laço, para buscar os conceitos de todos os PPGs. Onde cada *loop* corresponde à verificação de uma linha do *data frame*. Como cada item da lista contém a sigla da instituição e o nome do programa, a cada iteração esta informação é dividida e armazenada em duas variáveis. Nas planilhas, essa busca é realizada nas planilhas obtidas na página da Capes, comparando a coluna contendo a sigla da instituição com a sigla armazenada da lista, e a coluna correspondente ao nome do programa com o nome armazenado na lista. Se as duas comparações forem verdadeiras, a linha encontrada na planilha é armazenada em uma variável. Então o conteúdo corresponde à coluna do conceito desta variável é armazenado no campo específico do *data frame* correspondente ao período pesquisado. Como na planilha disponibilizada pela Capes não há um padrão na caixa do texto, sendo algumas células em caixa alta e outras em caixa baixa, foi necessário deixar toda a coluna referente aos nomes em caixa alta, com str.upper. Também foi necessário tratar espaçamento duplicado com str.replace, pois na planilha obtida há um caso detectado de nome separado por dois espaços.

Para pesquisar no documento de texto, foi utilizada a biblioteca re[11] para expressões regulares do Python. Para isso foi especificado um padrão de texto para ser possível extrair a nota do documento correspondente à instituição e programa.

Nos quadriênios 2013–2016 e 2017–2020, alguns PPGs existentes atualmente ainda não tinham sido criados. Desse modo, a busca por suas notas nesses períodos não obteve resultados. Quando isso ocorre, um contador é utilizado para verificar sua linha correspondente no *data frame*, e seu valor é armazenado em uma lista. Sendo que, ao salvar a planilha final, as linhas correspondentes aos programas sem nota são desconsideradas.

Uma dificuldade encontrada é quando uma instituição possui mais de um PPG com

[11] <https://docs.python.org/3/library/re.html>

o mesmo nome, causando erro ao armazenar o conceito. Pois, a busca sempre retorna a primeira ocorrência encontrada. Neste caso, não foi possível contornar o problema de forma automatizada, e o usuário deve realizar a correção de maneira manual. Para isso, o programa identifica as ocorrências duplicadas e gera um aviso em tela com a lista de inconsistências.

## 4.3 Processamento dos dados

Para o processamento dos dados, foi desenvolvido um programa na linguagem de programação Python chamado processar.py, e utilizada a biblioteca Scikit-learn[12], que possui ferramentas para análise preditiva de dados. Como parâmetros, o usuário deve fornecer o endereço do arquivo que será analisado, a técnica de seleção das características, e o classificador que será utilizado para teste. Também é possível selecionar a quantidade de atributos que devem ser selecionados, o grau de informação que será exibido em tela durante a execução, e o local onde os resultados serão armazenados. A Figura 9 apresenta a modelagem do processamento dos dados.

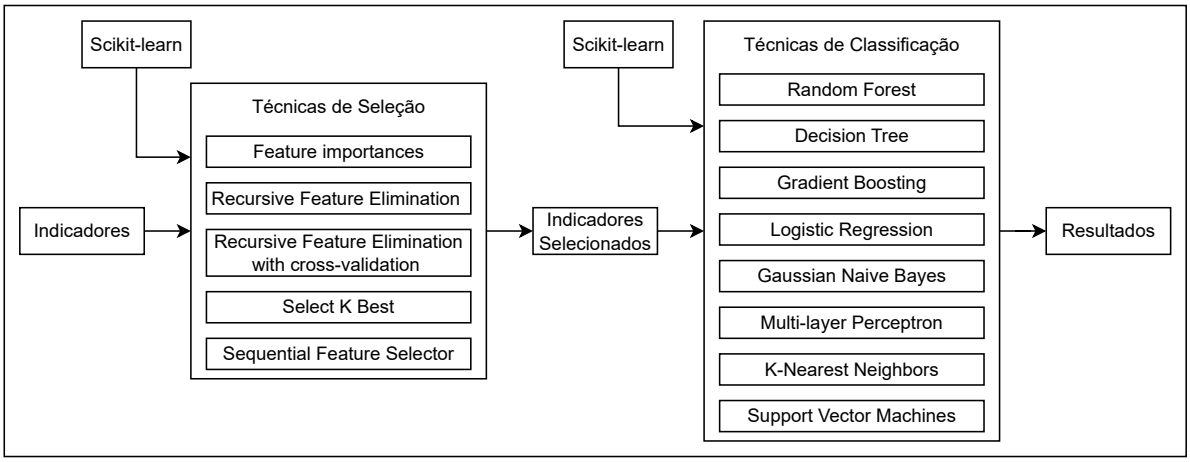


Figura 9 – Processamento dos Dados

Neste programa foram utilizadas cinco técnicas de seleção disponíveis no Scikit-learn, apresentadas na Tabela 2 com descrições obtidas em seu guia de usuário (PEDREGOSA et al., 2011). O Scikit-learn possui várias técnicas de seleção de atributos, e escolher quais seriam utilizadas é um desafio. Feature Importances, RFE e RFEC são utilizadas por já serem atributos das técnicas Random Forest, Decision Tree, Gradient Boosting e Logistic Regression. A definição da técnica Select K Best supõe que ela seja adequada para o objetivo deste trabalho, pois consta que seleciona uma quantidade determinada de atributos. A técnica Sequential Feature Selector é mais robusta, por realizar várias combinações com validação cruzada e por ter a opção de selecionar automaticamente a quantidade ideal de atributos.

[12] <https://scikit-learn.org/>

Tabela 2 – Técnicas de Seleção

| **Nome** | **Descrição** (Adaptado da documentação do Scikit-learn (PEDREGOSA et al., 2011), tradução livre.) |
|---|---|
| Feature Importances | Calcula a importância de cada atributo utilizando o índice de Gini. O programa desenvolvido seleciona as que possuem maior importância conforme especificado na execução. |
| RFE (Recursive Feature Elimination) | Elimina recursivamente os atributos de menor importância, utilizando um estimador externo que atribui pesos às características, como o Feature Importances. Retorna a quantidade especificada na execução. |
| RFEC (Recursive Feature Elimination with Cross-Validation) | Executa RFE em um loop de validação cruzada para encontrar o número ideal de *features*. |
| Select K Best | Remove tudo menos as k *features* de pontuação mais alta, usando o teste estatístico ANOVA. |
| Sequential Feature Selector | Realiza combinações entre as *features*. Em cada estágio escolhe o melhor atributo para adicionar ou remover com base na pontuação da validação cruzada. Pode retornar a quantidade ideal de atributos, ou a informada na execução. |

Como classificadores são utilizadas oito técnicas, também já incluídas na biblioteca Scikit-learn. A utilização delas justifica-se por serem comumente citadas em artigos científicos, como em (HISHAMUDDIN et al., 2020) e em (KESAVARAJ; SUKUMARAN, 2013). Uma breve descrição extraída do guia do usuário (PEDREGOSA et al., 2011) do Scikit-learn define cada uma na Tabela 3

Inicialmente o programa carrega o arquivo fornecido para um *data frame* e atribui a duas variáveis o seu conteúdo. Uma variável representa as *features*, armazenando todo o conteúdo do *data frame*, ignorando-se as colunas referentes aos nomes dos PPGs e ao conceito. A outra variável representa as classes, sendo o conteúdo referente à coluna das notas dos PPGs. Para a variável das classes foi necessário converter seu conteúdo para o tipo *string*, caso contrário os classificadores acusam erro.

Posteriormente é realizada a seleção dos atributos mais influentes nos conceitos atribuídos pela Capes. Como algumas técnicas de seleção dependem de um classificador, realiza-se um teste *if* para selecionar qual classificador deve ser utilizado. Com isso, uma função chamada impactos_indicadores é solicitada, retornando um *data frame* contendo somente as características selecionadas. Esta função verifica qual técnica de seleção deve ser utilizada e aplica a técnica necessária.

Para verificar a confiabilidade dos atributos selecionados é realizado processo de classificação sistemático usando um método de validação cruzada com cinco dobras (80% para treinamento e 20% para classificação). Para isso foi utilizado o método StratifiedK-

Tabela 3 – Técnicas de Classificação

| **Nome** | **Descrição** (Adaptado da documentação do Scikit-learn (PEDREGOSA et al., 2011), tradução livre.) |
|---|---|
| Random Forest | Uma floresta aleatória é um meta-estimador que ajusta vários classificadores de árvore de decisão em várias subamostras do conjunto de dados e usa a média para melhorar a precisão preditiva e controlar o ajuste excessivo. |
| Decision Tree | O objetivo é criar um modelo que preveja o valor de uma variável alvo, aprendendo regras de decisão simples inferidas a partir dos recursos dos dados. |
| Gradient Boosting | Este algoritmo constrói um modelo aditivo de maneira progressiva, fase por fase; ele permite a otimização de funções de perda diferenciáveis arbitrárias. Em cada fase, são ajustadas árvores de regressão n_classes_ com base no gradiente negativo da função de perda. |
| Logistic Regression | Implementa regressão logística regularizada. |
| Gaussian Naive Bayes | Baseada no teorema de Bayes, que assume independência entre os recursos. |
| K-Nearest Neighbors | A classificação é calculada a partir de uma votação por maioria simples dos vizinhos mais próximos de cada ponto. |
| Support Vector Machines | Busca encontrar um hiperplano de decisão ótimo. |
| Multi-layer Perceptron | Este modelo otimiza a função de perda logarítmica usando LBFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) ou descida de gradiente estocástica. |

Fold, que procura manter um equilíbrio entre as classes para cada amostra. A cada dobra uma função chamada classificador é utilizada. Ela é responsável por realizar o treinamento das amostras e por aplicar a classificação na amostra de testes.

## 4.4 Visualização dos dados

Para a visualização dos resultados, foram utilizadas as bibliotecas matplotlib.pyplot, e csv. O mesmo programa que realiza o processamento dos dados também é responsável por apresentar os resultados. A Figura 9 demonstra as bibliotecas utilizadas em cada etapa.

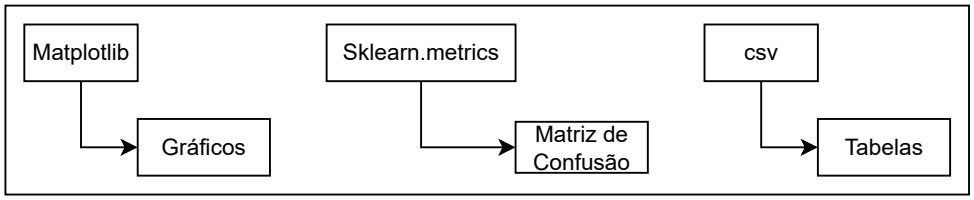


Figura 10 – Visualização dos Dados

Os atributos selecionados são salvos em um documento csv (comma-separated values). A técnica Feature Importances permite também salvar um gráfico demonstrando o percentual de importância de cada atributo referente ao total. Com a técnica RFECV é possível salvar um gráfico contendo a acurácia e seu desvio padrão para cada quantidade de *features* selecionadas no teste. Na etapa de classificação, são geradas matrizes de confusão para cada ciclo de dobras e ao final uma matriz de confusão contendo todas as predições realizadas. A cada execução do programa, um arquivo csv é incrementado com as seguintes informações: técnica de seleção utilizada, classificador, quantidade selecionada, acurácia média e desvio padrão. Este arquivo fornece um resumo da execução, contendo os parâmetros utilizados para a execução, e a confiabilidade do resultado obtido. Um arquivo txt é criado contendo as métricas de cada classe e total: *precision*, *recall*, *f1-score*, *support*, e também a acurácia total. Se for necessária uma análise mais profunda, também são salvas as predições realizadas e os conceitos verdadeiros de cada PPG.

Se o usuário optar, também é possível visualizar o resultado em tempo de execução. Com o auxílio do parâmetro verbosity, é possível escolher entre: não visualizar nada, visualizar apenas as *features* selecionadas e o resultado do teste, ou ainda adicionar a visualização do resultado de cada dobra. No Capítulo 5 serão apresentadas as visualizações resultantes de um estudo de caso.

# 5 ESTUDO DE CASO

Como estudo de caso, o PPGEva foi utilizado no Programa de Pós-Graduação em Engenharia de Software (PPGES), da Universidade Federal do Pampa, que é avaliado pela Capes na área de Ciência da Computação. Objetivando avaliar o comportamento do software e seus resultados. Foram analisados três conjuntos de dados, um considerando somente os PPGs acadêmicos, outro considerando somente os PPGs profissionais, e um terceiro considerando ambos. A Tabela 4 apresenta um resumo com os parâmetros utilizados na realização deste estudo.

Tabela 4 – Parâmetros Utilizados

| **Fase** | **Parâmetros** | **Valores** |
|---|---|---|
| Obtenção dos dados | Fonte | Sucupira |
| | Modalidade | Profissional<br>Acadêmico<br>Ambos |
| | Área de Avaliação | Ciência da Computação |
| | Qualis | 2017 à 2020 |
| Processamento dos dados | Dobras para testes | 5 |
| | Técnicas de Seleção | Feature Importances<br>RFE<br>RFECV<br>Select K Best<br>Sequential Feature Selector |
| | Classificador | Random Forest<br>Decision Tree<br>Gradient Boosting<br>Logistic Regression<br>Gaussian Naive Bayes<br>K-Nearest Neighbors<br>Support Vector Machines<br>Multi-layer Perceptron |
| | Quantidade de Atributos | 10<br>20<br>50<br>auto |

## 5.1 Metodologia

Para o *download* das planilhas é utilizado o programa desenvolvido baixar.py, sendo fornecidos os parâmetros responsáveis por selecionar o seguinte filtro: fonte Sucupira, área de avaliação Ciência da Computação, qualis 2017 a 2020, e modalidade. Como foram analisados três conjuntos de dados, é necessário realizar três execuções do programa, cada uma fornecendo um parâmetro diferente referente à modalidade,

sendo eles, Profissional, Acadêmico e Ambos. Após cada execução, as planilhas baixadas são movidas manualmente para uma pasta interna do programa desenvolvido, chamada dados. Estas pastas foram nomeadas como Sucupira_Academico_CC_2017-2020, Sucupira_Profissional_CC_2017-2020 e Sucupira_Ambos_CC_2017-2020.

A organização dos dados é realizada pelo programa organizar.py, sendo fornecido como parâmetro o caminho para a pasta onde as planilhas estão armazenadas. Da mesma forma que na etapa anterior, o programa precisou ser executado três vezes, cada uma com um endereço de pasta diferente. Após leitura e organização dos dados foram geradas 6 planilhas, armazenadas em uma pasta chamada dados, interna à pasta especificada como parâmetro. As planilhas foram nomeadas como conceitos_2013_2016, conceitos_2013_2016_principais, conceitos_2017_2020, conceitos_2017_2020_principais, conceitos_2021_2022, e conceitos_2021_2022_principais. Para este estudo é utilizada a planilha conceitos_2017_2020_principais, pois contém os conceitos atribuídos pela Capes no último quadriênio, sendo o ciclo completo mais atual. Não foram considerados sub-indicadores, pois a maioria destes refere-se somente à variação do indicador de um quadriênio para outro. Desse modo, as planilhas utilizadas possuem 115 atributos. A referente à modalidade acadêmica possui 66 amostras, e à Profissional possui 12 amostras. Sendo que a planilha com ambas consiste na soma destas amostras.

Para o processamento dos dados foi utilizado o programa processar.py, fornecendo os parâmetros referentes à técnica de seleção de *features*, o classificador utilizado, a quantidade de *features* a serem selecionadas, e o caminho do arquivo a ser analisado. Esta fase também foi dividida em três etapas, uma para cada modalidade avaliada. O processamento consiste em selecionar determinada quantidade de atributos com a técnica selecionada, e posteriormente realizar uma classificação considerando apenas estes atributos. A acurácia obtida determina a confiabilidade da seleção realizada.

Como o programa possui cinco possibilidades de técnicas de seleção e oito de classificadores, onde estas técnicas podem ser combinadas, e, há também diferentes parâmetros para a quantidade de atributos selecionados, foi necessário utilizar uma técnica de automatização dos testes. Para isso, foi escrito um programa em Python chamado executar_campanha.py. Nele são especificados dicionários, e cada chave deste dicionário corresponde a uma lista de parâmetros. A Figura 11 apresenta um exemplo deste dicionário, onde tcc_a é a identificação deste dicionário, e as demais chaves correspondem aos parâmetros que devem ser fornecidos ao programa processar.py. Para executar os testes especificados no dicionário, basta executar o programa executar_campanha.py e fornecer como parâmetro a sua identificação.

O dicionário demonstrado realiza vinte e quatro execuções do programa processar.py, pois realiza todas as combinações entre os parâmetros especificados. Sendo a primeira execução com a técnica de seleção 1, classificar 1, quantidade 10, e a última com a técnica de seleção 2, classificador 4, quantidade 50. Nesse exemplo todas as execuções

```
campaigns_available['tcc_a'] = {
    'arquivo': ['dados/Sucupira_Academico_CC_2017-2020/dados/'
                'conceitos_2017_2020_principais.xlsx'],
    'tecnica_selecao' : ['1', '2'],
    'classificador_para_teste': ['1','2','3','4'],
    'quantidade': ['10','20','50'],
}
```

Figura 11 – Exemplo de especificação do dicionário

foram realizadas com o arquivo referente à modalidade acadêmica. Outros dicionários semelhantes foram especificados, para contemplar todos os algoritmos disponíveis no programa, e considerando 10, 20 e 50 como quantidades.

Para a escolha da quantidade de *features* a serem selecionadas, foi levado em consideração a variância explicada cumulativa, obtida pelo algoritmo PCA (Principal Component Analysis)[1]. A Figura 12 demonstra que 10 atributos representam quase a totalidade do conjunto referente à modalidade profissional. Para as modalidades acadêmica e ambos, os 20 principais atributos representam 90% da totalidade, e 50 atributos representam quase todo o conjunto.

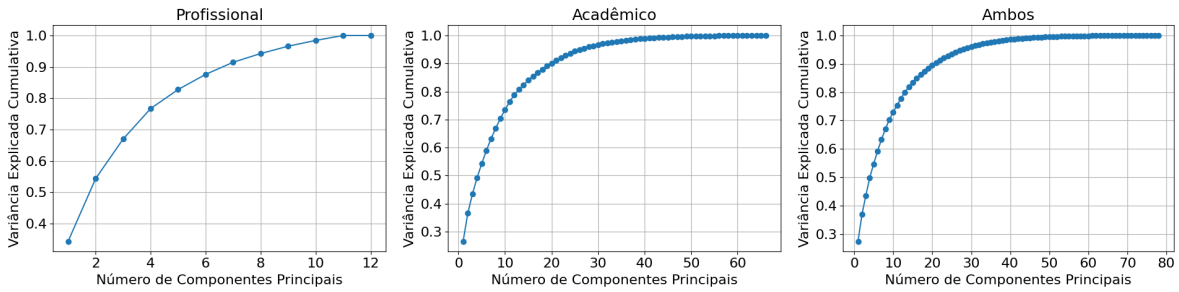


Figura 12 – Variância Explicada Cumulativa

Para verificar quais algoritmos são os que produzem um resultado mais confiável, é necessário armazenar a acurácia de cada combinação e depois analisá-las. Para isso, ao final de cada execução, os parâmetros selecionados, a acurácia média obtida, e seu desvio padrão são armazenados em um arquivo csv. Ao final dos testes esta tabela é classificada em ordem decrescente de acurácia, obtendo-se a melhor combinação de parâmetros.

A Tabela 5 apresenta como exemplo as cinco combinações que obtiveram melhores resultados referentes à modalidade Acadêmica. Onde a técnica de seleção RFE selecionou 10 atributos, e testou-os com o classificador Gradient Boosting, obtendo uma acurácia de 74%.

[1] <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>

Tabela 5 – Combinação de Parâmetros

| **Técnica Seleção** | **Classificador** | **Quantidade** | **Acurácia** | **Desvio Padrão** |
|---|---|---|---|---|
| RFE | Gradient Boosting | 10 | 74% | 11% |
| RFE | Logistic Regression | 10 | 67% | 9% |
| Feature Importances | Gradient Boosting | 10 | 65% | 13% |
| Sequential Feature Selector | Random Forest | auto | 65% | 16% |
| RFECV | Random Forest | auto | 64% | 13% |

## 5.2 Resultados

Após a realização dos testes, pode-se observar quais combinações de parâmetros produzem um resultado mais confiável. Para as três modalidades testadas, os parâmetros de entrada com melhor resultado foram os mesmos. Sendo a técnica de seleção RFE, o classificador Gradient Boosting, e uma quantidade de dez *features* selecionadas.

As Tabelas 6, 7 e 8 apresentam os indicadores selecionados pelo PPGEva, considerando os parâmetros que obtiveram melhor acurácia. Sendo elas correspondentes às modalidades Profissional, Acadêmico e Ambos, respectivamente. Ao lado de cada atributo pode-se observar a importância verificada para cada um em relação aos demais.

Tabela 6 – *Features Importances* - Profissional

| **Nº** | **Atributos Selecionados** | **Peso** |
|---|---|---|
| 1 | Média anual de discentes titulados de mestrado por DP. | 23 % |
| 2 | Média anual ponderada (2D + 1M) de discentes titulados por DP | 21 % |
| 3 | Média anual de discentes titulados por DP. | 19 % |
| 4 | Média de cursos de curta duração dos DPs por ano | 10 % |
| 5 | Média de artigos em jornais ou revistas dos DPs por ano | 10 % |
| 6 | % DP com orientações concluídas de qualquer nível | 5 % |
| 7 | % DP com orientações de mestrado concluídas | 4 % |
| 8 | Média de registros/patentes únicos no PPG por ano (e por DPs) | 3 % |
| 9 | Média de publicações únicas em anais de eventos no PPG por ano (e por DPs) | 3 % |
| 10 | Média de artigos únicos em jornais ou revistas no PPG por ano (e por DPs) | 2 % |

Tabela 7 – *Features Importances* - Acadêmico

| Nº | Atributos Selecionados | Peso |
|---|---|---|
| 1 | Percentual de docentes permanentes (DP) com artigo A2+ (A1 e A2) com discentes ou egressos | 19 % |
| 2 | Docentes permanentes (DPs) por ano | 14 % |
| 3 | Média anual ponderada (2D + 1M) de discentes matriculados por DP | 11 % |
| 4 | Percentual de docentes permanentes (DP) com artigo em periódico com discentes ou egressos | 9 % |
| 5 | Percentual de discentes com artigos A2+ (A1 e A2) | 9 % |
| 6 | Desvio padrão do tempo de titulação no mestrado | 8 % |
| 7 | % DP com orientações de mestrado concluídas | 8 % |
| 8 | Média de livros únicos no PPG por ano (e por DPs) | 8 % |
| 9 | Percentual de artigos A1 a A4 | 7 % |
| 10 | Média anual ponderada (2D + 1M) de discentes titulados por DP | 6 % |

Tabela 8 – *Features Importances* - Ambos

| Nº | Atributos Selecionados | Peso |
|---|---|---|
| 1 | Média de artigos A1 e A2 com discentes ou egressos no PPG por ano (e por DPs) | 17 % |
| 2 | Média anual ponderada (2D + 1M) de discentes titulados por DP | 13 % |
| 3 | Docentes permanentes (DPs) por ano | 13 % |
| 4 | Média anual ponderada (2D + 1M) de discentes matriculados por DP | 12 % |
| 5 | Média do tempo de titulação no mestrado | 9 % |
| 6 | Média anual de discentes titulados de mestrado por DP. | 8 % |
| 7 | % DP com orientações de mestrado concluídas | 8 % |
| 8 | Desvio padrão do tempo de titulação no mestrado | 7 % |
| 9 | Percentual de discentes com artigos A2+ (A1 e A2) | 6 % |
| 10 | Percentual de docentes permanentes (DP) com artigo A2+ (A1 e A2) com discentes ou egressos | 5 % |

A Figura 13 permite visualizar na forma de gráficos o percentual de importância das *features* selecionadas em cada modalidade.

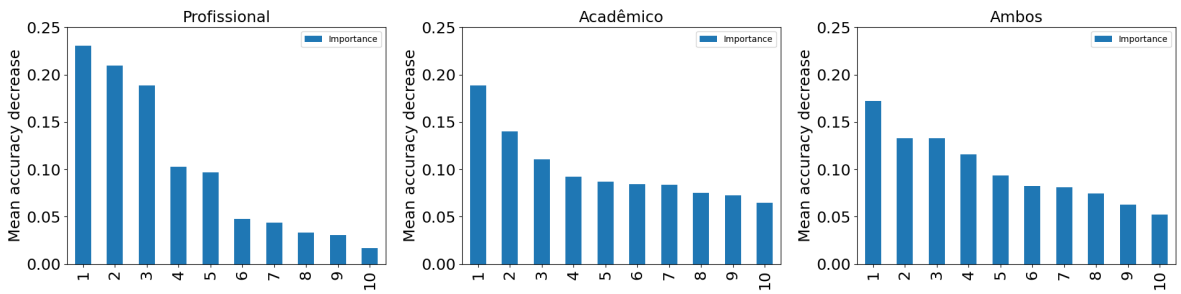


Figura 13 – Importâncias das *Features*

A credibilidade dos atributos selecionados pode ser verificada na Figura 14, onde é demonstrada a quantidade de predições certas e erradas. Observa-se que na modali-

dade profissional todas as predições, considerando apenas os atributos selecionados, são realizadas corretamente, obtendo-se uma acurácia de 100%.

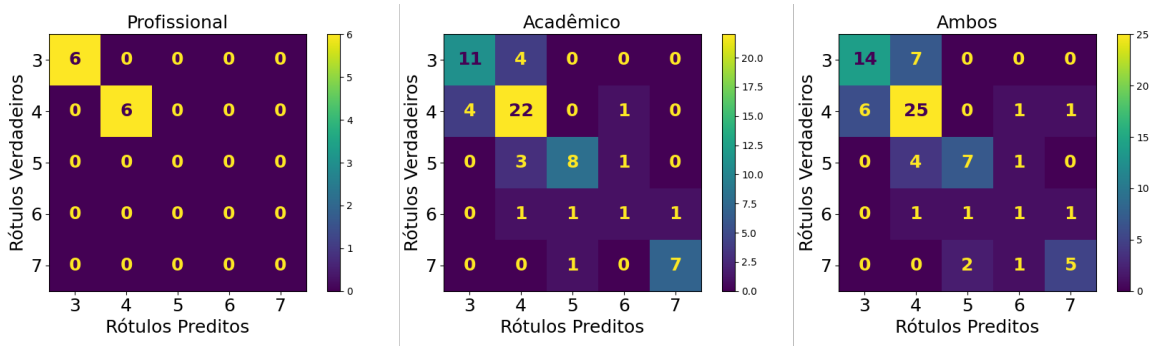


Figura 14 – Matriz de Confusão

Para a modalidade acadêmico, das 66 amostras, 49 foram classificadas corretamente, e 17 foram classificadas incorretamente, obtendo uma acurácia de 74%. Considerando ambas modalidades, das 78 amostras, ocorreu a classificação correta de 52, e 26 incorretas, resultando em uma acurácia de 66%.

## 5.3 Discussão

Ao analisar as *features* selecionadas, pode-se observar que as Tabelas 7 e 8, referentes às modalidades acadêmicos e ambos, possuem uma semelhança. Pois, de dez atributos, sete se repetem nas duas tabelas. Já a Tabela 6, referente à modalidade profissional, apresenta dissimilaridade com as demais, sendo que apenas dois atributos selecionados nesta modalidade também foram selecionados nas demais.

Observa-se também, que "Média anual ponderada (2D + 1M) de discentes titulados por DP"e "% DP com orientações de mestrado concluídas"são atributos selecionados em todas as modalidades. Demonstrando que eles se destacam entre os demais.

Outro ponto de destaque é a taxa de acertos na modalidade profissional, a qual o PPGES da Unipampa faz parte. Em que foi obtida uma acurácia de 100% na etapa de testes de credibilidade das *features* selecionadas.

# 6 CONSIDERAÇÕES FINAIS

Os programas de pós-graduação são avaliados periodicamente considerando diversas métricas quantitativas de desempenho. Embora a nota final seja pública e a nota das métricas possam ser estimadas, não há uma regra clara sobre a importância de cada métrica para o resultado (conceito) final. Para focar suas diretrizes e buscar um melhor conceito CAPES, os coordenadores dos programas precisam analisar dados, e buscar maneiras de melhorar os índices. Esses dados podem ser obtidos através da mineração de dados dos currículos Lattes ou baixados de plataformas avançadas. Mas apenas a obtenção dos indicadores não é suficiente para uma análise mais profunda, é necessário o uso de alguma técnica de inteligência que identifique quais métricas mais influenciam na obtenção de um conceito CAPES melhor. Sendo assim, este trabalho propôs o desenvolvimento de uma ferramenta, denominada PPGEva, que analisa os dados bibliométricos dos programas de pós-graduação, mediante técnicas de inteligência artificial, e classifique-os em ordem de influência na obtenção de conceitos CAPES maiores. Para atender ao que foi proposto, foi desenvolvida uma ferramenta capaz de fazer o *download* das planilhas contendo os indicadores de maneira automatizada. Também cumpre a tarefa de organizar os dados, unindo as planilhas e tratando os dados inconsistentes. Além disso, a aplicação realiza o processamento dessas informações, por meio de algoritmos de classificação e técnicas de seleção de atributos.

O PPGEva fornece ao usuário uma lista com os atributos selecionados e demonstra a acurácia dos testes. Para avaliar o *software*, foi realizado um estudo de caso no Programa de Pós-Graduação em Engenharia de Software da Unipampa, avaliado pela Capes na área de Ciência da Computação. Os testes demonstraram que a ferramenta está funcional, sendo capaz de selecionar atributos, e estes atributos selecionados permitem predizer a nota do programa. Além disso, este trabalho foi apresentado no 15º SIEPE (Salão internacional de Ensino, Pesquisa e Extensão).

Como trabalhos futuros pretende-se avaliar a ferramenta com outros PPGs do Campus Alegrete e da Unipampa, de modo a considerar outras áreas de avaliação. A aplicação também pode receber algumas melhorias, como o desenvolvimento de uma interface gráfica, e a inclusão de novos algoritmos e métricas de avaliação. Outro objetivo é disponibilizar o programa como serviço, para que ele possa ser utilizado pelos demais usuários de forma remota.

# REFERÊNCIAS

ARCHANA, S.; ELANGOVAN, K. Survey of classification techniques in data mining. **International Journal of Computer Science and Mobile Applications**, Bharathidasan University, v. 2, n. 2, p. 65–71, 2014. ISSN 2321-8363. Citado na página 17.

BENIWAL, S.; ARORA, J. Classification and feature selection techniques in data mining. **International Journal of Engineering Research and Technology**, v. 1, 08 2012. Citado na página 16.

CALISTO, A.; NóBREGA, A. Um estudo sobre os impactos dos relacionamentos sociais na avaliação do mérito científico. In: **Anais Estendidos do XIX Simpósio Brasileiro de Sistemas Multimídia e Web**. Porto Alegre, RS, Brasil: SBC, 2013. p. 17–20. ISSN 2596-1683. Disponível em: <https://sol.sbc.org.br/index.php/webmedia_estendido/article/view/4943>. Citado na página 20.

CAPARELLI, L.; DIGIAMPIETRI, L. A. Combinando análise bibliométrica e análise de redes sociais para a avaliação de grupos acadêmicos. In: **Anais do VII Brazilian Workshop on Social Network Analysis and Mining**. Porto Alegre, RS, Brasil: SBC, 2018. ISSN 2595-6094. Disponível em: <https://sol.sbc.org.br/index.php/brasnam/article/view/3580>. Citado 3 vezes nas páginas 13, 19 e 20.

DIGIAMPIETRI, L. A. et al. Brax-ray: An x-ray of the brazilian computer science graduate programs. **PLoS ONE**, Public Library of Science, v. 9, n. 4, p. e94541, 2014. Disponível em: <https://doi.org/10.1371/journal.pone.0094541>. Citado 3 vezes nas páginas 13, 20 e 21.

FRANK, E.; HALL, M. A.; WITTEN, I. H. **The WEKA Workbench**. Fourth. Morgan Kaufmann, 2016. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques". ISBN 9780128042915. Disponível em: <https://www.cs.waikato.ac.nz/ml/weka/book.html>. Citado na página 16.

GIL, A. C. **Como elaborar projetos de pesquisa**. 7. ed. Barueri [SP]: Atlas, 2022. ISBN 978-65-597-7164-6. Citado na página 15.

HAN, J.; KAMBER, M.; PEI, J. **Data mining: concepts and techniques**. 3. ed. San Francisco, CA: Morgan Kaufmann, 2011. ISBN 978-0-12-381479-1. Citado 3 vezes nas páginas 16, 17 e 20.

HISHAMUDDIN, M. N. F. et al. Improving classification accuracy of scikit-learn classifiers with discrete fuzzy interval values. In: **2020 International Conference on Computational Intelligence (ICCI)**. [S.l.: s.n.], 2020. p. 163–166. Citado na página 30.

KAMESHWARAN, K.; MALARVIZHI, K. Survey on clustering techniques in data mining. **International Journal of Computer Science and Information Technologies (IJCSIT)**, Coimbatore Institute of Technology, v. 5, n. 2, p. 2272–2276, 2014. Citado na página 17.

KESAVARAJ, G.; SUKUMARAN, S. A study on classification techniques in data mining. In: **2013 Fourth International Conference on Computing, Communications**

**and Networking Technologies (ICCCNT)**. [S.l.: s.n.], 2013. p. 1–7. Citado 2 vezes nas páginas 16 e 30.

LOPES, L.; PINTO, F.; BRITO, R. de. Sistema inteligente de apoio a decisão no processo de triagem de pacientes com suspeita do covid-19. In: **Anais da VIII Escola Regional de Computação do Ceará, Maranhão e Piauí**. Porto Alegre, RS, Brasil: SBC, 2020. p. 220–227. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/ercemapi/article/view/11488>. Citado na página 19.

NIKAM, S. S. A comparative study of classification techniques in data mining algorithms. **Orient. J. Comp. Sci. and Technol**, v. 8, n. 1, 2015. Disponível em: <http://www.computerscijournal.org/?p=1592>. Citado na página 16.

PANDAS. **pandas Documentation**. 2023. Date: Nov 10, 2023 Version: 2.1.3. Disponível em: <https://pandas.pydata.org/docs/>. Citado na página 27.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado 3 vezes nas páginas 29, 30 e 31.

PHYU, T. N. Survey of classification techniques in data mining. In: . [S.l.: s.n.], 2009. Citado na página 17.

SANTOS, T.; COSTA, O. Sistema de tomada de decisão no mercado de ações utilizando aprendizado de máquina. In: **Anais do II Brazilian Workshop on Artificial Intelligence in Finance**. Porto Alegre, RS, Brasil: SBC, 2023. p. 25–36. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/bwaif/article/view/24950>. Citado na página 19.

SHARDA, R.; DELEN, D.; TURBAN, E. **Business Intelligence e Análise de Dados para Gestão do Negócio**. 4. ed. Porto Alegre: Bookman, 2019. ISBN 978-85-8260-520-2. Citado 2 vezes nas páginas 13 e 19.

SILVA, E.; CRUZ, J. Mineração de dados educacionais: uso de redes neurais artificiais na predição do perfil acadêmico do aluno - ifal campus maragogi. In: **Anais da XIX Escola Regional de Computação Bahia, Alagoas e Sergipe**. Porto Alegre, RS, Brasil: SBC, 2019. p. 556–564. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/erbase/article/view/9018>. Citado na página 20.

SILVA, G. et al. Aplicação de business intelligence no processo de autoavaliação de instituições de ensino superior. In: **Anais Estendidos do XVIII Simpósio Brasileiro de Sistemas de Informação**. Porto Alegre, RS, Brasil: SBC, 2022. p. 1–4. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/sbsi_estendido/article/view/21559>. Citado na página 20.

SILVA, L. A. d.; PERES, S. M.; BOSCARIOLI, C. **Introdução à mineração de dados: com aplicações em R**. 1. ed. Rio de Janeiro: Elsevier, 2016. ISBN 978-85-352-8446-1. Citado 3 vezes nas páginas 15, 16 e 17.

ULMANN, G. et al. Sistema de sugestão de produtos para e-commerce utilizando inteligência artificial. In: **Anais da XXI Escola Regional de Alto Desempenho da Região Sul**. Porto Alegre, RS, Brasil: SBC, 2021. p. 53–56. ISSN 2595-4164. Disponível

em: <https://sol.sbc.org.br/index.php/eradrs/article/view/14773>. Citado na página 19.