Programa de Pós Graduação em Instrumentação, Controle e Automação de Processos de Mineração - PROFICAM
Universidade Federal de Ouro Preto - Escola de Minas
Associação Instituto Tecnológico Vale - ITV

Dissertação

ALGORITMO ACO RANK BASED APRIMORADO PARA USO NA SELEÇÃO DE VARIÁVEIS EM MODELOS DE CLASSIFICAÇÃO

Roberto Alexandre Delamora

Ouro Preto

Março de 2023

Roberto Alexandre Delamora

ALGORITMO ACO RANK BASED APRIMORADO PARA USO NA SELEÇÃO DE VARIÁVEIS EM MODELOS DE CLASSIFICAÇÃO

Dissertação apresentada ao curso de Mestrado Profissional em Instrumentação, Controle e Automação de Processos de Mineração da Universidade Federal de Ouro Preto e do Instituto Tecnológico Vale, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Controle e Automação.

Linha de Pesquisa: Tecnologias da Informação, Comunicação e Automação Industrial

Orientador: Prof. D.Sc. Bruno Nazário Coelho
Coorientador: Prof. D.Sc. Jodelson Aguilar Sabino

Ouro Preto, MG – Brasil

Março de 2023

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

D336a Delamora, Roberto Alexandre.
Algoritmo aco rank based aprimorado para uso na seleção de variáveis em modelos de classificação. [manuscrito] / Roberto Alexandre Delamora. - 2023.
62 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Bruno Nazário Coelho.
Coorientador: Dr. Jodelson Aguilar Sabino.
Dissertação (Mestrado Profissional). Universidade Federal de Ouro Preto. Programa de Mestrado Profissional em Instrumentação, Controle e Automação de Processos de Mineração. Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração.

1. Algoritmos computacionais - Colônia de formigas. 2. Algoritmos - Meta-heurística. 3. Aprendizado do computador. 4. Mineração de dados (Computação) - Seleção de atributos. I. Coelho, Bruno Nazário. II. Sabino, Jodelson Aguilar. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 622.7:681.5

Bibliotecário(a) Responsável: Maristela Sanches Lima Mesquita - CRB-1716

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
PROGR. POS GRAD. PROF. INST. CONT. E AUT.
PROCESSOS DE MIN.

FOLHA DE APROVAÇÃO

Roberto Alexandre Delamora

Algoritmo ACO rank based aprimorado para uso na seleção de variáveis para modelos de classificação

Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração (PROFICAM), Convênio Universidade Federal de Ouro Preto/Associação Instituto Tecnológico Vale - UFOP/ITV, como requisito parcial para obtenção do título de Mestre em Engenharia de Controle e Automação na área de concentração em Instrumentação, Controle e Automação de Processos de Mineração.

Aprovada em 02 de março de 2023

Membros da banca

Doutor - Bruno Nazário Coelho - Orientador (Universidade Federal de Ouro Preto)
Doutor - Jodelson Aguilar Sabino - Coorientador (Vale)
Doutor - Marcone Jamilson Freitas Souza (Universidade Federal de Ouro Preto)
Doutor - Matheus Nohra Haddad - Universidade Federal de Viçosa

Bruno Nazário Coelho, orientador do trabalho, aprovou a versão final e autorizou seu depósito no Repositório Institucional da UFOP em 20/03/2023

Documento assinado eletronicamente por **Bruno Nazário Coelho**, **COORDENADOR(A) DE CURSO DE PÓS-GRADUAÇÃO EM INST. CONTROLE AUTOMAÇÃO DE PROCESSOS DE MINERAÇÃO**, em 20/03/2023, às 14:30, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0493766** e o código CRC **5B4F336A**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.003427/2023-51 SEI nº 0493766

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3552-7352 - www.ufop.br

"O preço da luz é menor que o custo da escuridão."

Arthur Nielsen

Agradecimentos

Ninguém chega ao fim de uma jornada como esta sozinho. Definitivamente, o meu caso não foi uma exceção. Para realizar este sonho, tive o apoio de inúmeras pessoas, cada uma com sua parcela individual de contribuição que me permitiu chegar ao final.

Agradeço primeiramente a Deus pela vida e por todas as oportunidades e desafios que me foram apresentados. Tenho gratidão especial pelas dificuldades e percalços que se apresentaram durante a fase de aulas, pois elas contribuíram sobremaneira para que eu aprendesse a ter uma percepção clara sobre o que realmente é importante na vida e me ajudam todos os dias a me tornar alguém melhor.

Não tenho palavras para expressar todo meu amor e agradecimento à minha esposa Andréa, luz da nossa família, e às minhas filhas Natália e Mariana, pois sem elas a vida não teria sentido. Não foram poucos os momentos em que fez-se necessário, da parte delas, muita paciência, compreensão e amor para garantir os meios e condições para que eu seguisse estudando e desenvolvendo a pesquisa.

Agradeço também ao Instituto Tecnológico Vale (ITV), todo o seu corpo docente e discente, direção e administração que não medem esforços para nos proporcionar um ensino de alta qualidade, com seriedade e comprometimento. Em tempo, estendo os agradecimentos ao time administrativo do ITV, especialmente a Franciele Barbosa que faz toda essa máquina girar.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001; do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq); da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG); e da Vale SA.

Resumo da Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ALGORITMO ACO RANK BASED APRIMORADO PARA USO NA SELEÇÃO DE VARIÁVEIS EM MODELOS DE CLASSIFICAÇÃO

Roberto Alexandre Delamora

Março/2023

Orientadores: Bruno Nazário Coelho
Jodelson Aguilar Sabino

Seleção de atributos é um processo onde se busca o melhor subconjunto de variáveis em um determinado conjunto de dados. Em um mundo em que as decisões são cada vez mais baseadas em dados, torna-se essencial o uso de ferramentas que realizem, de forma mais eficiente, essa seleção de variáveis, visando melhorar o desempenho final dos modelos. Neste trabalho, é utilizado como referência o algoritmo *Rank-based Ant System* pertencente à meta-heurística de otimização por colônia de formigas (ACO), originalmente criado para tratar o problema do Caixeiro Viajante (TSP), e são introduzidas melhorias para adequá-lo à tarefa de seleção de variáveis. O novo algoritmo proposto utiliza métodos *Filter-Wrapper* em sua estrutura e uma função de aptidão criada especificamente para refinar a seleção de soluções. Esta abordagem foi avaliada em conjuntos de dados do repositório de aprendizado de máquina UCI e os resultados foram comparados com outro algoritmo recentemente publicado que é considerado referência na seleção de variáveis usando ACO. O algoritmo proposto apresentou ganhos importantes no desempenho, superando o algoritmo de comparação na maioria dos casos estudados.

Palavras-chave: colônia de formigas, meta-heurística, aprendizado de máquina, seleção de atributos, redução de dimensionalidade

Macrotema: Mina; **Linha de Pesquisa:** Tecnologias da Informação, Comunicação e Automação Industrial; **Tema:** Redução de Variabilidade e Melhoria de Controle.

Abstract of Dissertation presented to the Graduate Programa on Instrumentation, Control and Automation of Mining Process as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

IMPROVED ACO RANK-BASED ALGORITHM FOR USE IN SELECTING FEATURES IN CLASSIFICATION MODELS

Roberto Alexandre Delamora

March/2023

Advisors: Bruno Nazário Coelho
Jodelson Aguilar Sabino

Feature selection is a process where the best subset of variables in a given data set is sought. In a world where decisions are increasingly based on data, it is essential to use tools that more efficiently carry out this selection of variables, in order to improve the final performance of the models. In this work, the Rank-based Ant System algorithm belonging to the ant colony optimization meta-heuristic (ACO) is used as a reference, originally created to deal with the Traveling Salesman Problem (TSP), and improvements are introduced to adapt it. to the variable selection task. The proposed new algorithm uses Filter-Wrapper methods in its structure and a fitness function created specifically to refine the selection of solutions. This approach was evaluated on datasets from the UCI machine learning repository and the results were compared with another recently published algorithm that is considered a benchmark in variable selection using ACO. The proposed algorithm presented important gains in performance, surpassing the comparison algorithm in most of the studied cases.

Keywords: ant colony, metaheuristic, machine learning, feature selection, dimensionality reduction

Macrotheme: Mine; **Research Line:** Information Technologies, Communication and Industrial Automation; **Theme:** Reduction of Variability and Control Improvement.

Lista de Figuras

2.1 Representação de múltiplos caminhos que podem ser utilizados para alcançar o mesmo subconjunto de variáveis 17
2.2 Fluxograma WFACOFS 19
2.3 Resultados do coeficiente ϕ_K para amostras de referência. 28

3.1 Fluxograma da macro etapa de montagem da matriz de "distâncias". . . . 30
3.2 Fluxograma ACOFS$_{rank}$ 32
3.3 Representação da análise de *F-score* para cada variável através da ANOVA 35
3.4 Representação através de grafo de um problema de seleção de variáveis . . 36
3.5 Exemplo de situação em que a função de aptidão é utilizada. 42
3.6 Etapas iniciais da execução do algoritmo 44
3.7 Etapas de seleção de variáveis 45
3.8 Etapas de armazenamento e seleção final das melhores soluções. 45

4.1 Opções para valores de hiperparâmetros utilizadas pelo IRACE. 48
4.2 Tela de execução inicial do IRACE com as configurações da ferramenta. . . 49
4.3 Resultado do IRACE e conjunto de hiperparâmetros escolhidos. 49
4.4 Gráficos de acurácia por classificador. 51
4.5 Gráficos do percentual de variáveis selecionadas em relação ao total de variáveis de cada base de dados. 53

Lista de Tabelas

2.1 Evolução dos algoritmos AS com foco no TSP 10
2.2 Exemplo de dados originais de categorias 25
2.3 Exemplo de tabela de contingência 25

4.1 Descrição das bases de dados utilizadas 47
4.2 Descrição dos valores dos parâmetros passados ao algoritmo 48
4.3 Valores finais de acurácia por classificador. 50
4.4 Percentuais de variáveis selecionadas em relação ao total de variáveis de cada base de dados. 52
4.5 Média e Desvio Padrão dos melhores valores de acurácia obtidos em cada rodada de testes. 52
4.6 Comparação entre os melhores resultados de cada algoritmo. 54

Lista de Abreviaturas e Siglas

ACO *Ant Colony Optimization* (Otimização com Colônia de Formigas)

ACOFS$_{rank}$ *ACO Feature Selection Rank-Based System*

ANOVA *Analysis of Variance*

AS *Ant System*

AS$_{rank}$ *Ant System with elitist strategy and ranking*

FS *Feature Selection* (Seleção de variáveis)

ITV Instituto Tecnológico Vale

KNN *K-Nearest Neighbors*

ML *Machine Learning* (Aprendizado de Máquina)

MLP *Multi Layer Perceptron*

SA *Simulated Annealing*

TS *Tabu Search* (Busca Tabu)

TSP *Travelling Salesman Problem* (Problema do Caixeiro Viajante)

UCI University of California Irvine Machine Learning Repository

WFACOFS *Wrapper-Filter ACO Feature Selection*

Lista de Símbolos

α alfa	(Influência do feromônio)
β beta	(Influência da correlação entre variáveis)
η eta	(Visibilidade ou distância entre variáveis)
γ gamma	(Acurácia calculada para subconjunto de variáveis)
Ω Omega	(Variáveis ainda não visitadas pelas formigas)
ω omega	(Número de formigas elitistas)
ϕ phi	(Fator de balanceamento da acurácia)
ρ rho	(Fator de evaporação do feromônio)
τ tau	(Intensidade do feromônio)

Sumário

1 Introdução **1**

1.1 Contextualização 1

1.2 Motivação e Justificativa 3

1.3 Objetivos Gerais e Específicos 5

1.4 Delimitações do estudo 5

2 Revisão Bibliográfica **7**

2.1 Algoritmo de Colônia de Formigas 8

2.1.1 *Ant System* (AS) 12

2.1.2 *Rank Based System* (AS_{rank}) 14

2.1.3 WFACOFS - *Wrapper-Filter ACO Feature Selection* 16

2.2 Métodos *Filter* e *Wrapper* 19

2.3 Cálculo de distâncias - Correlação 20

2.3.1 Correlação entre variáveis numéricas 22

2.3.2 Correlação entre variáveis categóricas 24

2.3.3 Correlação entre variáveis numéricas e categóricas 26

2.3.4 Método phik - ϕ_K 27

3 Materiais e Métodos **29**

3.1 Preparação dos dados para uso do modelo 29

3.2 Ajustes no algoritmo *Rank Based AS* (AS_{rank}) 31

3.2.1 Método *Filter* 33

3.2.2 Etapa de construção 36

3.2.3 Método *Wrapper* 40

3.2.4 Avaliação de aptidão ou *fitness* 41

3.2.5 Descrição do funcionamento geral do algoritmo 43

4 Experimentos e Resultados **46**

5 Considerações Finais **55**

5.1 Conclusões 55

5.2 Trabalhos Futuros 56

Referências Bibliográficas **57**

Capítulo 1

Introdução

Este capítulo traz a contextualização do atual cenário de disponibilidade de dados e sua importância para as organizações e para a economia global. Em sequência, estabelece a motivação para a realização da pesquisa e os objetivos gerais e específicos que se busca alcançar sem deixar de lado as definições sobre limites da abrangência da pesquisa.

1.1. Contextualização

A forma mais simples de representação dos dados é realizada por meio de atributos. O tamanho do conjunto de dados pode ser medido em duas dimensões: número de atributos e número de registros, sendo que seu volume pode interferir na aplicação direta de métodos para construção de modelos que representam o conhecimento embutido nesse conjunto de dados. Torna-se necessário que eles sejam representados de maneira apropriada, processados e o modelo construído, avaliado e validado (GOLDSCHMIDT *et al.*, 2015).

Há poucos anos atrás, para realizar a coleta de volumes significativos de dados era preciso fazer análises por longos períodos de tempo. Atualmente, dados são gerados por sensores em milhões de dispositivos, máquinas, veículos e até mesmo postes de rua. No passado, manter essa quantidade de dados era caro e difícil, mas as capacidades de armazenamento avançaram e os custos diminuíram muito, tornando os dados armazenados um recurso renovável.

A oportunidade que a inovação em dados oferece ao mundo é praticamente inédita. Ferramentas inovadoras de aprendizado de máquina já revolucionam questões das vidas dos indivíduos de maneiras incríveis. Agora, essas ferramentas estão ajudando as pessoas a revelarem as respostas ocultas com a crescente abundância de recursos de dados. Essas novas tecnologias transformadoras estão convertendo dados em novos produtos, novas

soluções e inovações que prometem mudar significativamente as vidas das pessoas e seu relacionamento com o mundo.

Esse crescimento na disponibilidade de dados ajuda a ter mais poder em mãos, colocando informações muito necessárias ao alcance dos dedos. Conforme os dados deixam de ser um recurso escasso passando a ser cada vez mais abundantes, valiosos e renováveis, eles se tornam uma fonte essencial de benefícios econômicos e sociais.

Historicamente o acesso a recursos como terra, mão de obra e capital foi o diferencial econômico entre quem teve êxito e quem fracassou. Hoje, boa parte dos líderes empresariais citam dados como um dos recursos essenciais e como um diferencial fundamental para os negócios, talvez no mesmo nível como ocorria com os recursos básicos no passado. Do ponto de vista econômico e em uma estimativa conservadora, economistas avaliam que se um aproveitamento mais eficaz de dados gerasse pequenos ganhos, tornando os setores de atividade apenas 1% mais eficientes, isso agregaria quase US$ 15 trilhões ao PIB global até 2030 (BSA THE SOFTWARE ALLIANCE, 2015).

A grande questão neste contexto não é somente ter mais dados, já que isso acontecerá naturalmente, mas saber quais dados devam ser utilizados para se alcançar o objetivo esperado da melhor forma e minimizando gastos de tempo e recursos, materiais e financeiros.

De acordo com Sharda *et al.* (2019), embora sua proposta de valor seja inegável, para fazer jus à sua promessa, os dados precisam obedecer a alguns parâmetros básicos de usabilidade e qualidade. Nem todos os dados são úteis para todas as tarefas, em outras palavras, os dados precisam corresponder às tarefas para as quais visam ser usados.

Parte desse trabalho já é feito hoje com grande eficiência pelos modelos de aprendizado de máquina (*machine learning*), que são nada mais nada menos que programas treinados para reconhecer determinados tipos de padrões. Em resumo, o programa lê um conjunto de dados, reconhece relacionamentos e padrões e gera um modelo matemático que pode ser usado para ponderar e aprender com esses dados e com novos dados ainda não analisados, podendo até tomar decisões com o mínimo de intervenção humana.

O crescente desenvolvimento de modelos de aprendizado de máquina tem permitido analisar dados em volumes cada vez maiores e mais complexos e entregar resultados mais rápidos e precisos. Ao construir modelos eficientes, uma organização tem mais chances de identificar novas, boas e lucrativas oportunidades, além de evitar riscos ainda não vislumbrados.

Os algoritmos de aprendizado de máquina (*machine learning*) são bastante eficientes em descobrir padrões em grandes volumes de dados mas, contraditoriamente, também

são bastante afetados por vieses[1] e relacionamentos contidos nesses dados.

Incluir atributos redundantes pode ser enganoso para algoritmos de modelagem. Métodos baseados em instâncias, como o KNN (k vizinhos mais próximos) usam pequenas vizinhanças no espaço de atributos para determinar as previsões de classificação e regressão e essas previsões podem ser bastante distorcidas por atributos redundantes.

Além disso, manter atributos irrelevantes em seu conjunto de dados pode resultar em sobreajuste (*overfitting*[2]) que gera perda da capacidade de generalização e de desempenho.

Seleção de atributos ou variáveis (*Feature Selection - FS*) é um processo pelo qual se busca, de forma automatizada, identificar e selecionar um subconjunto de variáveis em um determinado conjunto de dados, que se mostre mais relevante no comportamento da variável alvo de acordo com Brownlee (2019a).

Segundo Kuhn e Johnson (2013), o processo de seleção de atributos foca primariamente em remover preditores redundantes ou não-informativos do modelo.

Uma maneira útil de pensar sobre o problema de seleção de variáveis é uma pesquisa no espaço-solução. O espaço de pesquisa é discreto e consiste em todas as combinações possíveis de variáveis prováveis de escolha no conjunto de dados. O objetivo é navegar pelo espaço-solução e localizar a melhor combinação ou boa o suficiente que melhore o desempenho em relação ao uso de todas as variáveis (BROWNLEE, 2014).

Assim, num mundo em que as decisões são cada vez mais baseadas em dados, torna-se essencial o desenvolvimento de ferramentas que permitam realizar de forma mais eficiente essa seleção de variáveis de um conjunto de dados, visando melhorar o desempenho final dos modelos e retirando do contexto da análise aquelas que não são significativas ou que causam impactos negativos no resultado.

A literatura descreve variadas técnicas e algoritmos para realizar a tarefa de seleção de variáveis, cada um deles com características e limitações próprias. Entretanto isso não impede a pesquisa por novas técnicas, na busca constante por melhorias.

1.2. Motivação e Justificativa

Na visão geral do ciclo de vida de um projeto de mineração de dados, a etapa de preparação dos dados é formada por um conjunto de atividades que envolvem organização,

[1] Viés representa uma tendência ou propensão desvirtuada ou preconceituosa. No âmbito dos dados, viés consiste em um erro sistemático (MICHAELIS DICIONÁRIO ONLINE).

[2] Num cenário de *overfitting*, o modelo "decora" as relações existentes nos dados de treino. Ao receber novos dados, o modelo tenta aplicar as regras decoradas, mas elas não têm validade.

estruturação e transformação dos dados. Essas atividades são precedentes fundamentais para a realização de análises e predições.

Frequentemente essa é uma das etapas que mais consomem tempo. De fato, estima-se que esta etapa consuma entre 50-70% do tempo e esforço do projeto e, ao mesmo tempo, trata-se de esforço crucial para o sucesso da etapa posterior de modelagem (CHAPMAN *et al.*, 1999)

O presente trabalho se concentra no desenvolvimento de estudos para avaliar a possibilidade de melhorias no modelo computacional de seleção de variáveis baseado na classe de algoritmos de Otimização de Colônia de Formigas (ACO) (STÜTZLE e DORIGO, 1999), mais especificamente na meta-heurística AS_{rank} (*Ant System with elitist strategy and ranking*) (BULLNHEIMER *et al.*, 1997). Essa meta-heurística foi desenvolvida em 1997 como evolução do modelo original AS (*Ant System*) (DORIGO, 1991) e ainda hoje é muito utilizada como base para novos modelos, adaptados a necessidades e aplicações específicas.

Utilizando como referência literária o algoritmo WFACOFS (*Wrapper-Filter ACO Feature Selection*) (GHOSH *et al.*, 2019) de seleção de variáveis não-supervisionado do tipo *Filter-Wrapper*, e considerando como função de avaliação a medida de acurácia obtida através da classificação de subconjuntos de atributos selecionados, este trabalho propõe ajustes no AS_{rank} buscando melhorias de desempenho.

O ACO foi inspirado na observação do comportamento das formigas ao saírem de sua colônia para encontrar comida, é baseado em probabilidade e foi criado para solução de problemas computacionais que envolvem procura de caminhos em grafos.

Embora o ACO (STÜTZLE e DORIGO, 1999) e o AS_{rank} (BULLNHEIMER *et al.*, 1997) fossem originalmente desenvolvidos para resolver o Problema do Caixeiro Viajante (*Travelling Salesman Problem - TSP*), eles podem ser personalizados para tratar o problema de seleção de variáveis.

Por uma questão de simplicidade, daqui em diante, será utilizada a denominação ACOFS_{rank} (*ACO Feature Selection Rank-Based System*) para o método proposto.

Com o objetivo de proporcionar o máximo de comparabilidade entre o cenário apresentado pelo WFACOFS e aquele apresentado pelo ACOFS_{rank}, foram consideradas as mesmas bases de dados e os mesmos parâmetros possíveis nos estudos envolvendo os dois modelos.

Atualmente, o método WFACOFS proposto por Ghosh *et al.* (2019) é o que atinge os melhores resultados nas bases estudadas. Desenvolver estudos para buscar uma evolução a partir do já amplamente conhecido algoritmo AS_{rank} (BULLNHEIMER *et al.*,

1997), avaliando ajustes para executar a seleção de variáveis e para utilizar estatística de correlação como referência de distâncias de forma a avaliar a possibilidade de se obter um algoritmo melhorado que possa se equiparar ao WFACOFS ou até mesmo superá-lo em parte ou totalmente, seria uma grande contribuição ao tema.

Desta forma, o presente trabalho procura contribuir na pesquisa por novos métodos de solução para o problema da seleção de variáveis ao propor uma nova possibilidade para uma abordagem construída a partir de uma meta-heurística originalmente criada para busca de melhores rotas, e que apresenta características bastante desejáveis ao problema proposto.

1.3. Objetivos Gerais e Específicos

Este trabalho tem como objetivo geral realizar estudos para buscar um aprimoramento do algoritmo *Rank-based Ant System* (AS_{rank}), originalmente criado para solução do problema do Caixeiro Viajante (TSP) e avaliar seu desempenho para seleção de variáveis, comparando com outro algoritmo recentemente criado e de performance reconhecida dentre modelos que utilizam ACO.

Os objetivos específicos são os seguintes:

- Revisar a bibliografia do ACO ressaltando sua evolução e os principais modelos apresentados ao longo do tempo;
- Revisar literatura atualizada sobre métodos estatísticos para definição de correlação entre variáveis de diferentes tipos (numéricas, categóricas e categóricas-numéricas);
- Criar uma documentação dos métodos estudados e dos testes realizados, bem como repositório para o código desenvolvido, disponibilizando esse código e artigos de referência.

1.4. Delimitações do estudo

Constituem-se em restrições do presente trabalho:

- As variáveis são selecionadas somente através de métodos supervisionados, sendo necessária prévia informação sobre a classe de cada amostra, ou variável dependente a ser modelada;
- A influência de métodos de pré-tratamento no desempenho dos algoritmos de classificação não é testada.
- Análises estatísticas para avaliação e adequação dos dados como, por exemplo, a

validação de normalidade, não são descritas e nem mesmo aplicadas por se entender que fazem parte do processo de pré-tratamento dos dados.

Capítulo 2

Revisão Bibliográfica

Durante as discussões iniciais para desenvolvimento desta pesquisa, surgiram vários pontos que teriam que ser tratados ou construídos antes do início da fase de codificação e realização de experimentos.

Na solução do TSP faz-se necessária a criação de uma matriz de tamanho n x n, sendo n o número de cidades a serem percorridas, onde cada célula contém a distância entre as cidades correspondentes. De modo geral, na literatura são utilizadas distâncias cartesianas (no caso do TSP) ou distâncias de cossenos (nos algortimos de seleção de variáveis). Neste trabalho foi estabelecido que as "distâncias" da matriz seriam valores de correlação medidos entre as variáveis da base de dados sob análise.

A utilização de valores de correlação como métrica de distância possui a vantagem de ter o mesmo significado matemático da distância real entre cidades, utilizada no TSP. Quanto mais relacionadas estão as variáveis, maior a correlação e, consequentemente, maior a "distância" entre elas. Essa analogia traz o efeito positivo de que variáveis com informação semelhante ou muito próxima não são interessantes aos modelos de aprendizado de máquina. Neste caso, com uma correlação (ou "distância") elevada, esse par de variáveis se torna menos interessante.

O desafio apresentado para a criação da matriz de distâncias é que as bases do mundo real, de modo geral possuem variáveis de tipos diversos: numéricos (discretos e contínuos) e categóricos (ordinal e nominal). A forma utilizada para tentar equacionar essa questão e apresentar um valor de correlação dentro de uma métrica universal foi a pesquisa por diferentes métodos estatísticos que são descritos nos tópicos destinados ao aprofundamento teórico descritos neste capítulo.

Trata-se de fundamentação teórica importante no contexto da proposta de se utilizar correlação como métrica de distâncias, fugindo do tradicionalismo do cálculo por

similaridade. Além disso, abre oportunidades para uso de bases reais, com suporte inclusive para variáveis categóricas.

2.1. Algoritmo de Colônia de Formigas

O algoritmo de Colônia de Formigas (ACO) é uma meta-heurística de busca populacional para otimização combinatória que foi criada para a solução de problemas computacionais que envolvem procura de caminhos em grafos, baseando-se em probabilidade.

O algoritmo simula o comportamento de um conjunto de agentes (formigas) pertencentes a uma colônia na busca por alimentos, cooperando entre si para minimizar o caminho a ser seguido entre a colônia e a fonte de alimentos, utilizando comunicação indireta.

Segundo (GASPAR-CUNHA *et al.*, 2012, capítulo 5), a interação entre os indivíduos da colônia ocorre através de um fenômeno denominado por estigmergia[1], termo introduzido pelo zoológo francês Pierre-Paul Grassé em 1959. Estigmergia refere-se à noção de que qualquer ação de um determinado agente deixa sinais no meio ambiente, e este sinal poderá ser percebido por outros agentes (normalmente da mesma espécie) de forma a determinar suas ações subsequentes.

Como as formigas são animais quase sem nenhuma visão, esta sinalização (ou comunicação) é feita através do depósito de uma substância chamada feromônio[2] que é utilizado como forma de comunicação química. Uma formiga em movimento deposita certo volume de feromônio no solo, marcando assim o caminho que segue com um rastro dessa substância. Enquanto uma formiga isolada se move ao acaso, uma formiga que encontra uma trilha previamente colocada pode detectá-la e decidir, com alta probabilidade, por segui-la, reforçando esta trilha com seu próprio feromônio. Daí emerge um comportamento coletivo em que quanto mais as formigas seguem uma determinada trilha, mais atraente essa trilha se apresenta para ser seguida.

Faz-se necessário dizer que, com o passar do tempo, o feromônio inicialmente depositado sofre evaporação e perde intensidade. Desta forma, aqueles caminhos que não recebem o reforço da substância dado por outras formigas vão naturalmente sendo "apagados".

De acordo com (GASPAR-CUNHA *et al.*, 2012, capítulo 5), computacionalmente, a meta-heurística ACO é um método de busca construtivo no qual uma população de

[1]Estigmergia: palavra proveniente do grego *stigma* (marca, sinal) e *érgon* (ação, trabalho) (GASPAR-CUNHA *et al.*, 2012)

[2]Feromônio: palavra proveniente do grego *féro* (transportar, transmitir) e *órmon*, particípio presente de *órmao* (excitar) (GASPAR-CUNHA *et al.*, 2012).

agentes (formigas artificiais) constrói cooperativamente soluções candidatas para um dado problema. A construção é probabilística, guiada pela heurística do problema e por uma memória compartilhada entre os agentes, contendo a experiência de iterações anteriores. Essa memória consiste em uma trilha artificial de feromônio.

As formigas são consideradas agentes que executam procedimentos estocásticos e constroem os subconjuntos de variáveis iterativamente, usando tanto as informações heurísticas quanto a quantidade de feromônio acumulada nas trilhas. O componente estocástico traz uma exploração mais completa do espaço das soluções e cria uma grande variedade de subconjuntos em comparação com uma heurística gulosa. A estratégia de busca de formigas é remanescente do aprendizado por reforço (DORIGO e STÜTZLE, 2019).

O processo é caracterizado por um *loop* de retorno positivo, onde a probabilidade de uma formiga escolher um caminho aumenta com o número de formigas que escolheram o mesmo caminho anteriormente (DORIGO, 1991).

O primeiro algoritmo de Colônia de Formigas, chamado *Ant System* (AS), foi desenvolvido por Dorigo (1991) e foi aplicado ao Problema do Caixeiro Viajante (TSP) com resultados bastante positivos.

Como descrito por Dorigo *et al.* (1996), esse algoritmo apresenta características bastante desejáveis:

- **É versátil**, pois pode ser aplicado a versões semelhantes do mesmo problema;
- **É robusto**, pois pode ser aplicado com apenas mudanças mínimas a outros problemas de otimização combinatória;
- É uma abordagem baseada na **população**. Tal aspecto permite a exploração de *feedback* positivo como um mecanismo de busca.

Ainda segundo Dorigo *et al.* (1996), essas propriedades desejáveis são contrabalançadas pelo fato de que, para algumas aplicações, o AS pode ser superado por algoritmos mais especializados. Este é um problema compartilhado também por outras abordagens populares como Recozimento Simulado (*Simulated Annealing - SA*) e Busca Tabu (*Tabu Search - TS*).

As formigas artificiais do AS se comportam de maneira semelhante às formigas do mundo real. Elas diferem de suas contrapartes naturais em dois aspectos: **(i)** elas não são cegas, ou seja, elas têm informações sobre seu ambiente e usam essas informações para serem gananciosas, além de serem adaptativas e; **(ii)** elas têm uma memória, que é necessária para garantir que apenas soluções viáveis sejam geradas (BULLNHEIMER *et al.*, 1997).

A partir desse primeiro modelo, várias melhorias foram propostas e novamente testadas no TSP. Todas essas versões aprimoradas do AS têm em comum uma exploração mais forte das melhores soluções encontradas para direcionar o processo de busca das formigas. Elas diferem principalmente em alguns aspectos do controle de pesquisa (STÜTZLE e DORIGO, 1999). A Tabela 2.1 contém uma lista não exaustiva de variações de sucesso do ACO para a solução do TSP e dos métodos desenvolvidos com foco na seleção de atributos (FS - *Feature Selection*).

Tabela 2.1: Evolução dos algoritmos AS com foco no TSP

Algoritmo/Autor/Ano	**Informações relevantes**
Ant System (AS) Dorigo (1991)	Primeira proposta de nova abordagem viável para otimização combinatória estocástica em analogia com o funcionamento das colônias de formigas.
Elitist AS (EAS) Dorigo (1992)	Introduziu o conceito de elitismo utilizado nos algoritmos genéticos. Os aspectos da busca local se tornam mais importantes que os da busca global.
Ant-Q Gambardella e Dorigo (1995)	Considerado uma nova família de algoritmos para a solução do TSP e que introduziu o uso de escolha de ação proporcional pseudoaleatória e reforço retardado de melhor iteração.
Ant Colony System (ACS) Gambardella e Dorigo (1996)	Modelo adaptado do AS para resolução do TSP. Apenas a melhor formiga deposita feromônio.
Max-Min AS (MMAS) Stützle e Hoos (1996)	Exploração por método guloso. Somente as melhores formigas utilizam o feromônio que possui um valor MIN e MAX.
Rank Based AS (AS_{rank}) Bullnheimer *et al.* (1997)	Introduz o conceito de *ranking* onde a quantidade de feromônio depositada é maior em função de quão melhor é a solução apresentada por uma determinada formiga.
ANTS Maniezzo e Carbonaro (2000)	Introdução do conceito de que a atratividade de um movimento pode ser efetivamente estimada por meio de limites inferiores (limites superiores no caso de problemas de maximização) para o custo da conclusão de uma solução parcial.

Fonte: Ayres (2021), Gaspar-Cunha *et al.* (2012)

Algoritmo/Autor/Ano	**Informações relevantes**
Hyper-Cube AS (HCAS) Blum *et al.* (2001)	Modelo híbrido com utilização de *framework* Hyper-Cube (análise de hiperespaço) para atualização do feromônio.
Best-Worst (BWAS) Cordón *et al.* (2002)	Utiliza conceitos de algoritmos genéticos. A melhor solução é considerada para atualização do feromônio e a pior solução é penalizada para reduzir sua atratividade.
TFSACO Aghdam *et al.* (2009)	As regras de atualização global e local alteram de acordo com o nível de feromônio dos caminhos após cada iteração.
ACOFS Kabir *et al.* (2009)	Modelo híbrido, combina as tecnologias do tipo *Filter* e *Wrapper*. Incorpora o IG do atributo no cálculo da informação heurística e avalia subconjuntos após adicionar novo atributo.
TSIACO Zhang e Feng (2012)	Atualização do feromônio é realizada em dois estágios.
UFSACO Tabakhi *et al.* (2014)	Primeiro método não supervisionado proposto. Seleciona sub conjuntos baseado na similaridade entre os atributos. O feromônio é atualizado com base na frequência dos atributos selecionados.
C-RACOFS Forsati *et al.* (2014)	A atualização do feromônio local previne que o algoritmo convirja prematuramente. Formulações diminuem os níveis de feromônio nos nós percorridos.
FS *based on the graph clustering and* ACO Forsati *et al.* (2014)	Agrupam os atributos usando o método Louvain para detecção de comunidades.
WFACOFS Ghosh *et al.* (2019)	Modelo híbrido. A regra de atualização do feromônio é balanceada após cada iteração. Introduzido o conceito de *fitness-based memory*.

Fonte: Ayres (2021), Gaspar-Cunha *et al.* (2012)

O presente trabalho foca no desenvolvimento de pesquisa teórica e experimentos práticos utilizando uma versão modificada do algoritmo *Rank Based System* (AS_{rank}), aqui denominada ACOFS_{rank}, com objetivo de realizar a seleção de variáveis em bases de dados e considera, como referência para a comparação de resultados, o algoritmo WFACOFS (GHOSH *et al.*, 2019).

Como parte do desenvolvimento, realizou-se detalhamento teórico dos algoritmos

AS, AS_{rank} e WFACOFS. Também foram abordadas considerações sobre refinamentos matemáticos e estatísticos que foram realizados previamente à aplicação do algoritmo $ACOFS_{rank}$, como a criação de uma tabela de "distâncias" com informações sobre correlação de pares de variáveis independentes. Por fim, no Capítulo de Metodologia será dedicado espaço ao detalhamento do $ACOFS_{rank}$.

2.1.1. *Ant System* (AS)

Dado um TSP de cidades com distâncias d_{ij}, formigas artificiais são distribuídas para as cidades de acordo com alguma regra. Cada formiga decide de forma independente a cidade a ser visitada a seguir, até que o trajeto seja completamente concluído. No início de uma iteração, todas as cidades, exceto aquela em que a formiga está localizada, podem ser selecionadas. Como cada formiga tem que visitar cada cidade exatamente uma vez, as cidades até então visitadas têm que ser armazenadas, por isso faz-se necessária a memória.

A probabilidade de uma cidade ser selecionada é tanto maior quanto mais intenso for o nível da trilha que leva a essa cidade e quanto mais perto essa cidade estiver localizada. Assim, a probabilidade p de a cidade j ser selecionada para ser visitada imediatamente após a cidade i pela formiga k foi descrita por Dorigo (1991) através da Equação 2.1.

O comportamento relacionado à adaptação baseia-se nos níveis de feromônio da trilha e é regulado por um parâmetro α, enquanto o segundo comportamento com características gulosas se baseia na visibilidade e é regulado por um parâmetro β.

$$p_{ij}^{k} = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}}{\sum_{h \in \Omega} ([\tau_{ih}(t)]^{\alpha} \cdot [\eta_{ih}(t)]^{\beta})} \tag{2.1}$$

$$\eta_{ij} = \frac{Q}{d_{ij}}$$

em que:

t iteração em questão, cada execução de uma rota completa

τ_{ij} intensidade do feromônio presente na rota entre as cidades i e j

α parâmetro que regula a influência do feromônio

η_{ij} visibilidade (inverso da distância) entre as cidades i e j

β parâmetro que regula a influência da distância entre as cidades i e j

Ω lista de cidades ainda não visitadas pela formiga

Q constante de atualização do feromônio

d_{ij} distância entre as cidades i e j

Este processo é repetido até que todas as formigas tenham completado uma rota ou trajeto completo. Em cada etapa da iteração t o conjunto de cidades a serem visitadas é reduzido em uma cidade e quando sobrar apenas uma cidade, ela é selecionada com probabilidade $p_{ij} = 1$.

A distância da rota escolhida por cada formiga é calculada e a melhor rota encontrada até o momento é então atualizada. Em seguida, os níveis de feromônio da trilha são atualizados da seguinte forma: no trajeto, cada formiga deixa uma quantidade de feromônios que é dada por Q/d_k, onde Q é a constante de atualização do feromônio e d_k é a distância percorrida pela formiga (DORIGO, 1991).

A atualização do nível de feromônio τ_{ij} foi também descrita por Dorigo (1991) através da Equação 2.2. Por analogia com a natureza, parte do feromônio se perde por evaporação. As trilhas de feromônio são reduzidas por um fator $(1 - \rho)$ antes que uma nova camada de feromônio seja colocada. Isso é feito para evitar a convergência precoce e é regulado por um parâmetro chamado persistência de trilha e denotado por ρ.

$$\tau_{ij}^{k}(t+1) = (1-\rho) \cdot \tau_{ij}^{k}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{2.2}$$

$$\Delta\tau_{ij}^{k} = \frac{Q}{d_k}$$

em que:

t iteração em questão, cada execução de uma rota completa

$\rho \in [0, 1]$ parâmetro para regular a perda por evaporação do feromônio

$\Delta\tau_{ij}^{k}$ feromônio depositado pela formiga k após passar pela rota (i, j)

m número de formigas

Q constante de atualização do feromônio

d_k distância total da trilha percorrida pela formiga k

A estrutura genérica de um AS é ilustrada no Algoritmo 1.

Algoritmo 1: *Ant System*

```
1:  Início
2:      Inicializar parâmetros
3:      Inicializar matriz de feromônio
4:      repetir para cada colônia
5:          repetir para cada formiga
6:              Construir solução
7:          até final
8:          Atualizar feromônio
9:      até final
10: Fim
```

Fonte: Gaspar-Cunha *et al.* (2012)

De acordo com Bullnheimer *et al.* (1997), o AS alcança melhores resultados quando o número de formigas é igual ao número de cidades do TSP e cada formiga inicia seu trajeto em uma cidade diferente.

Quanto aos demais parâmetros do algoritmo (α, β, ρ e Q), verificou-se que se α é muito alto em relação a β, o algoritmo tende a entrar em comportamento de estagnação sem encontrar boas soluções. No entanto, se ao contrário, α for muito baixo, o algoritmo opera como uma heurística de construção repetida e gera boas soluções, mas não pode explorar o retorno positivo (DORIGO *et al.*, 1996).

O mesmo é verdadeiro para valores de ρ reduzidos a zero. A maior parte das informações dos níveis de feromônio na trilha "evaporam" imediatamente e o aprendizado não ocorre. Se ρ for muito próximo à unidade, existe o perigo de convergência precoce do algoritmo.

O parâmetro Q mede a influência da distância da trilha em relação à influência dos níveis de feromônio iniciais τ_{ij}. Desde que Q não seja muito pequeno, este parâmetro não apresenta grande influência para a convergência do algoritmo.

Segundo Dorigo *et al.* (1996), o ajuste mais vantajoso para os parâmetros do algoritmo seria $\alpha = 1$, $\beta = 5$ e $\rho = 0{,}5$.

2.1.2. *Rank Based System* ($\mathbf{AS}_{rank}$)

A variante do AS baseada em classificação denominada AS_{rank} (BULLNHEIMER *et al.*, 1997) é uma das melhorias implementadas no modelo original. As diferenças são sutis, mas acrescentaram maior desempenho ao algoritmo, que também teve o tempo geral de convergência reduzido.

No AS_{rank}, sempre a melhor trilha (menor distância percorrida) global é utilizada para atualizar os níveis de feromônios, o que caracteriza uma estratégia elitista. Além disso, somente algumas das melhores formigas da iteração atual são habilitadas a adicionar feromônios. Para que isso aconteça, as formigas são classificadas pela distância da trilha ($d^1(t) \leq d^2(t) \leq \cdots \leq d^m(t)$), e a quantidade de feromônio que uma formiga pode depositar é definida de acordo com a o índice de ranqueamento r. Somente as $(\omega - 1)$ melhores formigas de cada iteração podem depositar feromônio. A melhor solução global, aquela que fornece o retorno mais forte, recebe o peso ω. A r-ésima melhor formiga da iteração contribui para a atualização do feromônio com um peso dado por $max\{0, \omega - r$ (STÜTZLE e DORIGO, 1999).

A descrição matemática do mecanismo de atualização descrita por Stützle e Dorigo (1999) é apresentada na Equação 2.3.

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{r=1}^{\omega-1} (\omega - r) \cdot \Delta\tau_{ij}^{r}(t) + \omega \cdot \Delta\tau_{ij}^{gb}(t) \tag{2.3}$$

$$\Delta\tau_{ij}^{r}(t) = \frac{1}{d^r(t)}$$

$$\Delta\tau_{ij}^{gb}(t) = \frac{1}{d^{gb}(t)}$$

em que:

r	índice de ranqueamento
$\Delta\tau_{ij}^{r}$	acréscimo de feromônio causado pela r-ésima melhor formiga
$\Delta\tau_{ij}^{gb}$	acréscimo de feromônio causado pela melhor formiga dentre todas
ω	número de formigas elitistas
d^r	distância total da trilha percorrida pela r-ésima melhor formiga
d^{gb}	distância total da trilha considerando a melhor solução final

A estrutura genérica de um AS_{rank} é ilustrada no Algoritmo 2.

Algoritmo 2: AS_{rank}

```
1:  Início
2:      Inicializar parâmetros
3:      Inicializar matriz de feromônio
4:      repetir para cada colônia
5:          repetir para cada formiga
6:              Construir solução
7:          até final
8:          Ordenar soluções
9:          Atualizar feromônio
10:     até final
11: Fim
```

Fonte: Gaspar-Cunha *et al.* (2012)

Segundo Bullnheimer *et al.* (1997), as atualizações produzidas pelo AS_{rank} apresentaram bons resultados enfatizando bons caminhos e a exploração mais ampla do espaço de soluções, mesmo sem alteração nos parâmetros inicialmente definidos para o AS (descritos na subseção anterior).

Para as instâncias maiores de TSP, as abordagens baseadas em AS mostraram-se superiores ao algoritmo genético e ao procedimento de recozimento simulado. Entre os algoritmos baseados em AS, o AS_{rank} obteve desempenho significativamente melhor do que o AS original (STÜTZLE e DORIGO, 1999).

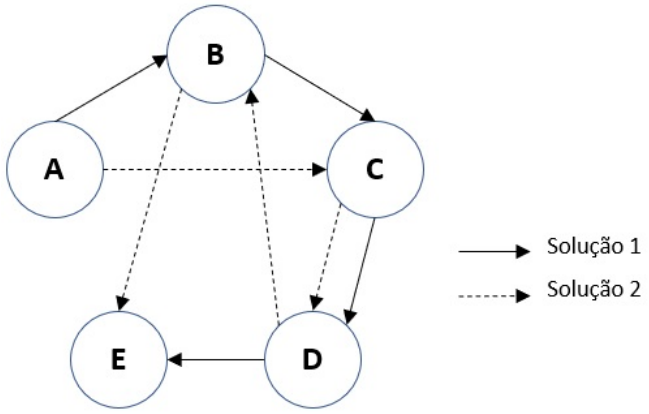
2.1.3. WFACOFS - *Wrapper-Filter ACO Feature Selection*

De acordo com Ayres (2021), o algoritmo WFACOFS (GHOSH *et al.*, 2019) do tipo híbrido foi implementado de maneira a combinar as melhores vantagens dos métodos do tipo *Filter* e *Wrapper*. Para o seu desenvolvimento foram considerados como base os algoritmos UFSACO (TABAKHI *et al.*, 2014) e TFSACO (AGHDAM *et al.*, 2009), propondo técnicas de sucesso para contornar as deficiências observadas em cada um.

Como descrito por Ghosh *et al.* (2019), no ACO as formigas se movem no grafo de um nó para outro através das arestas e vão depositando feromônio ao longo dos caminhos. Essa abordagem possui uma característica em que um mesmo subconjunto de nós pode ser obtido com as formigas percorrendo caminhos diferentes. Se, como exemplo, for considerado o grafo apresentado na Figura 2.1 com os nós (variáveis) representados pelo conjunto $\{A, B, C, D, E$, trechos percorridos que consideram esses nós podem ser construídos a partir de $\{A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ ou de $\{A \rightarrow C \rightarrow D \rightarrow B \rightarrow E$ ou ainda de $\{A \rightarrow E \rightarrow D \rightarrow B \rightarrow C$. Isso indica que o feromônio é depositado de forma desigual ao

longo do trecho. Para a solução do TSP isso não representa um problema, mas do ponto de vista de seleção de variáveis é uma limitação importante.

Figura 2.1: Representação de múltiplos caminhos que podem ser utilizados para alcançar o mesmo subconjunto de variáveis



Fonte: Adaptado de Ghosh *et al.* (2019)

Para superar esse problema, o WFACOFS trabalha com a proposta de realizar o depósito do feromônio no nó e não no caminho (aresta).

No ACO padrão e nos algoritmos antecessores sobre os quais o WFACOFS foi desenvolvido, a atualização de feromônios é feita de forma global e localmente, mas o valor do feromônio não é delimitado. Assim, uma variável escolhida mais vezes adquire um alto valor de feromônio o que leva à sua seleção múltiplas vezes. Eventualmente, isso torna a seleção de variáveis tendencioso e dificulta a exploração de todas as variáveis (GHOSH *et al.*, 2019). Para resolver este problema, foi implementada a normalização dos valores de feromônio para garantir um limite no intervalo]0,1] após cada iteração.

Foi mantido o uso do conceito de distância entre variáveis calculado por similaridade de cossenos que foi apresentado inicialmente no algoritmo UFSACO. De acordo com DBpedia (2020), a similaridade por cossenos é uma medida da similaridade entre dois vetores num espaço vetorial que avalia o valor do cosseno do ângulo compreendido entre eles. Esta função trigonométrica proporciona um valor igual a "1" se o ângulo compreendido é zero, isto é, se ambos os vetores apontam para um mesmo lugar (variáveis completamente correlacionadas) ou "0" se os vetores possuírem ângulo de 90° entre si, indicando, nesse caso, que as variáveis são completamente independentes.

Segundo (GHOSH *et al.*, 2019), o uso da similaridade de cossenos na matriz de distâncias contribui em grande medida para a redução do custo computacional.

A forma de cálculo da similaridade e a descrição matemática do modelo estão descritas nas Equações 2.4 a 2.9. O fluxograma do WFACOFS é representado na Figura

2.2.

$$sim(X,Y) = \frac{\sum_{i=1}^{a} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{a} x_i^2} \cdot \sqrt{\sum_{i=1}^{a} y_i^2}} \tag{2.4}$$

$$\eta_i = \frac{1}{1 + sim(i,j)} \tag{2.5}$$

$$\rho_i(t+1) = (1-d)\rho_i(t) + \sum_{l=1}^{m} \Delta\rho_i^l(t) + \Delta\rho_i^g(t) \tag{2.6}$$

$$\Delta\rho_i^l(t) = \begin{cases} \phi \cdot \gamma(G) + \dfrac{(1-\phi)\cdot(n-|G|)}{n} & , \textit{ se } i \in G \\ 0 & , \textit{ caso contrário} \end{cases} \tag{2.7}$$

$$P_{i \in G} = \frac{[\rho_i(t)]^\alpha \cdot [\eta_i(t)]^\beta}{\sum_{u \in G}([\rho_u(t)]^\alpha \cdot [\eta_u(t)]^\beta)} \tag{2.8}$$

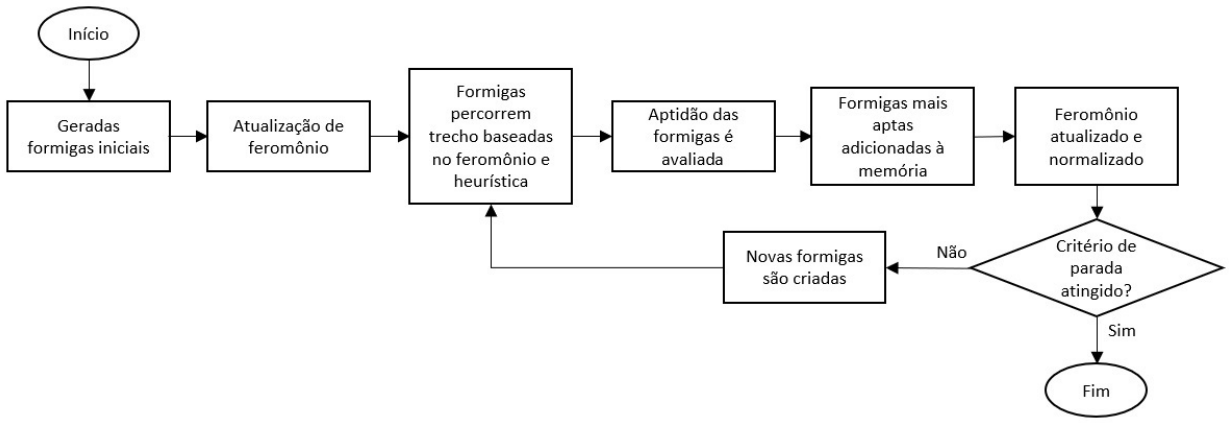
$$fitness(G) = w_1 \cdot \gamma(G) + w_2 \cdot \frac{(n-|G|)}{n} \tag{2.9}$$

em que:

$sim(X,Y)$	similaridade de cossenos
η_i	distância entre variáveis estabelecida na matriz de distâncias
ρ_i	densidade de feromônio num determinado nó i
d	taxa de evaporação do feromônio
$\Delta\rho_i^l$	acréscimo de feromônio causado pela l-ésima melhor formiga
$\Delta\rho_i^g$	acréscimo de feromônio causado pela melhor formiga dentre todas
ϕ	parâmetro de equilíbrio entre acurácia e quantidade de variáveis selecionadas
γ(G)	acurácia calculada sobre o subconjunto de variáveis selecionado

$P_{i \in G}$	probabilidade de escolha da próxima variável (nó)
n	número total de variáveis da base selecionada
G	subconjunto de variáveis selecionadas
w_1	parâmetro de peso para acurácia
w_2	parâmetro de peso para número de variáveis do subconjunto selecionado

Figura 2.2: Fluxograma WFACOFS



Fonte: Adaptado de Ghosh *et al.* (2019)

2.2. Métodos *Filter* e *Wrapper*

De acordo com Ghosh *et al.* (2019), os algoritmos de seleção de variáveis podem ser divididos em três categorias: métodos de filtro (*Filter*), métodos incorporados (*Wrapper*) e métodos combinados (*Wrapper-Filter*). Definição similar é também apresentada por Brownlee (2019b).

Os métodos de filtro funcionam nas propriedades intrínsecas dos dados e não requerem um algoritmo de aprendizado. Isso tende a torná-los muito rápidos, mas a precisão do processo de seleção de variáveis geralmente é menor do que aquela alcançada pelo método *Wrapper* (GHOSH *et al.*, 2019).

Os métodos de seleção de recursos baseados em filtros usam medidas estatísticas para pontuar a correlação ou a dependência entre as variáveis de entrada que podem ser filtradas para escolher os recursos mais relevantes (BROWNLEE, 2019b).

Nos métodos *Wrapper* é utilizado um algoritmo de aprendizado de máquina para a seleção das variáveis e são criados vários modelos com diversas combinações de variáveis e escolhe-se aquelas que resultaram na melhor desempenho para o modelo, de acordo com métricas de avaliação.

Segundo (KUHN e JOHNSON, 2013), os métodos *Wrapper* avaliam vários modelos usando procedimentos que adicionam e/ou removem preditores para encontrar a combinação ideal que maximiza o desempenho do modelo.

Na mesma linha de desenvolvimento adotada no algoritmo WFACOFS, o algoritmo ACOFS$_{rank}$ também considera em sua construção o método combinado *Wrapper-Filter* para seleção das variáveis.

2.3. Cálculo de distâncias - Correlação

De forma simplificada, correlação é definida como a relação de interdependência entre duas ou entre múltiplas variáveis (DICIO, 2021). Numa descrição mais científica, correlação é qualquer relação dentro de uma ampla classe de relações estatísticas que envolva dependência entre duas variáveis (MORETTIN e BUSSAB, 2017).

Conforme (BRUCE *et al.*, 2020, capítulo 7) e (GOMES, 2017), existem duas grandes categorias de variáveis que devem ser abordadas de maneiras diferentes, a saber:

Quantitativas ou Numéricas:

- Contínua: Medidas (ex.: Peso, altura, renda, dinheiro, comprimento);
- Discreta: Contagem (ex.: quantidade de coisas).

Qualitativas ou Categóricas:

- Nominais: Nomes;
- Ordinais: Quando é possível ordenar os arquivos.

Quando duas variáveis quantitativas ou numéricas são apresentadas para análise, a literatura provê diversas maneiras de estabelecer o nível de relacionamento entre elas. Em maior ou menor grau de dificuldade, é sempre possível estabelecer a correlação entre valores numéricos pois esses representam grandezas ou quantidades que são matematicamente demonstradas.

No entanto, no mundo real o espectro de variáveis é mais amplo e, de forma até bastante frequente, é encontrado um outro tipo de dado conhecido como qualitativo ou categórico. Para esse tipo de dado, não é possível aplicar as equações usadas nas análises de dados contínuos, pois dados categóricos, em grande parte, não representam grandezas matemáticas e/ou não estão diretamente associados a quantidades. Mas nem por isso, deixam de ser altamente representativos de situações ou condições do mundo real.

Considerando a necessidade já descrita anteriormente de se criar uma matriz de valores numéricos que represente a "distância" entre as variáveis de uma fonte de dados,

matriz essa que funcionará como hiper parâmetro para o modelo ACOFS$_{rank}$, apresenta-se então um problema que deve ser resolvido: como criar um mecanismo de análise entre variáveis que consiga tratar todos os tipos de dados e definir as distâncias tanto entre variáveis numéricas quanto entre variáveis categóricas?

No campo da computação, essa questão pode ser dividida em dois subproblemas: **i)** testar se existe uma correlação estatisticamente significativa entre duas variáveis; e **ii)** quantificar a associação ou "adequação" entre essas variáveis. Idealmente, faz-se necessário comparar essas métricas de adequação em alguma escala universal.

É possível dividir em três conjuntos a análise da correlação entre variáveis:

- Numéricas vs. Numéricas;
- Categóricas vs. Categóricas;
- Numéricas vs. Categóricas (e vice-versa).

Em geral, saber se duas variáveis estão correlacionadas é útil para entender as estruturas de variância nos dados e a seleção de recursos no aprendizado de máquina. Expandindo o conceito, para exploração eficaz de dados e teste de hipóteses é necessário ser capaz de compreender as associações entre as variáveis.

Além disso, existe vasta literatura reforçando o conceito de que para construir modelos preditivos eficientes, idealmente devem ser considerados apenas variáveis que explicam de forma exclusiva alguma quantidade de variação no resultado. Em todas essas aplicações, é provável que seja necessário a comparação entre pares de dados numéricos, categóricos e numéricos-categóricos entre si e, portanto, ter uma estimativa compartilhada de associação entre pares de variáveis é essencial.

Um ponto adicional a se notar é que, embora um teste de significância estatística de correlação entre as duas variáveis seja útil, é muito mais importante quantificar a associação de uma maneira comparável, quer dizer, ter uma métrica comparativa de adequação.

Neste trabalho, são listadas de forma sucinta algumas métricas comuns para análise da correlação entre as três combinações de tipos de dados e, em seguida, discutidos pontos fortes e fracos relativos a essas métricas em seu conceito amplo. Ao final é apresentada uma técnica unificada para correlacionar todos os três conjuntos, recentemente divulgada no meio acadêmico.

Faz-se necessária a abordagem desses conceitos para estabelecer a fundação da criação da matriz de distâncias, item essencial ao funcionamento do ACO. Para mais informações sobre aplicação e restrições dos métodos apresentados, recomenda-se consulta

ao material de referência bibliográfica apresentado.

2.3.1. Correlação entre variáveis numéricas

Correlacionar duas variáveis numéricas tem sido um problema de longa data e, portanto, ao longo dos anos, várias métricas robustas foram desenvolvidas. Neste trabalho são apresentadas duas abordagens gerais para compreender associações entre variáveis numéricas, a saber: correlações lineares e correlações baseadas em classificação.

Abordagem por correlações lineares

A correlação de Pearson é um dos coeficientes de correlação mais antigos desenvolvidos para medir e quantificar a similaridade entre duas variáveis numéricas.

Conforme Edelmann *et al.* (2021), a correlação de Pearson mede o grau da correlação linear entre duas variáveis quantitativas com dois sinais contínuos que co-variam ao longo do tempo e indicam a relação linear como um número entre -1 (correlacionado negativamente) a 0 (não correlacionado) a 1 (perfeitamente correlacionado).

O uso do coeficiente de Pearson é bastante intuitivo e sua demonstração matemática aplicada a uma amostra está descrita na Equação 2.10. Entretanto, de acordo com Colosimo, algumas considerações devem ser levadas em conta para sua correta utilização, a saber:

1. Forte influência de *outliers*[3];
2. Suposição de linearidade (relacionamento linear entre as variáveis);
3. Suposição de homocedasticidade [4].

À luz dessas suposições, é fortemente recomendável inspecionar essas propriedades antes de usar a correlação de Pearson para quantificar a similaridade, em especial quando se deseja utilizar o coeficiente de Pearson como uma medida de correlação.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{2.10}$$

[3]Os ***outliers*** são valores que apresentam um grande afastamento em relação aos demais da amostra (PORTO, 2020a)

[4]A variância dos erros, condicionada aos valores das variáveis explanatórias, é constante (MAIA, 2019)

em que:

n tamanho da amostra

x_i, y_i valores medidos de ambas as variáveis

$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ e $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ médias aritméticas de ambas as variáveis

Abordagem por correlações baseadas em classificação

Ao desenvolver análises em dados, é frequente o interesse em entender a associação entre variáveis que podem estar relacionadas por meio de uma função não linear. Nestes casos, mesmo quando as variáveis têm uma associação forte, a correlação de Pearson seria baixa.

Para identificar adequadamente a associação entre variáveis com relacionamentos não lineares, podem ser utilizadas abordagens de correlação baseadas em classificação. Os coeficientes de correlação de classificação são índices estatísticos que medem o grau de associação entre duas variáveis com categorias ordenadas (FORTHOFER R.N., 1981)

Abaixo, são apresentados dois exemplos:

1. Correlação de Spearman, (SPEARMAN, 1904);
2. Tau de Kendall, (ABDI, 2007).

Dentre essas técnicas, talvez a mais comumente utilizada por sua fácil interpretação e explicabilidade seja a Correlação de Spearman. O coeficiente foi introduzido em 1904, por Charles Spearman, como um novo método para medir a correlação entre duas variáveis. Ao invés de tomar os valores das variáveis, ele considera as classificações (ou ordens) das observações e calcula o coeficiente de correlação de Pearson para essas classificações. O coeficiente de correlação assim obtido é denominado coeficiente de correlação de classificação e é uma medida útil para lidar com variáveis categóricas (SPEARMAN, 1904).

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d^2}{n(n^2-1)} \tag{2.11}$$

em que:

r_s coeficiente de correlação de Spearman

d representa a diferença nas classificações entre os itens emparelhados

n representa o número de pares de observações

Pontos fortes e fracos das abordagens

A questão mais importante a ser destacada é que a correlação de Pearson é útil na detecção de relações lineares entre variáveis numéricas. Para detectar outras relações devem ser consideradas abordagens baseadas em classificação, como a correlação de Spearman.

2.3.2. Correlação entre variáveis categóricas

Dados categóricos podem ser classificados como ordinais ou nominais. Os dados ordinais possuem uma ordem de classificação para seus valores e, dessa forma, podem ser convertidos em dados numéricos por meio da codificação ordinal (BROWNLEE, 2020b). Como exemplo, é possível considerar as classificações de um teste variando entre A e F, que podem ser definidas usando números entre 1 a 6.

No entanto, para variáveis nominais, a situação é bastante distinta. Uma vez que não há relação quantitativa entre os valores individuais dessas variáveis, o uso da codificação ordinal pode potencialmente criar uma relação ordinal indesejada nos dados (BROWNLEE, 2020b) e danosa aos resultados finais.

Em termos gerais, pode ser considerado o conjunto de abordagens que abrange métricas estatísticas como uma forma eficiente de encontrar associação entre variáveis categóricas.

Abordagem por métricas estatísticas - tabelas de contingência

Ao comparar duas variáveis categóricas, contando as frequências das categorias é possível converter os vetores originais em tabelas de contingência.

Como exemplo conceitual, considere o problema prático de avaliar se existe correlação entre ser homem e conseguir uma bolsa de estudos. Os dados podem ser formatados como apresentado nas Tabelas 2.2 e 2.3.

Segundo Pearson (1904), as tabelas de contingência ou tabulação cruzada exibem a distribuição de frequência multivariada de variáveis e são amplamente utilizadas em pesquisas científicas entre disciplinas. Devido ao seu forte uso histórico em análises estatísticas, uma família de testes foi desenvolvida para determinar a significância da diferença entre duas categorias de uma variável em comparação com outra variável categórica. Uma abordagem popular para variáveis dicotômicas (ou seja, variáveis com apenas duas

categorias) é construída na distribuição *qui-quadrado*[5].

Tabela 2.2: Exemplo de dados originais de categorias

Sexo	**Aprovação**
Masculino	Sim
Masculino	Não
Feminino	Sim
Masculino	Sim
...	...
Feminino	Não
Masculino	Sim
Feminino	Não
Feminino	Não
Masculino	Sim

Tabela 2.3: Exemplo de tabela de contingência

		Sexo	
		Masculino	**Feminino**
Aprovação	Sim	100	40
	Não	600	500

Neste trabalho, o interesse principal não se pauta em testar a significância estatística, mas sim o tamanho do efeito e, especificamente, a força da associação entre as duas variáveis. Vários coeficientes já foram definidos na literatura para esse propósito, incluindo alguns que usam a distribuição qui-quadrado. Seguem dois exemplos:

- Lambda de Goodman-Kruskal, (GOODMAN e KRUSKAL, 1979);
- Coeficiente Phi (usa distribuição qui-quadrado), (GLEN, 2016; GUILFORD, 1941).

Pontos fortes e fracos

Embora a literatura indique que as técnicas estatísticas baseadas na análise de tabelas de contingência apresentam menos desvantagens em comparação com outras métricas, existem questões importantes que surgem principalmente sobre como o teste de significância estatística é convertido em uma medida de associação.

[5]Teste qui-quadrado (em ingles *chi-square*) também representado por χ^2, representa o somatório dos desvios entre as frequências observadas e as esperadas teoricamente, para um dado evento (PORTO, 2020b).

2.3.3. Correlação entre variáveis numéricas e categóricas

De todos os coeficientes de correlação que se faz necessário estimar para uma fonte de dados mista, talvez este provavelmente seja o mais complicado com o menor número de opções apresentadas na literatura.

Numa visão geral, serão considerados neste trabalho dois métodos para entender se um dado contínuo e um dado categórico estão significativamente correlacionados: correlação bisserial de ponto e regressão logística.

Correlação Bisserial de Ponto

Segundo Calkins (2005), o coeficiente de correlação bisserial de ponto é um caso especial do coeficiente de correlação de Pearson em que uma variável é quantitativa e a outra variável é dicotômica e nominal. Os cálculos são simplificados, pois normalmente os valores 1 (presença) e 0 (ausência) são usados para a variável dicotômica. O método de cálculo da correlação Bisserial de Ponto é apresentado na Equação 2.12.

$$r_{pb} = \frac{Y_1 - Y_0}{s_n}\sqrt{\frac{n_1 \cdot n_0}{n^2}} \tag{2.12}$$

$$s_n = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n}(X_i - \overline{X})^2}$$

em que:

r_{pb}	coeficiente de correlação bisserial de ponto
Y_1 e Y_0	médias Y para os pares de dados dos grupos 1 e 0, respectivamente
n_1 e n_0	número de pontos de dados dos grupos 1 e 0, respectivamente
n	tamanho total da amostra
s_n	desvio padrão considerando todos os membros da população

Para a correta utilização do coeficiente bisserial de ponto faz-se necessário destacar dois pontos de observação, a saber:

- De forma similar ao Coeficiente de Pearson, pode variar no intervalo [-1,1];
- Assume que a variável contínua possui distribuição normal e é homocedástica.

Regressão Logística

A ideia por trás do uso de regressão logística é que se houver uma relação entre as variáveis categórica e contínua, deve ser possível construir um preditor preciso da variável categórica a partir da variável contínua. Se o classificador resultante tem um alto grau de ajuste, é preciso, sensível e específico, pode-se concluir que as duas variáveis compartilham uma relação e são de fato correlacionadas.

A regressão logística não faz muitas das suposições principais de regressão linear, particularmente no que diz respeito à linearidade, normalidade, homocedasticidade e nível de medição. No entanto, é necessário observar que a regressão logística assume que há uma relação linear entre os preditores e o log da variável de resultado e, muitas vezes, não apenas essa suposição é inválida, como também não é fácil de verificar.

Pontos fortes e fracos

A correlação Bisserial de Ponto é a mais intuitiva das opções apresentadas para medir a associação entre variáveis numéricas e categóricas. Possui pontos fortes óbvios - uma forte semelhança com a correlação de Pearson e tem baixo custo computacional. Por outro lado, assume suposições fortes sobre os dados relativos à normalidade e homocedasticidade. Além disso, semelhante à correlação de Pearson, é útil apenas na captura de relacionamentos um tanto lineares entre as variáveis. Para muitas aplicações, essas questões são demasiadamente restritivas e tornam o uso da correlação Bisserial de Ponto uma escolha inadequada.

A regressão logística é mais robusta, principalmente porque as variáveis numéricas não precisam ser normalmente distribuídas ou ter variância igual em cada grupo. Ademais, também pode ajudar a modelar e detectar relações não lineares entre as variáveis categóricas e numéricas. Um grande problema é que a regressão logística, como muitos outros classificadores, é sensível aos desequilíbrios de classe.

2.3.4. Método phik - ϕ_K

Em 2020 foi apresentado um método unificado de cálculo de correlação que é capaz de apresentar resultados satisfatórios para os três conjuntos de variáveis discutidos anteriormente neste trabalho. O trabalho acadêmico apresentado por Baak *et al.* (2020) estabelece um novo coeficiente de correlação, denominado ϕ_K baseado em vários refinamentos ao teste de hipótese de Pearson de independência de duas variáveis.

Segundo Baak *et al.* (2020), o coeficiente ϕ_K apresenta características combinadas que trazem vantagens em relação aos coeficientes existentes:

- Funciona consistentemente entre variáveis categóricas, ordinais e intervalares;

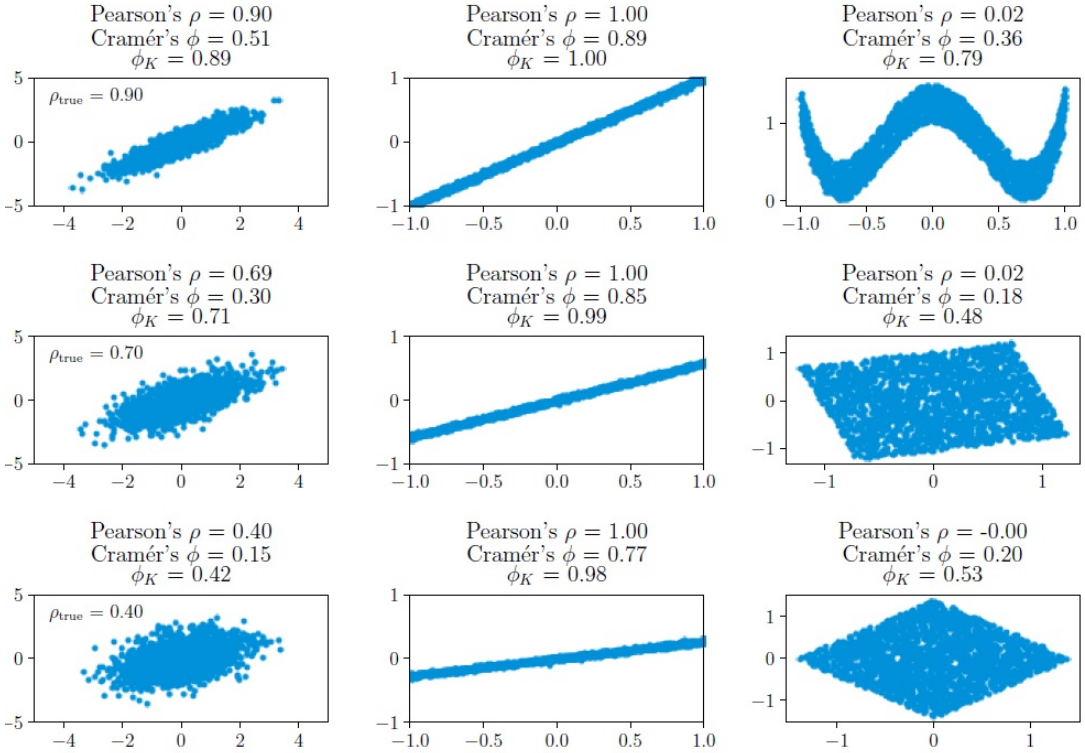
- Captura a dependência não linear;
- Reverte para o coeficiente de correlação de Pearson no caso de uma distribuição de entrada normal bivariada.

A correlação ϕ_K segue um tratamento uniforme para variáveis intervalares, ordinais e categóricas. Isso é particularmente útil na análise moderna ao estudar as dependências entre um conjunto de variáveis com tipos mistos, no qual algumas variáveis são categóricas. Os valores para níveis de correlação estão limitados no intervalo [0; 1], com 0 atribuído quando não há nenhuma associação e +1 para associação completa. Por construção, a interpretação é semelhante ao coeficiente de correlação de Pearson e é equivalente no caso de uma distribuição de entrada normal bivariada. Ao contrário de Pearson, que descreve a dependência linear média entre duas variáveis, ϕ_K também captura relações não lineares (BAAK *et al.*, 2020).

Uma comparação com coeficientes de correlação alternativos baseados em amostras de referência é fornecida na Figura 2.3.

Em função dessas vantagens e da não necessidade de se ter que trabalhar com vários métodos de definição de correlação, foi adotado o método ϕ_K para a criação da tabela de distâncias necessária ao modelo ACOFS$_{rank}$.

Figura 2.3: Resultados do coeficiente ϕ_K para amostras de referência.



Fonte: (BAAK *et al.*, 2020)

Capítulo 3

Materiais e Métodos

A aplicação do algoritmo ACOFS$_{rank}$ para execução da seleção de variáveis que tragam melhor desempenho aos modelos de aprendizado de máquina baseados em classificação requer a avaliação prévia dessas variáveis, bem como a construção de estruturas e mecanismos de suporte essenciais ao perfeito funcionamento do próprio algoritmo.

Parte-se da premissa que a base já tenha passado pelos processos iniciais de transformação e tratamento de dados com objetivo de torná-la consistente e apta para uso nos algoritmos de ML. Situações de anomalias, ausência de dados, erros na descrição, desbalanceamento de dados, dentre outros problemas usualmente encontrados nas bases de dados originais já estariam resolvidos ou, pelo menos, mitigados em grande parte. Da mesma forma, também considera-se que análises estatísticas para validação da normalidade dos dados também já tenham sido aplicadas no pré-tratamento.

Os códigos criados para implementação do algoritmo foram desenvolvidos em linguagem Python. Tanto os códigos quanto os principais artigos e materiais de referência que contribuíram para este projeto encontram-se disponíveis no GitHub do autor (DELAMORA, 2023).

3.1. Preparação dos dados para uso do modelo

Uma vez resolvida essa fase, a metodologia proposta para o problema consiste em duas macro etapas, sendo a primeira responsável pelo mapeamento do tipo de cada uma das variáveis, que podem ser classificadas como numéricas ou categóricas. Esse mapeamento será a base para a definição sobre que tipo de método estatístico de cálculo de correlação melhor se encaixa na validação dos pares de variáveis.

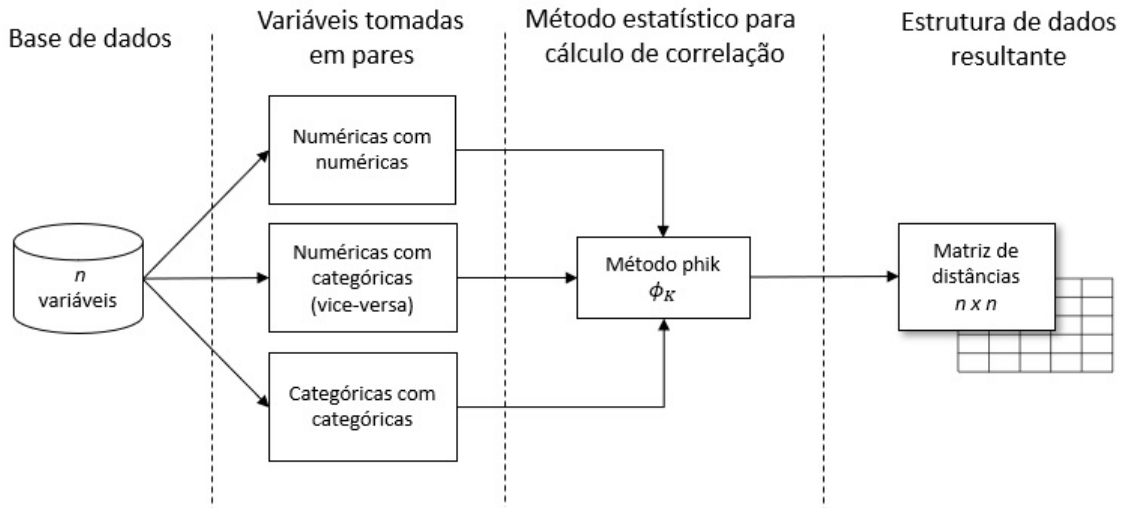
Como apresentado na revisão bibliográfica, o método phik consegue cobrir todos os tipos de variáveis permitindo que a matriz de distâncias possa ser construída a partir

de conjuntos de dados multi-tipo.

O cálculo de correlação é realizado tomando pares de variáveis e, com os resultados, é montada a matriz com estrutura similar à matriz de distâncias utilizada no TSP, ressaltando que as linhas e colunas representam as variáveis apresentadas na base de dados. A descrição desse processo é apresentada na Figura 3.1.

A proposta é que todo o processo de mapeamento das variáveis por tipo, cálculo das correlações e construção da matriz aconteça de forma automatizada. Para isso é utilizada uma biblioteca em Python que foi desenvolvida e disponibilizada no modelo *open source* à comunidade de desenvolvedores (BAAK *et al.*, 2018).

Figura 3.1: Fluxograma da macro etapa de montagem da matriz de "distâncias".



Fonte: Compilação do autor

Como o valor da correlação pode assumir valores entre $[0, 1]$ e o valor 0 não é desejado, pois pode gerar erros de divisão por zero durante a execução do algoritmo, é aplicada à matriz uma operação de redimensionamento (*re-scaling*[1]) para que todos os valores estejam no intervalo entre $[1, 10]$. Essa operação segue o cálculo apresentado na Equação 3.1.

$$vlr_ajustado = \frac{9 \cdot (valor - vlr_min)}{vlr_max - vlr_min} + 1 \tag{3.1}$$

[1]Alteração da escala dos valores preservando, no entanto, as proporções iniciais (COLLINS, 2021)

em que:

$vlr_ajustado$	valor de correlação após processo de *re-scaling*
$valor$	valor de correlação original
vlr_min	menor valor encontrado na matriz
vlr_max	maior valor encontrado na matriz

3.2. Ajustes no algoritmo *Rank Based AS* (AS_{rank})

O algoritmo *Rank Based AS* (AS_{rank}), desenvolvido por Bullnheimer *et al.* (1997), foi escolhido para ser a base para este trabalho por apresentar características favoráveis quando comparado com versões anteriores do AS: **(i)** excelente desempenho na resolução do TSP e **(ii)** rapidez na convergência para boas soluções.

No desenvolvimento do ACOFS_{rank} foram realizados ajustes para adaptar o algoritmo AS_{rank} original que foi desenvolvido para o TSP às novas necessidades apresentadas no problema de seleção de variáveis. Na versão original, o peso das arestas recebia o valor das distâncias entre as cidades e, nesta versão, ele foi substituído pelo valor de correlação entre as variáveis.

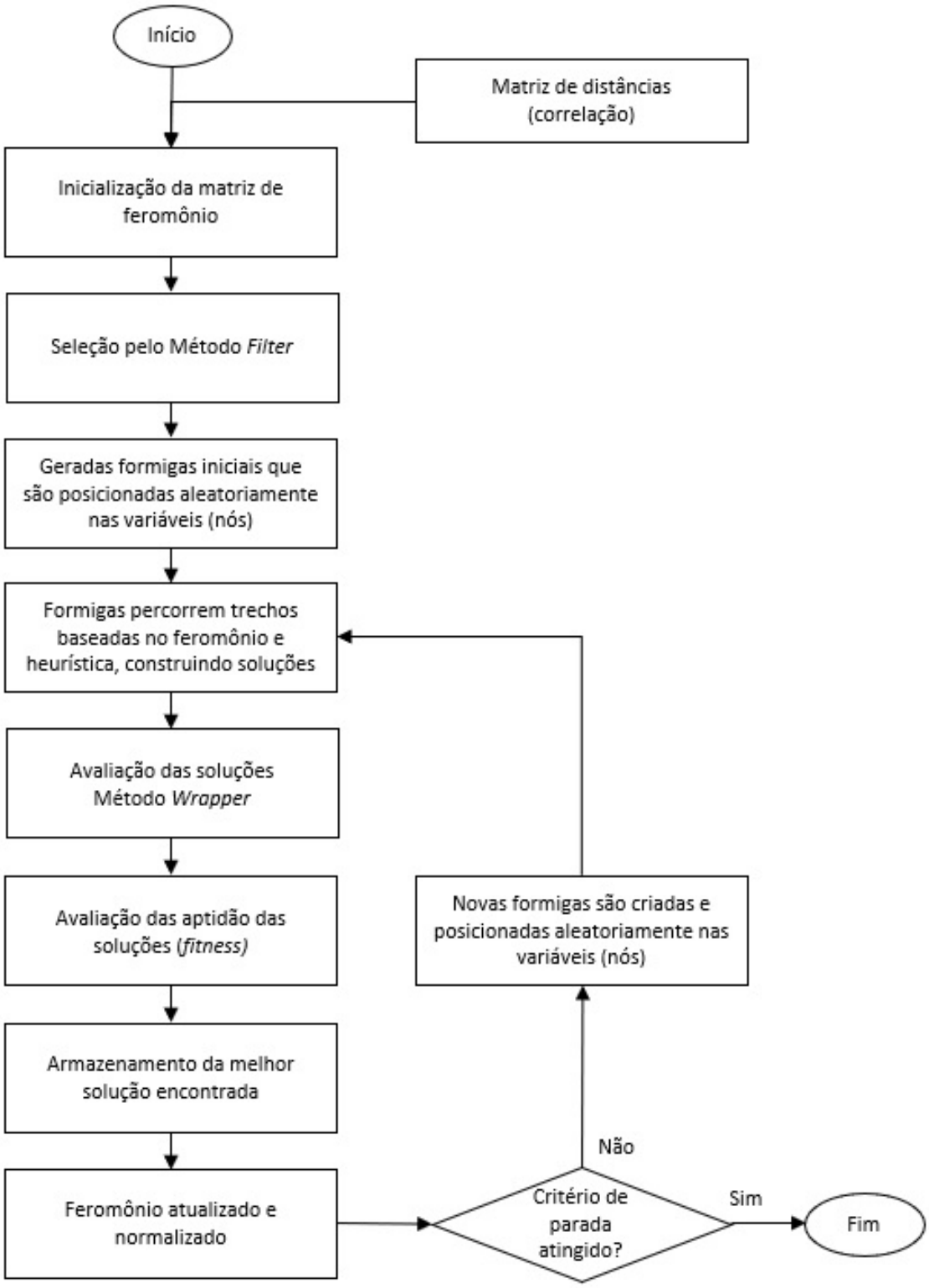
O fluxograma básico do algoritmo é mostrado na Figura 3.2 e seu pseudo-código genérico é ilustrado no Algoritmo 3.

Assim como no algoritmo original, a matriz de distâncias é um dado externo ao algoritmo e deve ser provido como parâmetro desse. Desta forma, a construção dessa matriz não faz parte do processamento executado pelo ACOFS_{rank} em si.

Com a matriz de distâncias disponível, se inicia a segunda macro etapa com a execução do algoritmo do ACOFS_{rank}.

Após a inicialização dos parâmetros e da matriz de feromônio, já acontece a etapa inicial de seleção de variáveis através do método *Filter*.

Figura 3.2: Fluxograma ACOFS$_{rank}$



Fonte: Compilação do autor

Algoritmo 3: $ACOFS_{rank}$

```
1:  Início
2:      Inicializar parâmetros
3:      Inicializar matriz de feromônio
4:      Primeira etapa de seleção - Método Filter
5:      repetir até critério de seleção satisfeito
6:          repetir para cada formiga
7:              Construir solução
8:              Calcular métrica da solução - Método Wrapper
9:          até final
10:         Identificar a melhor solução
11:         Armazenar a melhor solução
12:         Atualizar e normalizar feromônio
13:     até final
14: Fim
```

Fonte: Compilação do autor

3.2.1. Método *Filter*

O principal objetivo a ser alcançado nesta etapa é produzir uma redução minoritária na quantidade total de variáveis eliminando aquelas que, de forma estatisticamente mais evidente, não agregam valor ou possuem baixa influência sobre a variável resposta.

Como apresentado no capítulo de Revisão Bibliográfica, o método *Filter* trabalha na avaliação das propriedades intrínsecas dos dados e não requer um algoritmo de aprendizado de máquina o que o torna muito rápido na execução.

A implementação prática do método *Filter* foi realizada utilizando o teste estatístico ANOVA (*Analysis Of Variance*) levando em conta o coeficiente *F-score* como métrica de validação. Segundo Santos (2021), ANOVA é um método estatístico para verificar se existem diferenças significativas entre as médias de grupos de dados, sendo possível inferir se as variáveis são dependentes uma da outra.

Também como apresentado por (BRUCE *et al.*, 2020, capítulo 3) e Gajawada (2019), a variância de um recurso determina o quanto ele está impactando a variável de resposta. Se a variância for baixa, significa que não há impacto significativo dessa característica na resposta e vice-versa.

A ANOVA calcula a relação da variância entre grupos dividida pela variância dentro dos grupos como descrito na Equação 3.2. Desse modo, é possível dizer que quanto

maior a variância entre os grupos, mais diferentes as duas variáveis serão e maior será *F-score* (SANTOS, 2021).

$$F = \frac{Variância\,entre\,grupos}{Variância\,dentro\,dos\,grupos} \tag{3.2}$$

O *F-score* é um método de seleção de características uni-variadas, o que significa que pontua cada uma das características (x1, x2, x3, ...) individualmente sem considerar que uma característica pode apresentar resultados melhores se combinada com outras. Quanto maior *F-score*, maior a probabilidade de esse recurso ser mais discriminativo (CHEN e LIN, 2006).

A ANOVA utiliza a verificação pela tabela F-test para validar se existe alguma diferença significativa entre os grupos de valores que compõem uma variável. Se não houver diferença significativa entre os grupos, pode-se assumir que estatisticamente todas as variâncias são iguais. Esta variável deve então ser removida do modelo (GAJAWADA, 2019).

Segundo (MILONE, 2004, capítulo 12), os pressupostos básicos da análise de variância são:

- As amostras são aleatórias e independentes;
- As populações têm distribuição normal (o teste é paramétrico);
- As variâncias populacionais são iguais;

Ainda de acordo com (MILONE, 2004, capítulo 12), na prática, esses pressupostos não precisam ser todos rigorosamente satisfeitos. Os resultados são empiricamente verdadeiros sempre que as populações são aproximadamente normais e têm variâncias próximas.

Como descrito na Sessão 1.4 de Delimitações do estudo deste trabalho, não é considerado como escopo desta pesquisa a implementação de processos de análise de normalidade das bases de dados. As validações e correções necessárias nos dados devem ser realizadas na etapa de pré-tratamento que acontece antes da aplicação do algoritmo.

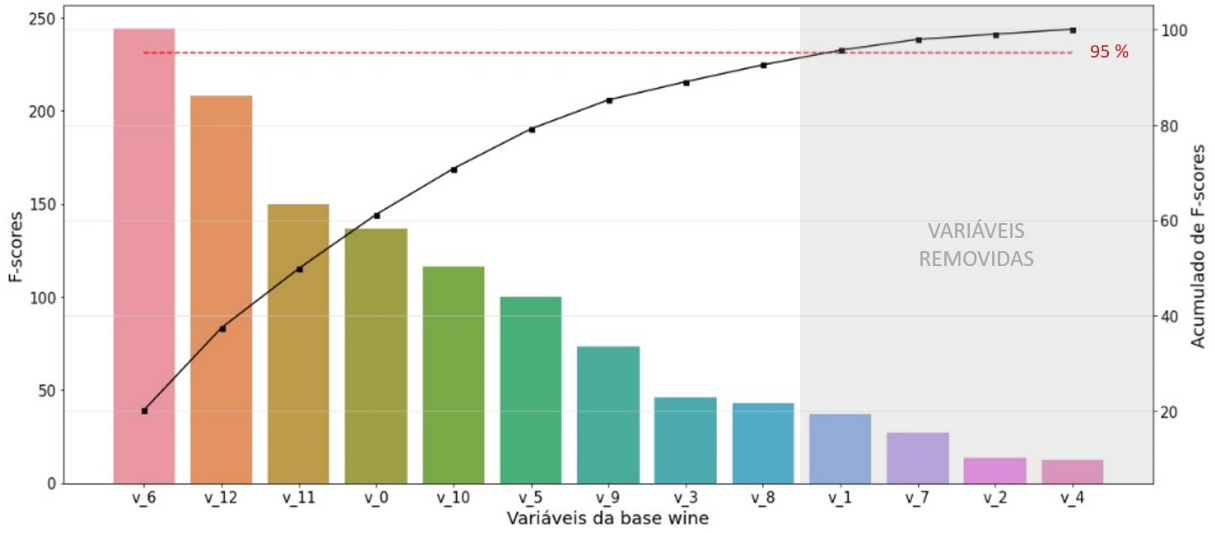
Uma vez aplicada a ANOVA para comparar cada variável independente com a variável resposta obtém-se então um coeficiente *F-score* associado a cada uma. Este coeficiente representa a influência do comportamento da variável no comportamento da variável resposta ou, em outras palavras, representa a parcela de explicabilidade do comportamento da variável resposta que é atribuída a esta variável sob análise.

Neste trabalho, considera-se que serão mantidas as variáveis que, no acumulado das somas dos *F-score* individuais, representam 95% da explicabilidade da variável resposta. Desta forma, é realizada a remoção de variáveis pouco importantes do ponto de vista estatístico sem que ocorra perda significativa da informação.

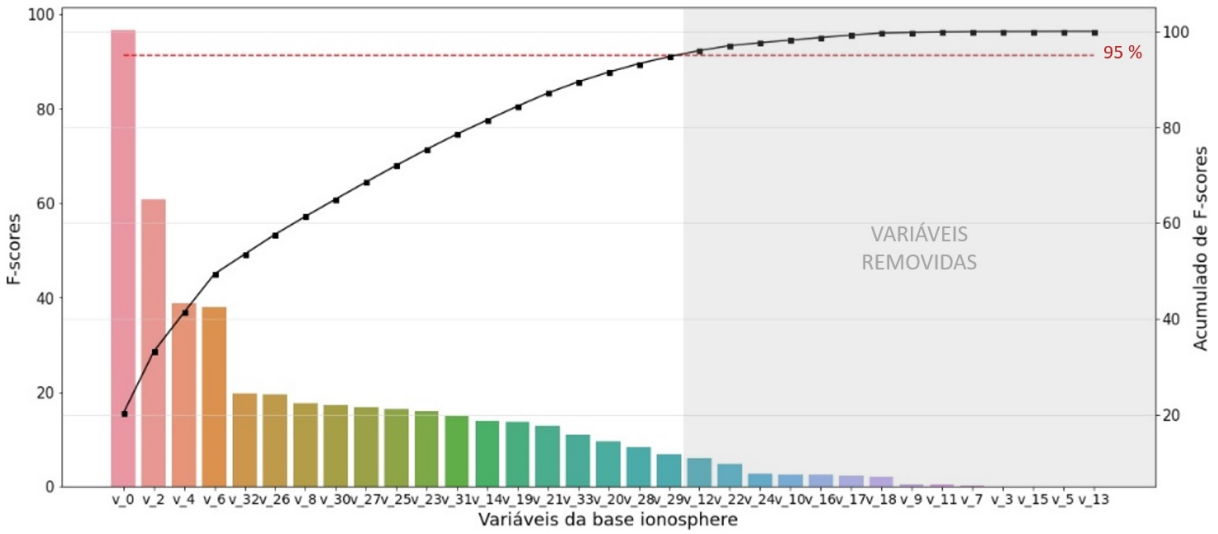
Um exemplo do processo está descrito nas Figuras 3.3a e 3.3b que apresentam gráficos com os valores calculados e acumulados de *F-score* para cada uma das variáveis das bases *Wine* e *Ionosphere*, respectivamente. A linha pontilhada indica o limiar de corte de 95%.

Figura 3.3: Representação da análise de *F-score* para cada variável através da ANOVA

(a) Base de dados *Wine*.



(b) Base de dados *Ionosphere*.



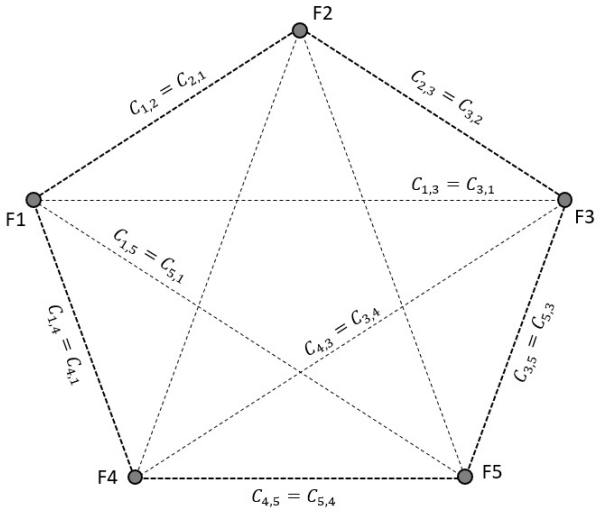
Fonte: Compilação do autor

Após a seleção das variáveis na etapa do método *Filter*, é gerada nova base de dados somente com as variáveis restantes, que segue então o fluxo de análises e processamentos do algoritmo.

3.2.2. Etapa de construção

Um exemplo do espaço das soluções, representado na Figura 3.4, consiste de um grafo completo, não direcionado com pesos, onde os nós representam as variáveis da base de dados inicial e os pesos das arestas as correlações ($C_{i,j}$) entre cada par de variáveis. Cada nó está associado a uma quantidade de feromônio, que é avaliado pelas formigas. Para cada variável é atribuída uma formiga (F_m) e essas formigas percorrem o grafo dando preferência a altos valores de feromônios e baixas correlações. Ao final do processo, as variáveis com maior incidência de feromônio são exatamente as que trazem mais informações ao sistema, levando ao descarte das demais (SOLORIO-FERNÁNDEZ *et al.*, 2020).

Figura 3.4: Representação através de grafo de um problema de seleção de variáveis



Fonte: Adaptado de Tabakhi *et al.* (2014)

Partindo de uma variável inicial diferente, cada formiga escolhe a próxima variável a ser visitada considerando uma probabilidade que é uma função do valor da correlação entre as variáveis e da quantidade de feromônio residual presente na aresta que se conecta a esta variável. O cálculo da probabilidade segue a mesma equação original descrita por Dorigo (1991) e é definido pela Equação 3.3.

$$p_{ij}^{k} = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}}{\sum_{h \in \Omega}([\tau_{ih}(t)]^{\alpha} \cdot [\eta_{ih}(t)]^{\beta})} \tag{3.3}$$

$$\eta_{ij} = \frac{1}{d_{ij}}$$

em que:

τ_{ij} intensidade do feromônio presente na rota entre as variáveis i e j

α parâmetro que regula a influência do feromônio

β parâmetro que regula a influência da correlação entre as variáveis i e j

d_{ij} valor de correlação entre as variáveis i e j

η_{ij} visibilidade entre as variáveis i e j

Ω lista de variáveis ainda não visitadas pela formiga

O número de variáveis que cada formiga percorre é definido de forma aleatória podendo assumir valores entre o mínimo de variáveis firmado e o máximo de variáveis disponíveis na base de dados resultante do método *Filter*. Foi estabelecido um número mínimo de 3 variáveis considerando-se que não faz sentido do ponto de vista estatístico permitir que seja criado um subconjunto com número de variáveis inferior a este.

Mesmo se tratando de processo estocástico, ou até mesmo por causa disso, a construção de uma solução por uma formiga pode gerar a situação indesejada de se ter soluções exatamente iguais a outras já geradas anteriormente por outras formigas. Nestes casos, a solução duplicada é descartada e uma nova opção é então construída, seguindo os mesmos preceitos de aleatoriedade na definição de cada variável ou nó do grafo. Este processo visa aumentar a exploração do espaço de soluções evitando que o algoritmo permaneça preso numa determinada região de buscas.

Assim como acontece no AS_{rank}, é mantida em cada formiga uma memória sobre as variáveis já visitadas, evitando assim que ocorram repetições de variáveis num mesmo trecho percorrido. Da mesma forma, o feromônio só é atualizado ao final da fase de construção levando em conta o incremento e a perda por evaporação, num processo que é realizado por todas as formigas.

No AS_{rank}, sempre a melhor trilha (menor distância percorrida) global é utilizada para atualizar os níveis de feromônios, o que caracteriza uma estratégia elitista. Além

disso, somente algumas das melhores formigas da iteração atual são habilitadas a adicionar feromônios. Para que isso aconteça, as formigas são classificadas pela distância da trilha ($d^1(t) \leq d^2(t) \leq \cdots \leq d^m(t)$), e a quantidade de feromônio que uma formiga pode depositar é definida de acordo com o índice de ranqueamento r. Somente as $(\omega - 1)$ melhores formigas de cada iteração podem depositar feromônio. A melhor solução global, aquela que fornece o retorno mais forte, recebe o peso ω. A r-ésima melhor formiga da iteração contribui para a atualização do feromônio com um peso dado por $max\{0, \omega - r\}$ (STÜTZLE e DORIGO, 1999).

No ACOFS$_{rank}$ o conceito básico continua sendo o mesmo, mas alguns ajustes foram implementados na estrutura da equação de atualização do feromônio.

As Equações 3.4, 3.5 e 3.6 descrevem o modelo original desenvolvido no AS$_{rank}$ enquanto que a versão ajustada utilizada no ACOFS$_{rank}$ é apresentada nas Equações 3.7 e 3.8

- Equações originais do AS$_{rank}$:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \sum_{r=1}^{\omega-1} \Delta\tau_{ij}^{r} + \Delta\tau_{ij}^{*} \tag{3.4}$$

$$\Delta\tau_{ij}^{r} = \begin{cases} (\omega - r) \cdot \dfrac{Q}{L_r} & \textit{, se r-ésima formiga fez o trecho (i, j)} \\ 0 & \textit{, caso contrário} \end{cases} \tag{3.5}$$

$$\Delta\tau_{ij}^{*} = \begin{cases} \omega \cdot \dfrac{Q}{L^*} & \textit{, se trecho (i, j) faz parte da melhor solução} \\ 0 & \textit{, caso contrário} \end{cases} \tag{3.6}$$

- Equações usadas no ACOFS$_{rank}$:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{r=1}^{\omega} (\omega - r) \cdot \Delta\tau_{ij}^{r}(t) + \omega \cdot \gamma_{best} \tag{3.7}$$

$$\Delta\tau_{ij}^{r}(t) = \begin{cases} \phi \cdot \gamma(G) + \dfrac{(1-\phi) \cdot (n - |G|)}{n} & \textit{, se r-ésima formiga} \in G \\ 0 & \textit{, caso contrário} \end{cases} \tag{3.8}$$

em que:

n	número total de formigas
r	índice de ranqueamento
L_r	distância percorrida pela r-ésima melhor formiga
L^*	distância percorrida pela melhor formiga dentre todas
Q	quantidade de feromônio depositada por uma formiga
$\Delta\tau_{ij}^r$	acréscimo de feromônio causado pela r-ésima melhor formiga
$\Delta\tau_{ij}^*$	acréscimo de feromônio no trecho (i, j) causado pelas formigas elitistas
ω	número de formigas elitistas
ϕ	constante de balanço de acurácia
G	subgrupo de variáveis selecionadas
$\gamma(G)$	valor de acurácia do subgrupo de variáveis definido pela r-ésima melhor formiga
γ_{best}	valor de acurácia da formiga com melhor desempenho

A Equação 3.8, desenvolvida originalmente para o WFACOFS, foi incorporada ao ACOFS$_{rank}$ por realizar de forma bastante eficaz o balanceamento entre acurácia e número de variáveis do subconjunto, para compor o valor de atualização do feromônio feito pelas formigas elitistas.

Cabe ao método *Wrapper* o cálculo das métricas estatísticas (acurácia, f1-score e aptidão) para cada subconjunto (solução). Detalhes sobre o processo são explicados em sessão específica sobre o método neste trabalho. Após esta seleção, a matriz de feromônio é então atualizada e a melhor solução é armazenada na memória do algoritmo.

Um novo ciclo ou iteração é então iniciado com novas formigas sendo construídas e é mantida a matriz de feromônios. Dessa forma, a matriz de feromônios funciona como uma memória das soluções mapeadas no ciclo anterior e possui a identificação daquelas consideradas as melhores.

Assim como descrito em Ghosh *et al.* (2019), o problema que se apresenta é que as soluções com maior quantidade de feromônio irão atrair com mais intensidade as novas formigas e isso pode reduzir a capacidade de exploração do espaço-solução. Mesmo considerando a componente estocástica do modelo, se uma determinada variável for escolhida muitas vezes, o valor do feromônio aumentará drasticamente levando a um baixo interesse das formigas em explorar novas opções.

Para mitigar esse efeito, a cada iteração, as formigas são alocadas aleatoriamente

nas variáveis e a matriz de feromônio é normalizada dividindo cada valor pelo valor máximo encontrado na matriz. Isso faz com que o valor do feromônio sempre permaneça entre $[0, 1]$.

3.2.3. Método *Wrapper*

O principal objetivo do ACOFS$_{rank}$ é selecionar subconjuntos de variáveis com baixa correlação entre si e alta aptidão. Para isso é utilizado um algoritmo classificador que calcula a acurácia do subconjunto na predição da variável alvo ou variável resposta da base de dados. No desenvolvimento da pesquisa, foram selecionados quatro classificadores diferentes para essa função, sendo definidos como parâmetros os valores padrão em todos eles, a saber:

- KNN Classifier (BROWNLEE, 2020a);
- MLP Classifier (FERREIRA, 2019; MOHANTY, 2019; MOREIRA, 2018);
- XGBoost Classifier (BROWNLEE, 2020a; CHEN e GUESTRIN, 2016);
- Random Forest Classifier (BROWNLEE, 2020a; HO, 1995).

A escolha do classificador é feita através de hiperparâmetro passado ao modelo ACOFS$_{rank}$ e somente um classificador é executado por vez. A proposta de implementar classificadores diferentes visa prover ao ACOFS$_{rank}$ melhor adaptabilidade na análise e seleção de variáveis em bases de dados com estruturas diferentes.

A acurácia foi definida como métrica estatística para avaliação dos subconjuntos (soluções) por ser amplamente utilizada na literatura em comparações entre resultados de modelos e por expressar a proximidade de um resultado com o seu valor de referência real. Desta forma, quanto maior o nível de acuracidade, mais próximo da referência ou valor real é o resultado encontrado. Além disso, é também a métrica de desempenho utilizada no modelo WFACOFS (GHOSH *et al.*, 2019). O cálculo é demonstrado na Equação 3.9.

$$Acurácia = \frac{(VP + VN)}{(VP + VN + FP + FN)} \tag{3.9}$$

em que:

VP verdadeiro positivo

VN verdadeiro negativo

FP falso positivo

FN falso negativo

Uma vez selecionados os subconjuntos de variáveis pelas formigas, eles são submetidos ao modelo de classificação definido, que calcula as métricas estatísticas e de controle associadas a cada solução (acurácia, f1-score e *fitness*) e armazena o resultado associando-o à formiga que construiu a solução.

3.2.4. Avaliação de aptidão ou *fitness*

Mesmo soluções com números de variáveis diferentes podem apresentar valores de acurácia idênticos. Isso pode representar um problema ao modelo na definição de qual seria a melhor opção.

Além disso, é necessário levar em consideração que somente a acurácia não é suficiente para garantir as melhores soluções, principalmente quando se trabalha com bases desbalanceadas. Nessas situações o "Paradoxo da Acurácia" deve ser levado em conta.

Em linhas gerais, o "Paradoxo da Acurácia" pode ser definido como a situação contraditória na qual uma acurácia elevada em um modelo de classificação pode evidenciar uma falha no próprio modelo em realizar predições de fato significativas (AZANK, 2020).

De forma a mitigar os problemas acima citados, foi implementada no algoritmo, dentro do método *Wrapper*, uma função de validação da aptidão das soluções, também chamada de função de *fitness*.

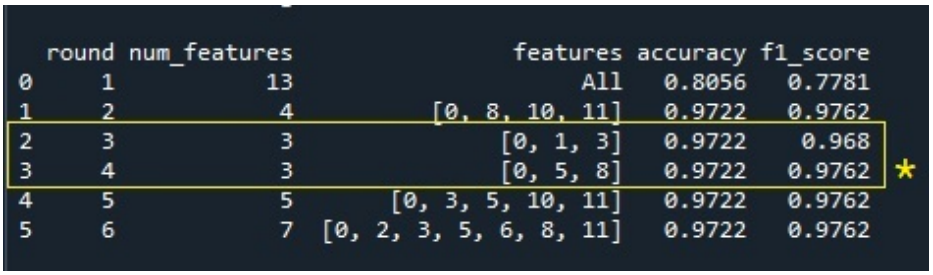
Para sua implementação, é utilizada a medida estatística F-measure (também conhecida como F1-score) que é a média harmônica entre a precisão (*precision*) e a revocação (*recall*) e pode ser interpretada como uma medida de confiabilidade da acurácia. Um valor alto nesta medida significa que a acurácia é relevante (BRUCE *et al.*, 2020, capítulo 5) (SILVA, 2018).

Como mostrado na Equação 3.10, a função de aptidão considera a relação entre os valores de acurácia, a medida F-measure e o número de variáveis desse mesmo subconjunto. O cálculo da acurácia e de F-measure é feito a partir do subconjunto de variáveis selecionado no método *Wrapper* em cada iteração. Ao final das iterações, as soluções são então ordenadas de forma decrescente a partir do valor de *fitness* e a melhor dentre todas as soluções é então armazenada.

Uma otimalidade baseada na aptidão permite salvar os melhores subconjuntos de variáveis em relação à acurácia e ao número de variáveis. Um exemplo de todo este contexto pode ser visto na Figura 3.5 que apresenta soluções geradas pelo algoritmo com a mesma quantidade de variáveis, os mesmos valores de acurácia, mas com valores diferentes

de F-measure (ou F1-score). Neste exemplo, a melhor solução é a que apresenta maior valor de F-measure.

Figura 3.5: Exemplo de situação em que a função de aptidão é utilizada.



	round	num_features	features	accuracy	f1_score	
0	1	13	All	0.8056	0.7781	
1	2	4	[0, 8, 10, 11]	0.9722	0.9762	
2	3	3	[0, 1, 3]	0.9722	0.968	
3	4	3	[0, 5, 8]	0.9722	0.9762	*
4	5	5	[0, 3, 5, 10, 11]	0.9722	0.9762	
5	6	7	[0, 2, 3, 5, 6, 8, 11]	0.9722	0.9762	

Fonte: Compilação do autor

Na estrutura da função de aptidão, a acurácia recebe maior prioridade do que F-measure e o número de variáveis selecionados, através da atribuição de um peso maior ao termo de acurácia (w_1). A Equação 3.10 fornece a função objetivo na determinação da aptidão de um subconjunto de variáveis G.

Com a aplicação da equação de aptidão, soluções que apresentem a mesma acurácia, mesmo número de variáveis selecionadas e valores de F-measure diferentes apresentarão valores de aptidão diferentes, sendo tanto maiores os valores quanto menor for o número de variáveis e maior o F-measure. Isto torna as soluções com menos variáveis e maior precisão mais atrativas.

$$f_s = w_1 \cdot \gamma(G) + F1(G) + w_2 \cdot e^{-\frac{r}{n}} \tag{3.10}$$

em que:

f_s	valor do componente de aptidão (*fitness*) relativo ao número de variáveis
w_1	parâmetro de peso para acurácia
w_2	parâmetro de peso para número de variáveis do subconjunto selecionado
$\gamma(G)$	valor de acurácia do subgrupo de variáveis definido no método *Wrapper*
$F1(G)$	valor de F-measure do subgrupo de variáveis definido no método *Wrapper*
n	número total de variáveis do conjunto de dados
r	número de variáveis da solução criada pela formiga

3.2.5. Descrição do funcionamento geral do algoritmo

Esta sessão traz uma visão bastante sumarizada de como o algoritmo trabalha, e tem por objetivo consolidar como os componentes descritos neste trabalho são organizados e executados no código.

Considera-se que a base de dados já tenha sido escolhida e que seu pré-tratamento já tenha acontecido. Também já deve ter sido criada a matriz de correlações e estabelecido qual classificador será utilizado, já que o algoritmo somente trabalha com um classificador por vez.

A Figura 3.6a mostra as informações que são apresentadas ao usuário durante a execução do algoritmo. A linha destacada traz a informação de que o algoritmo foi iniciado considerando a base de dados *Wine* e o classificador KNN.

A Figura 3.6b indica a etapa em que o algoritmo faz o cálculo da acurácia considerando todas as variáveis originais da base de dados. Este valor é utilizado como referência interna do algoritmo para validar se a redução na dimensionalidade dos dados está produzindo melhores resultados do que o que já se tem com todas as variáveis.

Na etapa demonstrada pela Figura 3.7a é executado o Método Filter e realizada a primeira seleção de variáveis. Após esta etapa, é gerada uma nova base de dados composta somente com as variáveis remanescentes e são definidos os parâmetros que estabelecem as quantidades mínima e máxima de variáveis que cada formiga poderá percorrer.

A escolha da quantidade de variáveis de cada formiga é feita de forma individual e aleatória. Esta é uma das principais implementações de ajustes em relação ao AS_{rank}. Enquanto que no TSP as formigas tem que percorrer todas as cidades e retornar ao ponto de partida, no ACOFS_{rank} cada formiga percorre um número de variáveis diferente e não retorna ao ponto de partida. Essa característica estabelece parâmetros para a construção de subconjuntos com diferentes tamanhos e variáveis. Cada subconjunto é uma solução viável e todos serão analisados nas etapas seguintes do algoritmo.

Após as definições e seleções iniciais, o algoritmo entra então na etapa de execução do modelo ACO seguindo as equações e critérios estabelecidos na estrutura do ACOFS_{rank}. Em cada iteração acontecem as sub-etapas:

1. Atribuição do número de variáveis para cada formiga;
2. Associação de forma aleatória de cada formiga a uma variável;
3. Construção das soluções;
4. Cálculo das métricas estatísticas (acurácia, f1-score e aptidão) para cada solução;

Figura 3.6: Etapas iniciais da execução do algoritmo

(a) Apresentação de parâmetros.

```
Starting ACOFS_rank running over dataset "wine" with "KNN" classifier.


Ciclo ->  1  do Classificador:  KNN
Accuracy with all  13 features = 0.8056

Minimo de variáveis  :  3
Máximo de variáveis  :  9


Cicle:  1  - Round:  2  - Classifier:  KNN
==========
Iteraction  1 - Best OF  -> accuracy = 0.9722 with   6 features
Iteraction  2 - No gains with   8 features
Iteraction  3 - No gains with   7 features
Iteraction  4 - No gains with   3 features
Iteraction  5 - Best OF  -> accuracy = 0.9722 with   5 features
Iteraction  6 - No gains with   7 features
Iteraction  7 - No gains with   6 features
Iteraction  8 - No gains with   6 features
Iteraction  9 - Best OF  -> accuracy = 0.9722 with   3 features
Iteraction 10 - No gains with   8 features

  round num_features   features accuracy f1_score
0     1           13        All   0.8056   0.7781
1     2            3  [0, 7, 8]   0.9722   0.9762


Final result
============
Using KNN classifier
Best OF after 11 iterations in  2 rounds was 0.9722 with   3 features
Features reduction factor ...: 66.67 %
Accuracy Average ............:  0.889
Accuracy Standard Deviation .:  0.118

List of possible solutions :

['v_0', 'v_6', 'v_9']
```

(b) Cálculo de acurácia para referência.

```
Starting ACOFS_rank running over dataset "wine" with "KNN" classifier.

Ciclo ->  1  do Classificador:  KNN
Accuracy with all  13 features = 0.8056

Minimo de variáveis  :  3
Máximo de variáveis  :  9


Cicle:  1  - Round:  2  - Classifier:  KNN
==========
Iteraction  1 - Best OF  -> accuracy = 0.9722 with   6 features
Iteraction  2 - No gains with   8 features
Iteraction  3 - No gains with   7 features
Iteraction  4 - No gains with   3 features
Iteraction  5 - Best OF  -> accuracy = 0.9722 with   5 features
Iteraction  6 - No gains with   7 features
Iteraction  7 - No gains with   6 features
Iteraction  8 - No gains with   6 features
Iteraction  9 - Best OF  -> accuracy = 0.9722 with   3 features
Iteraction 10 - No gains with   8 features

  round num_features   features accuracy f1_score
0     1           13        All   0.8056   0.7781
1     2            3  [0, 7, 8]   0.9722   0.9762


Final result
============
Using KNN classifier
Best OF after 11 iterations in  2 rounds was 0.9722 with   3 features
Features reduction factor ...: 66.67 %
Accuracy Average ............:  0.889
Accuracy Standard Deviation .:  0.118

List of possible solutions :

['v_0', 'v_6', 'v_9']
```

Fonte: Compilação do autor

5. Seleção da melhor solução daquela iteração utilizando o maior valor da função de aptidão;

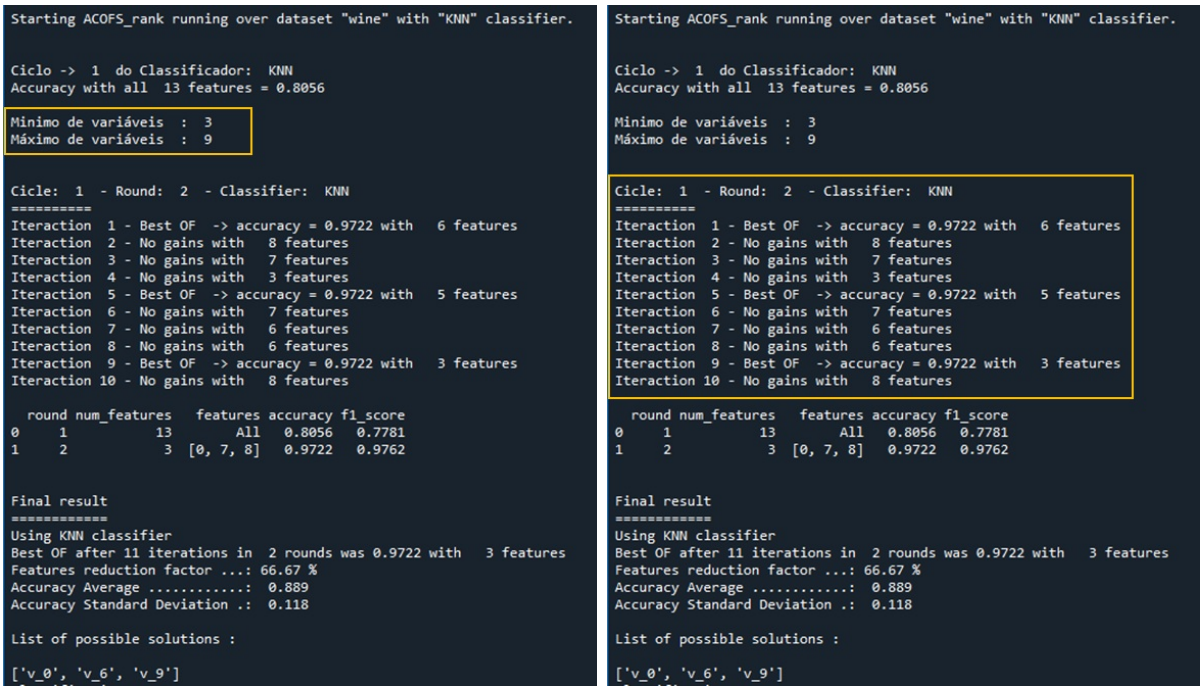
Ao final de cada iteração a matriz de feromônios é atualizada. O resultado de cada iteração é demonstrado na Figura 3.7b.

Ao final de um ciclo de iterações, a melhor solução dentre todas é armazenada e a matriz de feromônios é reinicializada com os valores originais. Esta etapa é mostrada na Figura 3.8a. Uma fez executados todos os ciclos, a solução com maior acurácia, menor número de variáveis e maior valor de f1-score é então selecionada como solução final do algoritmo. Esta etapa final está documentada na Figura 3.8b.

Figura 3.7: Etapas de seleção de variáveis

(a) Seleção pelo Método Filter.

(b) Execução do ACO e seleção pelo Méotodo Wrapper.

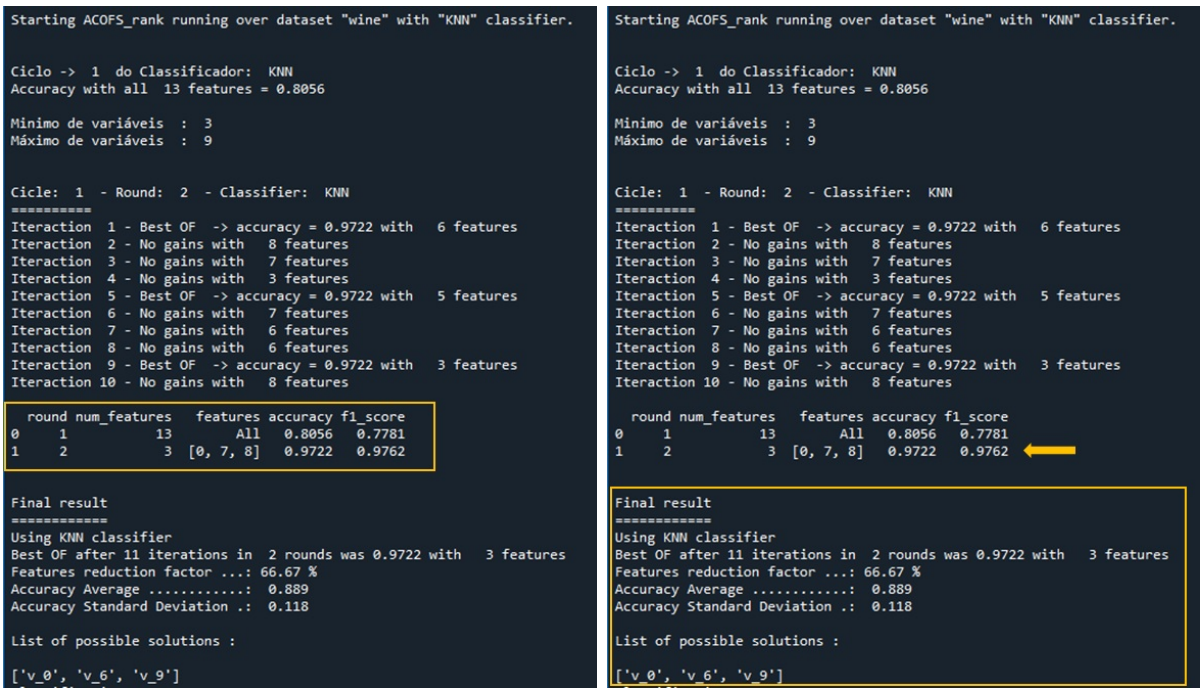


Fonte: Compilação do autor

Figura 3.8: Etapas de armazenamento e seleção final das melhores soluções.

(a) Armanzenamento das melhores soluções de cada ciclo.

(b) Escolha da solução final.



Fonte: Compilação do autor

Capítulo 4

Experimentos e Resultados

Para a avaliação do desempenho do ACOFS$_{rank}$ em relação ao WFACOFS, foram selecionadas as mesmas bases de dados utilizadas no WFACOFS e que ainda encontravam-se disponíveis. Estas bases são disponibilizadas através do repositório UCI (DUA e GRAFF, 2017) e são frequentemente citadas na literatura para avaliação de modelos de aprendizado de máquina. Este capítulo apresenta os resultados obtidos.

A Tabela 4.1 detalha as propriedades destas bases, que foram categorizadas de acordo com a quantidade de variáveis, seguindo o mesmo critério definido em Ghosh *et al.* (2019):

1. **Pequenos → Número de variáveis ≤ 10**
    - *Breast Cancer*
2. **Médios → 10 < número de variáveis ≤ 100**
    - *Wine*
    - *Ionosphere*
    - *Soybean Small*
    - *Hill Valley (without noise training)*
3. **Grandes → Número de variáveis > 100**
    - *Arrythmia*
    - *Madelon (training base)*

Bases de dados com elevado número de atributos (acima de 100) requerem uma análise de seu desempenho em relação ao tempo de processamento e geralmente são bases com características de dados densas (AYRES, 2021).

Tabela 4.1: Descrição das bases de dados utilizadas

Base de dados (Sigla)	**Variáveis**	**Classes**	**Amostras**
Breast Cancer (BC)	9	2	699
Wine (WI)	13	3	178
Ionosphere (IO)	34	2	351
Soybean Small (SS)	35	4	47
Hill Valley (without noise training) (HV)	100	2	606
Arrythmia (ARR)	279	16	452
Madelon (training base) (MN)	501	2	2.000

Fonte: Adaptado de Ghosh *et al.* (2019)

Mesmo nessas bases, foi realizado tratamento de dados inicial para preenchimento de campos sem informação e eliminação de variáveis com cardinalidade única (somente um valor para a variável em todas as instâncias) e com altas quantidades de valores nulos. Este processo visa garantir os requisitos iniciais básicos ao funcionamento do algoritmo. A partir da base tratada é, então, gerada a matriz de correlações.

Durante os experimentos iniciais, foram testadas matrizes geradas a partir de diferentes métodos: similaridade de cossenos, distância euclidiana e correlação de Spearman e pelo método phik - ϕ_K (BAAK *et al.*, 2020), não sendo identificadas variações representativas nos resultados do algoritmo ACOFS$_{rank}$. Seguindo a linha proposta para o trabalho, foi estabelecido o método phik - ϕ_K para a definição da matriz de correlações para todas as bases de dados.

O algoritmo ACOFS$_{rank}$ foi desenvolvido em linguagem Python, utilizando-se o IDE Spyder 4. Foi seguido modelo de codificação padrão não sendo utilizados recursos para uso de multiprocessamento ou processamento gráfico (GPU). Todos os experimentos computacionais foram executados em um computador Intel Core i5-7200U CPU @ 2.50GHz (2 núcleos de processamento) com 16GB de memória RAM e sistema operacional Windows 10 Home 64 bits.

Todos os hiperparâmetros utilizados no algoritmo são dados na Tabela 4.2. A calibração dos valores aplicáveis aos hiperparâmetros ρ, α, β, ϕ, w_1 e w_2 foi feita através da biblioteca IRACE (LÓPEZ-IBÁÑEZ *et al.*, 2016), que utiliza o pacote de *software* estatístico R (CORE TEAM, 2015). Para esta calibração foram consideradas somente as bases *Wine* e *Ionosphere*.

Tabela 4.2: Descrição dos valores dos parâmetros passados ao algoritmo

Parâmetro	**Descrição**	**Valor**
n	Número de formigas	Igual ao número de variáveis
m	Número de formigas elitistas	30% do número de variáveis limitado a 15
α	Influência do feromônio	2,0
β	Influência da correlação entre variáveis	1,0
Iterações	Número de ciclos de execução	10, 20 e 30
ρ	Fator de evaporação do feromônio	0,15
ϕ	Constante de balanço da acurácia	0,50
w_1	Parâmetro de peso para a acurácia	150
w_2	Parâmetro de peso para o número de variáveis do subconjunto selecionado	2

Fonte: Compilação do autor

A Figura 4.1 apresenta os valores utilizados pelo IRACE para definição do melhor conjunto de hiperparâmetros. As Figuras 4.2 e 4.3 contêm, respectivamente, as telas de início de execução com as configurações do IRACE e os resultados apresentados pela ferramenta como opções ótimas. Nesta última figura, também está indicada a seleção de hiperparâmetros adotada no ACOFS$_{rank}$.

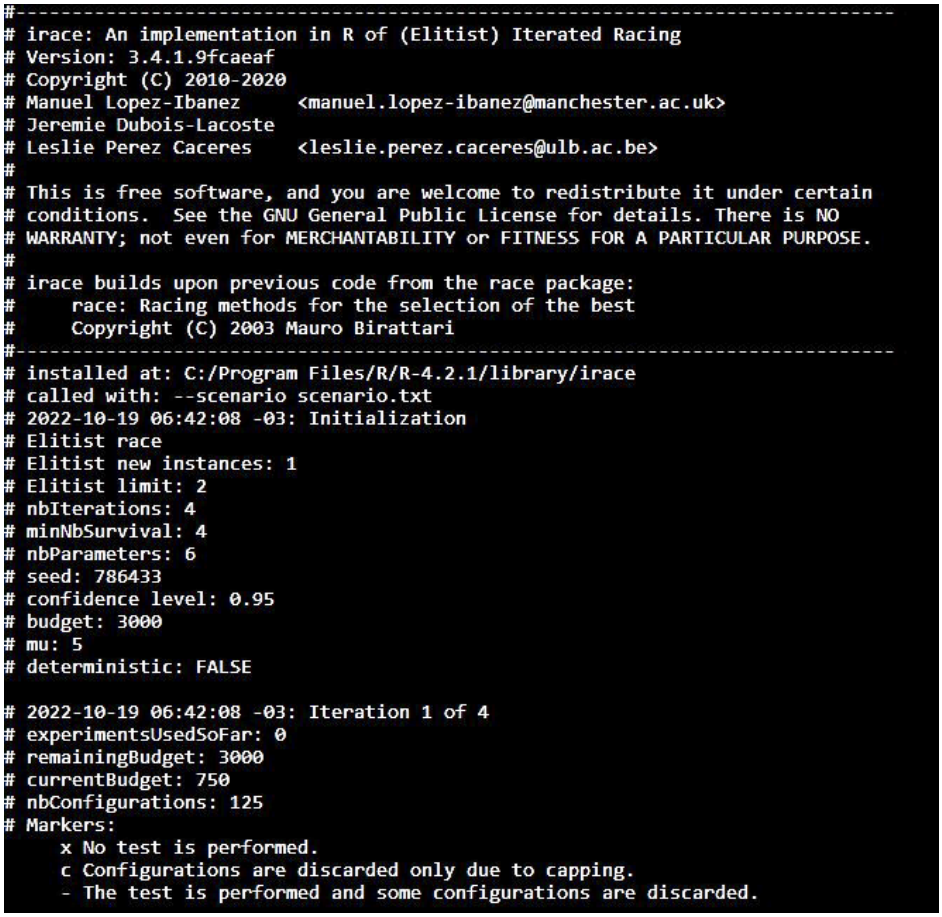
O algoritmo foi executado em todas as bases e com todos os classificadores em três blocos, com quantidades de 10, 20 e 30 iterações cada um. Para cada bloco de iterações (10, 20 e 30), foram realizados três ciclos com cinco rodadas de testes cada um para cada um dos quatro classificadores escolhidos. No total, foram realizados 3.600 testes.

Figura 4.1: Opções para valores de hiperparâmetros utilizadas pelo IRACE.

```
### Parameter file for MAIN
# name   switch      type    values      [conditions (using R syntax)]
rho      "--rho "    c       (0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35)
a        "--a "      c       (1.0, 2.0, 2.5, 3.0, 3.5)
b        "--b "      c       (0.5, 1.0, 2.0, 2.5, 3.0, 3.5, 4.0, 5.0, 8.0)
phi      "--phi "    c       (0.25, 0.50, 0.75, 0.85, 0.90, 0.95, 0.96, 0.97)
w1       "--w1 "     c       (100, 150, 175, 200)
w2       "--w2 "     c       (0.75, 1.0, 1.5, 1.75, 2.0, 2.5)
```

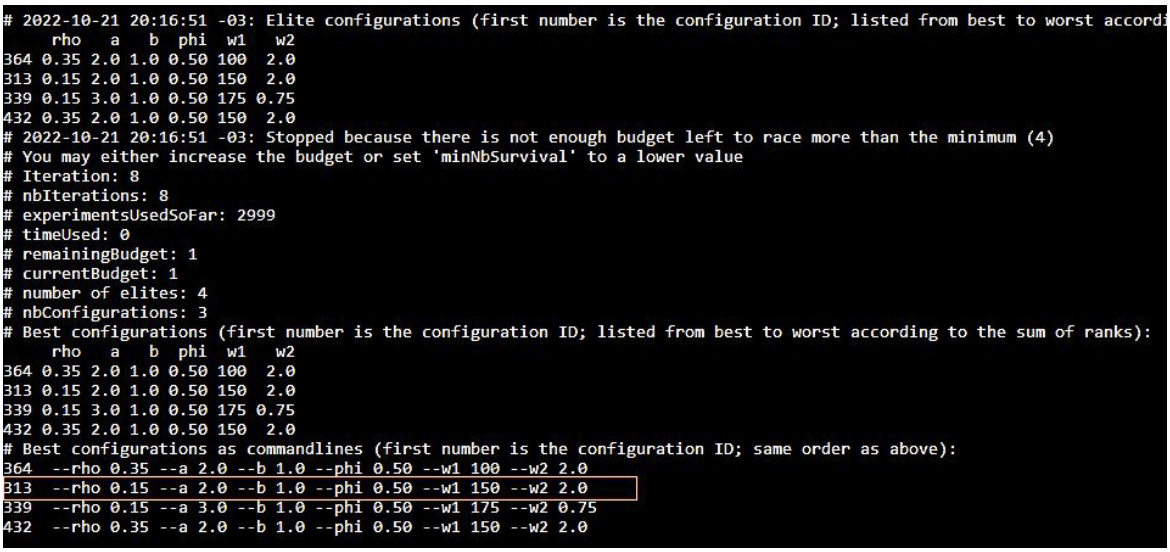
Fonte: Compilação do autor

Figura 4.2: Tela de execução inicial do IRACE com as configurações da ferramenta.



Fonte: Compilação do autor

Figura 4.3: Resultado do IRACE e conjunto de hiperparâmetros escolhidos.



Fonte: Compilação do autor

A Tabela 4.3 e a Figura 4.4 mostram os resultados obtidos nos experimentos com

configurações de 10, 20 e 30 iterações. Na Tabela 4.3 a quantidade de variáveis selecionada em cada caso está descrita entre parênteses e o melhor resultado para cada base de dados está em negrito.

O objetivo buscado é obter o maior valor de acurácia com o menor número de variáveis. No entanto, nem sempre este par perfeito de requisitos é alcançado. Neste trabalho, o valor da acurácia é considerado o fator de definição sobre qual seria o melhor resultado na comparação entre os valores alcançados pelo ACOFS$_{rank}$ e o WFACOFS, ficando o número de variáveis em segundo plano.

A Tabela 4.4 apresenta os percentuais de variáveis selecionadas pelo ACOFS$_{rank}$ para cada classificador e em cada base de dados, em relação aos respectivos números totais de variáveis. Os valores obtidos pelo ACOFS$_{rank}$ são comparados com os valores definidos no WFACOFS.

Nessa tabela, quanto menor o percentual, melhor, pois indica que menos variáveis foram escolhidas para compor as soluções. É possível observar que o algoritmo ACOFS$_{rank}$ atinge maior redução na dimensionalidade na maioria das bases de dados estudadas.

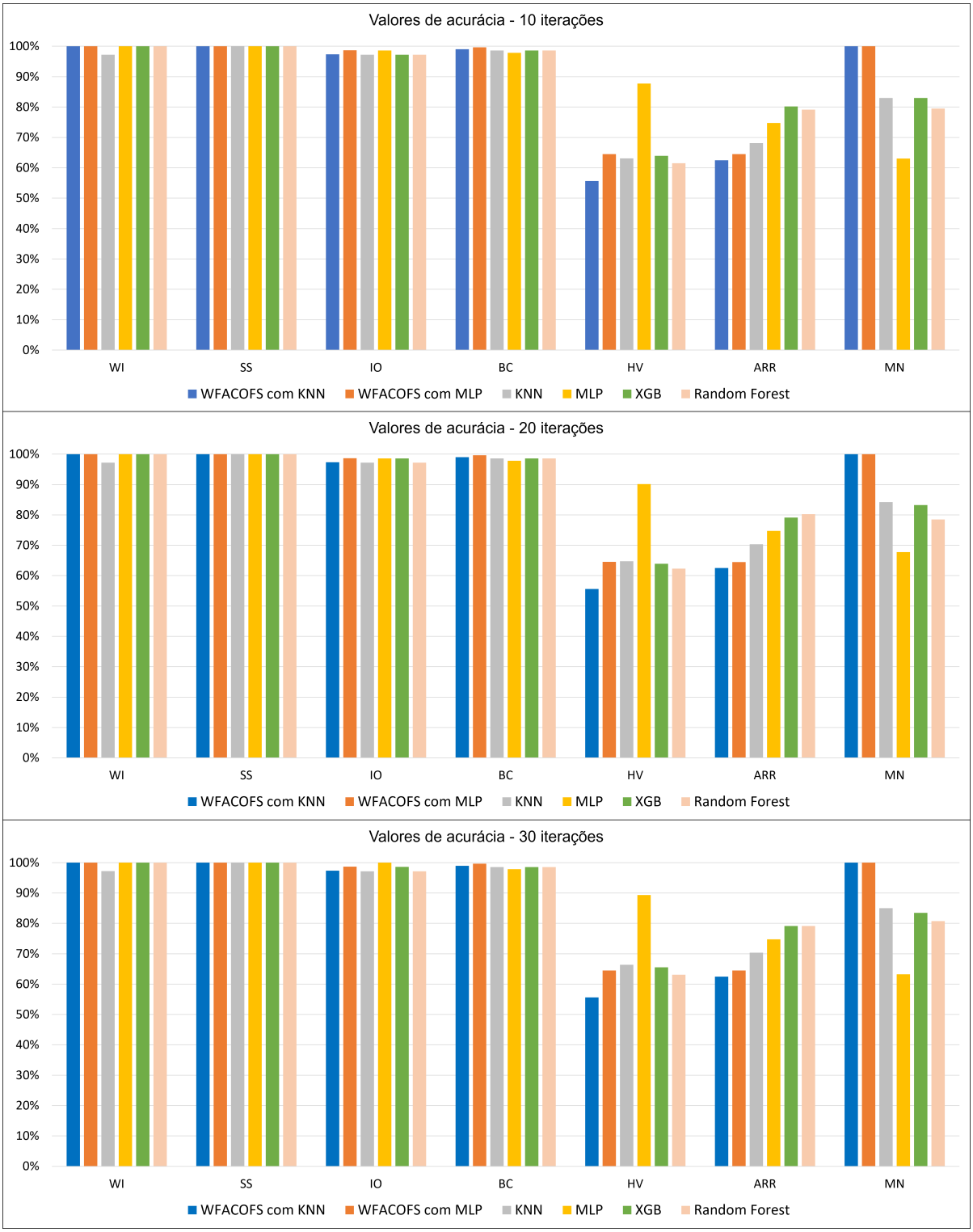
Como descrito no Capítulo de Materiais e Métodos, em cada rodada de testes o algoritmo seleciona o melhor resultado considerando valores de acurácia, *fitness* e número de variáveis. A Tabela 4.5 apresenta os valores de média e desvio padrão para estes melhores resultados.

Tabela 4.3: Valores finais de acurácia por classificador.

Bases de dados	Bases / Caracteristicas	Wine	Soybean Small	Ionosphere	Breast Cancer	Hill Valley	Arrhythmia	Madelon
	Sigla	WI	SS	IO	BC	HV	ARR	MN
	Variáveis / instancias	13 / 178	35 / 47	34 / 351	9 / 699	100 / 606	279 / 452	501 / 2000
WFACOFS com KNN	Acurácia % (Número variáveis)	100% (6)	100% (6)	97,35% (7)	99,00% (5)	55,61% (63)	62,50% (9)	100% (286)
WFACOFS com MLP	Acurácia % (Número variáveis)	100% (4)	100% (4)	98,68% (12)	99,67% (7)	64,52% (66)	64,47% (17)	100% (288)
10 iterações								
ACOFS_rank	KNN	97,22% (3)	100,00% (3)	97,18% (3)	98,57% (9)	63,11% (8)	68,13% (124)	**83,00% (37)**
	MLP	100,00% (7)	100,00% (3)	**98,59% (5)**	97,86% (4)	**87,70% (67)**	74,73% (97)	63,00% (5)
	XGB	**100,00% (3)**	**100,00% (3)**	97,18% (11)	98,57% (5)	63,93% (30)	**80,22% (109)**	83,00% (175)
	Random Forest	100,00% (3)	100,00% (3)	97,18% (6)	**98,57% (3)**	61,48% (8)	79,12% (45)	79,50% (38)
20 iterações								
ACOFS_rank	KNN	97,22% (3)	100,00% (3)	97,18% (3)	98,57% (9)	64,75% (10)	70,33% (14)	**84,25% (69)**
	MLP	100,00% (6)	100,00% (3)	**98,59% (4)**	97,86% (4)	**90,16% (68)**	74,73% (137)	67,75% (9)
	XGB	**100,00% (3)**	**100,00% (3)**	98,59% (8)	98,57% (5)	63,93% (11)	79,12% (75)	83,25% (27)
	Random Forest	100,00% (3)	100,00% (3)	97,18% (5)	**98,57% (3)**	62,30% (4)	**80,22% (34)**	78,50% (27)
30 iterações								
ACOFS_rank	KNN	97,22% (3)	100,00% (3)	97,18% (3)	98,57% (9)	66,39% (16)	70,33% (35)	**85,00% (38)**
	MLP	100,00% (6)	100,00% (3)	**100,00% (8)**	97,86% (4)	**89,34% (23)**	74,73% (32)	63,25% (3)
	XGB	**100,00% (3)**	**100,00% (3)**	98,59% (8)	98,57% (5)	65,57% (18)	79,12% (72)	83,50% (142)
	Random Forest	100,00% (3)	100,00% (3)	97,18% (5)	**98,57% (3)**	63,11% (19)	**79,12% (34)**	80,75% (17)

Fonte: Compilação do autor

Figura 4.4: Gráficos de acurácia por classificador.



Fonte: Compilação do autor

Tabela 4.4: Percentuais de variáveis selecionadas em relação ao total de variáveis de cada base de dados.

	WI	SS	IO	BC	HV	ARR	MN
WFACOFS com KNN	46%	17%	21%	56%	63%	3%	57%
WFACOFS com MLP	31%	11%	35%	78%	66%	6%	57%
10 iterações							
KNN	23%	9%	9%	100%	8%	44%	7%
MLP	54%	9%	15%	44%	67%	35%	1%
XGB	23%	9%	32%	56%	30%	39%	35%
Random Forest	23%	9%	18%	33%	8%	16%	8%
20 iterações							
KNN	23%	9%	9%	100%	10%	5%	14%
MLP	46%	9%	12%	44%	68%	49%	2%
XGB	23%	9%	24%	56%	11%	27%	5%
Random Forest	23%	9%	15%	33%	4%	12%	5%
30 iterações							
KNN	23%	9%	9%	100%	16%	13%	8%
MLP	46%	9%	24%	44%	23%	11%	1%
XGB	23%	9%	24%	56%	18%	26%	28%
Random Forest	23%	9%	15%	33%	19%	12%	3%

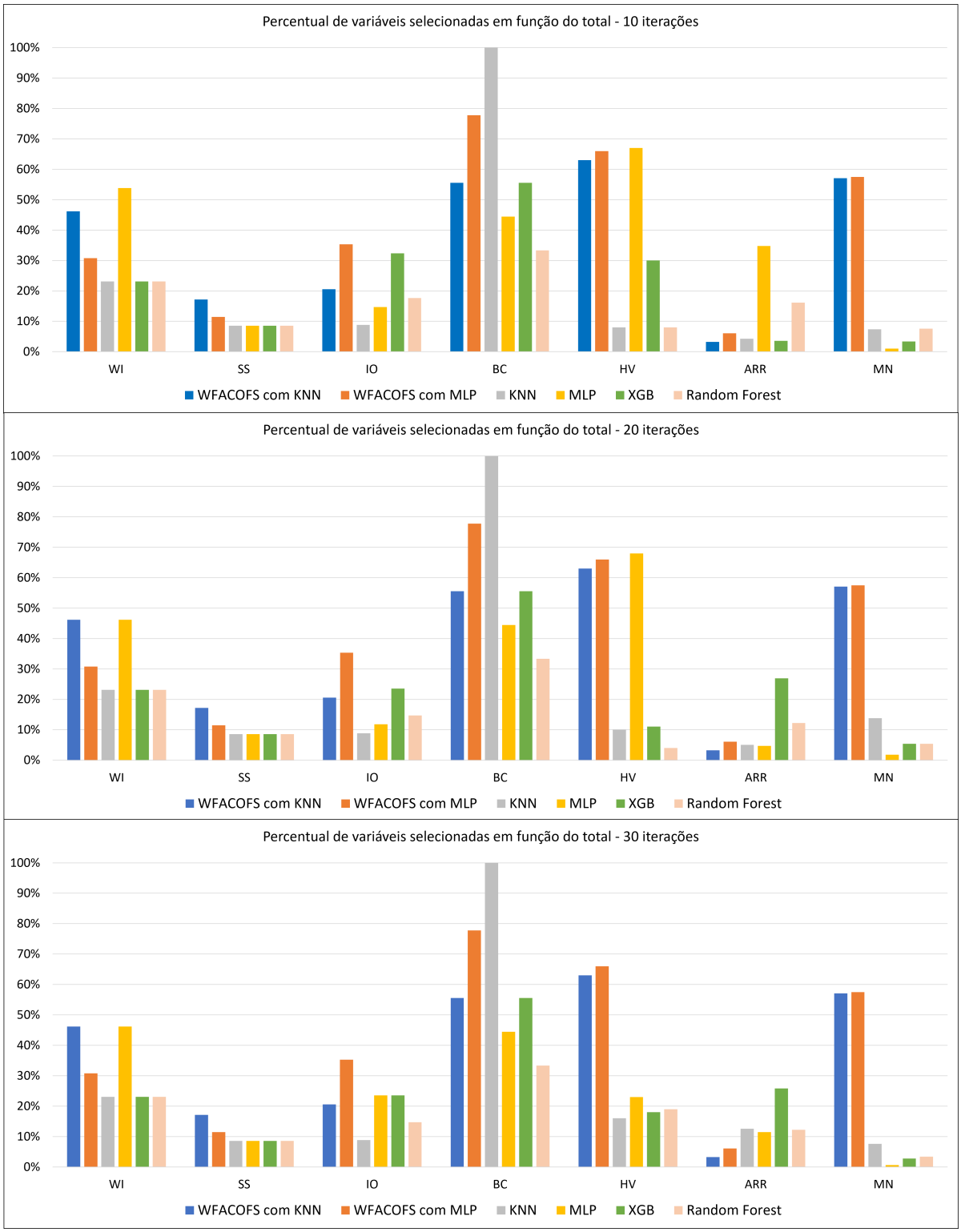
Fonte: Compilação do autor

Tabela 4.5: Média e Desvio Padrão dos melhores valores de acurácia obtidos em cada rodada de testes.

	WI		SS		IO		BC		HV		ARR		MN	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
	10 iterações													
KNN	94,44%	6,80%	98,33%	4,08%	93,66%	4,60%	97,98%	0,29%	60,38%	4,33%	65,38%	3,59%	79,54%	4,76%
MLP	98,61%	1,52%	100,00%	0,00%	96,48%	2,48%	96,91%	2,34%	78,69%	12,62%	71,25%	3,90%	58,92%	4,88%
XGB	99,54%	1,13%	96,67%	8,16%	94,84%	1,92%	97,02%	1,89%	60,93%	2,57%	76,92%	2,95%	81,79%	1,16%
Random Forest	100,00%	0,00%	100,00%	0,00%	95,30%	1,45%	97,62%	0,86%	58,61%	2,58%	76,37%	3,59%	75,62%	3,27%
	20 iterações													
KNN	94,44%	6,80%	98,33%	4,08%	93,19%	4,49%	97,98%	0,29%	61,61%	4,95%	65,93%	4,05%	80,71%	5,29%
MLP	98,61%	1,52%	100,00%	0,00%	96,71%	2,62%	96,91%	2,34%	80,88%	13,68%	71,61%	3,96%	59,63%	5,99%
XGB	99,54%	1,13%	96,67%	8,16%	96,01%	2,42%	96,66%	1,67%	61,88%	2,68%	76,92%	2,78%	82,08%	1,21%
Random Forest	100,00%	0,00%	100,00%	0,00%	95,07%	1,47%	97,74%	0,83%	59,98%	3,09%	77,29%	4,09%	75,75%	3,09%
	30 iterações													
KNN	94,44%	6,80%	98,33%	4,08%	93,66%	4,60%	97,98%	0,29%	62,56%	5,43%	66,12%	4,08%	81,71%	5,80%
MLP	98,15%	1,44%	100,00%	0,00%	97,42%	3,01%	96,91%	2,34%	80,74%	13,56%	71,80%	4,03%	59,62%	4,49%
XGB	99,54%	1,13%	96,67%	8,16%	96,24%	2,47%	97,14%	1,86%	62,57%	3,14%	77,29%	2,92%	82,62%	1,46%
Random Forest	100,00%	0,00%	100,00%	0,00%	95,77%	1,54%	98,21%	0,87%	60,52%	3,25%	77,10%	3,90%	77,00%	3,85%

Fonte: Compilação do autor

Figura 4.5: Gráficos do percentual de variáveis selecionadas em relação ao total de variáveis de cada base de dados.



Fonte: Compilação do autor

A Tabela 4.6 mostra uma comparação entre os melhores valores de acurácia obtidos pelos algoritmos em cada base de dados, independentemente do classificador ou número de iterações. O número de variáveis selecionadas em cada caso é descrito entre parênteses, e o melhor resultado para cada base de dados está em negrito e sublinhado, sempre considerando a acurácia como fator de decisão.

Tabela 4.6: Comparação entre os melhores resultados de cada algoritmo.

	WI	SS	IO	BC	HV	ARR	MN
Melhores resultados							
WFACOFS	100,00% (6)	100,00% (6)	98,68% (7)	**99,67% (5)**	64,52% (63)	64,47% (9)	**100,00% (286)**
ACOFSrank	**100,00% (3)**	**100,00% (3)**	**100,00% (8)**	98,57% (3)	**90,16% (68)**	**80,22% (34)**	85,00% (38)

Fonte: Compilação do autor

Através dos valores obtidos nos experimentos e apresentados nas Tabelas 4.3, 4.4 e, principalmente, na Tabela 4.6, pode-se observar que o algoritmo ACOFS$_{rank}$ proposto é superior ao WFACOFS na maioria das bases analisadas. O ACOFS$_{rank}$ perdeu em desempenho somente nas bases *Breast Cancer* (BC) e *Madelon* (MN).

No primeiro caso, a diferença nos valores de acurácia foi somente de 0,1%, com o ACOFS$_{rank}$ selecionando um número de variáveis inferior àquele obtido pelo WFACOFS. O ponto de atenção está na base *Madelon* (MN) na qual o ACOFS$_{rank}$ perde em todos os casos, mas ainda assim atinge valores de acurácia relevantes com números de variáveis significativamente menor do que os obtidos pelo WFACOFS.

Capítulo 5

Considerações Finais

5.1. Conclusões

O presente capítulo tem por objetivo documentar os comentários de conclusão deste trabalho, onde serão revistos os objetivos da pesquisa frente aos resultados alcançados.

A parte prática e essencial deste trabalho é um algoritmo desenvolvido em linguagem Python que permite a execução do ACOFS$_{rank}$ em bases de dados variadas e com opção de seleção de classificador para melhor adequação às características da base de dados sob análise.

Pode-se considerar que o objetivo geral de criar um algoritmo aprimorado utilizando como base o algoritmo *Rank-based Ant System* (AS$_{rank}$) foi alcançado, levando em consideração que os resultados obtidos pelo novo algoritmo ACOFS$_{rank}$ proposto superaram em grande parte das bases de análise àqueles auferidos pelo modelo de referência WFACOFS.

Mesmo na base Madelon (MN) onde o WFACOFS alcançou melhor desempenho na acurácia geral, ainda assim o ACOFS$_{rank}$ se saiu melhor na seleção de um conjunto de variáveis significativamente menor, com uma acurácia bastante representativa.

Desta forma, também foi alcançada a proposta de pesquisa de buscar um algoritmo que apresentasse desempenho similar ao WFACOFS, utilizado como referência e modelo de comparação.

A escolha da métrica estatística de correlação para a construção da "matriz de distâncias" utilizada no ACOFS$_{rank}$ e, em especial, o uso do novíssimo método phik (ϕ_K) para obtenção de valores de correlação entre tipos variados de dados cobre o objetivo específico de revisão da literatura sobre métodos estatísticos para definição de correlação entre variáveis de diferentes tipos.

O método phik (ϕ_K) foi identificado durante as pesquisas para este trabalho e mostrou-se muito eficaz em sua proposta, contribuindo de forma significativa na redução da codificação e na complexidade das análises para obtenção da matriz de correlação utilizada no ACOFS$_{rank}$.

Também foi alcançado o objetivo específico de implantação de um modelo atualizado utilizando o ACO e adaptado ao uso como ferramenta auxiliar nos processos de seleção de variáveis em projetos de *data analytics* e aprendizado de máquina (*machine learning*).

Assim, fica demonstrado que é possível estabelecer melhorias com efetiva melhora no desempenho em algoritmos baseados no ACO amplamente conhecidos e utilizados.

5.2. Trabalhos Futuros

Para trabalhos futuros é indicada a adequação do algoritmo ACOFS$_{rank}$ para modelos de regressão. Isto exigirá ajustes no código, pois diferentemente dos modelos de classificação que objetivam maximização de resultados, os modelos de regressão buscam a minimização. Além disso, deve ser estabelecida uma nova métrica aplicável em regressão tendo em vista que a métrica de acurácia é válida somente para modelos de classificação.

Outra oportunidade de estudos seria a avaliação de outros classificadores como SVM, Gaussian e até mesmo Regressão Logística na estrutura do método *Wrapper* que podem se mostrar mais adequados a determinadas estruturas de dados.

É recomendado também o uso do algoritmo em uma gama maior de bases de dados, em especial, bases do mundo real, visando avaliar se os classificadores adotados no método *Wrapper* são suficientes e adequados e até mesmo se o algoritmo em si mantém o mesmo desempenho alcançado nas bases de estudo.

Adicionalmente, é indicado que o código do algoritmo seja ajustado e reestruturado para multiprocessamento e uso de unidades de processamento gráfico (GPU). Isso certamente trará maior rapidez na execução do algoritmo, principalmente em bases com maior número de variáveis sob análise.

É de grande relevância o estudo de métodos eficientes e automatizados de seleção de variáveis, considerando que a cada dia, a coleta de dados se torna mais fácil e as bases maiores, sem necessariamente que todas as variáveis coletadas sejam importantes para a construção de modelos de aprendizado de máquina. Realizar esta seleção de forma eficiente torna-se fundamental para viabilizar estes modelos.

Referências Bibliográficas

ABDI, H. "The Kendall rank correlation coefficient", *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA*, pp. 508–510, 2007.

AGHDAM, M. H., GHASEM-AGHAEE, N., BASIRI, M. E. "Text feature selection using ant colony optimization", *Expert systems with applications*, v. 36, n. 3, pp. 6843–6853, 2009.

AYRES, P. F. *Seleção de atributos baseado no algoritmo de otimização por colônia de formigas para processos mineradores*. 2021, 64f., Dissertação, (Mestrado Profissional em Instrumentação, Controle e Automação de Processos de Mineração), UFOP - Universidade Federal de Ouro Preto, Ouro Preto, 2021.

AZANK, F. "Paradoxo da Acurácia". `https://medium.com/turing-talks/paradoxo-da-acur%C3%A1cia-56baa75334f2`, 2020. (acessado em 20 de setembro 2022).

BAAK, M., KOOPMAN, R., SNOEK, H., et al.. "phik - PyPi". `https://pypi.org/project/phik/`, 2018. (acessado em 03 de setembro 2022).

BAAK, M., KOOPMAN, R., SNOEK, H., et al.. "A new correlation coefficient between categorical, ordinal and interval variables with Pearson characteristics", *Computational Statistics & Data Analysis*, v. 152, pp. 107043, 2020.

BLUM, C., ROLI, A., DORIGO, M. "HC–ACO: The hyper-cube framework for Ant Colony Optimization". Em: *Proceedings of MIC*, v. 2, pp. 399–403, 2001.

BROWNLEE, J. "How to Calculate Feature Importance With Python". `https://machinelearningmastery.com/calculate-feature-importance-with-python/`, 2020a. (acessado em 2 de março de 2022).

BROWNLEE, J. "How to Perform Feature Selection with Categorical Data". `https://machinelearningmastery.com/feature-selection-with-categorical-data/#:~:text=The%20two%`

20most%20commonly%20used,and%20the%20mutual%20information%20statistic, 2019a. (acessado em 20 de junho 2022).

BROWNLEE, J. "Feature Selection to Improve Accuracy and Decrease Training Time". https://machinelearningmastery.com/feature-selection-to-improve-accuracy-and-decrease-training-time/, 2014. (accessado em 30 de setembro 2022).

BROWNLEE, J. "How to Choose a Feature Selection Method For Machine Learning". https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/, 2019b. (acessado em 03 de setembro 2022).

BROWNLEE, J. "Ordinal and One-Hot Encodings for Categorical Data". https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/, 6 2020b. (acessado em 22 de outubro de 2021).

BRUCE, P., BRUCE, A., GEDECK, P. *Practical statistics for data scientists: 50+ essential concepts using R and Python*. O'Reilly Media, 2020.

BSA THE SOFTWARE ALLIANCE, B. "What's the Big Deal With Data?" https://data.bsa.org/wp-content/uploads/2015/12/bsadatastudy_en.pdf, 10 2015. (acessado em 04 de Setembro de 2021).

BULLNHEIMER, B., HARTL, R. F., STRAUSS, C. "A new rank based version of the Ant System. A computational study." 1997.

CALKINS, K. G. "Applied Statistics - Lesson 13". https://www.andrews.edu/~calkins/math/edrm611/edrm13.htm#POINTB, 2005. (acessado em 31 de outubro de 2021).

CHAPMAN, P., CLINTON, J., KERBER, R., et al.. "The CRISP-DM user guide". Em: *4th CRISP-DM SIG Workshop in Brussels in March*, v. 1999. sn, 1999.

CHEN, T., GUESTRIN, C. "Xgboost: A scalable tree boosting system". Em: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

CHEN, Y.-W., LIN, C.-J. "Combining SVMs with various feature selection strategies". Em: *Feature extraction*, Springer, pp. 315–324, 2006.

COLLINS, E. D. "Significado de Re-scaling". https://www.collinsdictionary.com/pt/dictionary/english/rescale, 2021. (acessado em 01 de Março de 2022).

COLOSIMO, E. A. "Estatística II - Correlação e Regressão Linear Simples". http://www.est.ufmg.br/~enricoc/pdf/EstatisticaII/aula9-10_corr-reg.pdf. (acessado em 01 de Março de 2022).

CORDÓN, O., DE VIANA, I. F., HERRERA, F. "Analysis of the best-worst ant system and its variants on the TSP." *Mathware and Soft Computing*, v. 9, n. 2-3, pp. 177–192, 2002.

CORE TEAM, R. "R: A Language and Environment for Statistical Computing". https://www.R-project.org, 2015. (acessado em 20 de Setembro de 2022).

DBPEDIA. "Similaridade por cossenos". https://dbpedia.org/page/Cosine_similarity, 2020. (acessado em 01 de Março de 2022).

DELAMORA, R. A. "Master Thesis Improved ACO for selecting features". https://github.com/RobertoDelamora/Master_Thesis_Improved_ACO_for_selecting_features, 2023. (criado em 02 de março de 2023).

DICIO, D. O. D. P. "Significado de Correlacao". https://www.dicio.com.br/correlacao/, 2021. (acessado em 29 de Dezembro de 2021).

DORIGO, M. "Positive feedback as a search strategy", *Technical report*, pp. 91–16, 1991.

DORIGO, M. "Optimization, learning and natural algorithms", *Ph. D. Thesis, Politecnico di Milano*, 1992.

DORIGO, M., STÜTZLE, T. "Ant colony optimization: overview and recent advances", *Handbook of metaheuristics*, pp. 311–351, 2019.

DORIGO, M., MANIEZZO, V., COLORNI, A. "Ant system: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 26, n. 1, pp. 29–41, 1996.

DUA, D., GRAFF, C. "UCI Machine Learning Repository". 2017. Disponível em: <http://archive.ics.uci.edu/ml>.

EDELMANN, D., MÓRI, T. F., SZÉKELY, G. J. "On relationships between the Pearson and the distance correlation coefficients", *Statistics & Probability Letters*, v. 169, pp. 108960, 2021.

FERREIRA, C. A. "MLP Classifier". https://medium.com/@carlosalbertoff/mlp-classifier-526978d1c638, 2019. (acessado em 2 de março de 2022).

FORSATI, R., MOAYEDIKIA, A., JENSEN, R., et al.. "Enriched ant colony optimization and its application in feature selection", *Neurocomputing*, v. 142,

pp. 354–371, 2014.

FORTHOFER R.N., L. R. *Rank Correlation Methods*. Public Program Analysis. Springer, Boston, MA, 1981.

GAJAWADA, S. K. "ANOVA for Feature Selection in Machine Learning". https://towardsdatascience.com/anova-for-feature-selection-in-machine-learning-d9305e228476, 2019. (acessado em 03 de setembro 2022).

GAMBARDELLA, L. M., DORIGO, M. "Ant-Q: A reinforcement learning approach to the traveling salesman problem". Em: *Machine learning proceedings 1995*, Elsevier, pp. 252–260, 1995.

GAMBARDELLA, L. M., DORIGO, M. "Solving symmetric and asymmetric TSPs by ant colonies", pp. 622–627, 1996.

GASPAR-CUNHA, A., TAKAHASHI, R., ANTUNES, C. H. *Manual de computação evolutiva e metaheurística*. Imprensa da Universidade de Coimbra/Coimbra University Press, 2012.

GHOSH, M., GUHA, R., RAM, S., et al.. "A wrapper-filter feature selection technique based on ant colony optimization", *Neural Computing & Applications*, v. 32, n. 12, pp. 7839–7857, 2019.

GLEN, S. "Phi Coefficient (Mean Square Contingency Coefficient)". https://www.statisticshowto.com/phi-coefficient-mean-square-contingency-coefficient/, 6 2016. (acessado em 30 de Dezembro de 2021).

GOLDSCHMIDT, R., PASSOS, E., BEZERRA, E. *Data mining*. Elsevier Brasil, 2015.

GOMES, F. "Tipos de correlações". https://gomesfellipe.github.io/post/2017-12-01-tipos-de-correlacoes/, 2017. (acessado em 03 de setembro 2022).

GOODMAN, L. A., KRUSKAL, W. H. "Measures of association for cross classifications", *Measures of association for cross classifications*, pp. 2–34, 1979.

GUILFORD, J. P. "The phi coefficient and chi square as indices of item validity", *Psychometrika*, v. 6, n. 1, pp. 11–19, 1941. ISSN: 0033-3123.

HO, T. K. "Random decision forests". Em: *Proceedings of 3rd international conference on document analysis and recognition*, v. 1, pp. 278–282. IEEE, 1995.

KABIR, M. M., SHAHJAHAN, M., MURASE, K. "An efficient feature selection using

ant colony optimization algorithm". Em: *Neural Information Processing: 16th International Conference, ICONIP 2009, Bangkok, Thailand, December 1-5, 2009, Proceedings, Part II 16*, pp. 242–252. Springer, 2009.

KUHN, M., JOHNSON, K. *Applied predictive modeling*, v. 26. Springer, 2013.

LÓPEZ-IBÁÑEZ, M., DUBOIS-LACOSTE, J., CÁCERES, L. P., et al.. "The irace package: Iterated racing for automatic algorithm configuration", *Operations Research Perspectives*, v. 3, pp. 43–58, 2016.

MAIA, A. G. *Econometria: conceitos e aplicações*. Saint Paul Editora, 2019.

MANIEZZO, V., CARBONARO, A. "An ANTS heuristic for the frequency assignment problem", *Future Generation Computer Systems*, v. 16, n. 8, pp. 927–935, 2000.

MICHAELIS DICIONÁRIO ONLINE, M. "Definição de vies". `https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/vies/`. (acessado em 04 de Setembro de 2021).

MILONE, G. *Estatística: geral e aplicada*. Pioneira Thomson Learning, 2004.

MOHANTY, A. "Multi layer Perceptron (MLP) Models on Real World Banking Data". `https://becominghuman.ai/multi-layer-perceptron-mlp-models-on-real-world-banking-data-f6dd3d7e998f`, 2019. (acessado em 2 de março de 2022).

MOREIRA, S. "Multi layer Perceptron (MLP) Models on Real World Banking Data". `https://medium.com/ensina-ai/rede-neural-perceptron-multicamadas-f9de8471f1a9#:~:text=Perceptron%20Multicamadas%20(PMC%20ou%20MLP,sa%C3%ADda%20desejada%20nas%20camadas%20intermedi%C3%A1rias.`, 2018. (acessado em 2 de março de 2022).

MORETTIN, P. A., BUSSAB, W. O. *Estatística básica*. Saraiva Educação SA, 2017.

PEARSON, K. *On the theory of contingency and its relation to association and normal correlation*. Dulau and Company, 1904.

PORTO, E. "outlier no Dicionário Infopédia da Língua Portuguesa". `https://www.infopedia.pt/dicionarios/lingua-portuguesa/outlier`, 2020a. (acessado em 30 de outubro de 2021).

PORTO, E. "qui-quadrado no Dicionário Infopédia da Lingua Portuguesa)". `https://www.infopedia.pt/dicionarios/lingua-portuguesa/qui-quadrado`, 2020b. (acessado em 03 de Janeiro de 2022).

SANTOS, G. "Estatística para Seleção de Atributos". https://medium.com/data-hackers/estat%C3%ADstica-para-sele%C3%A7%C3%A3o-de-atributos-81bdc274dd2c, 2021. (acessado em 03 de setembro 2022).

SHARDA, R., DELEN, D., TURBAN, E. *Business Intelligence e Análise de Dados para Gestão do Negócio-4*. Bookman Editora, 2019.

SILVA, T. A. "Como implementar as métricas Precisão, Revocação, Acurácia e Medida-F". https://tiago.blog.br/precisao-revocacao-acuracia-e-medida-f/#:~:text=Medida%20F%20(F%20Measure%2C%20F1,medida%20de%20confiabilidade%20da%20acur%C3%A1cia., 2018. (acessado em 20 de setembro 2022).

SOLORIO-FERNÁNDEZ, S., CARRASCO-OCHOA, J. A., MARTÍNEZ-TRINIDAD, J. F. "A review of unsupervised feature selection methods", *Artificial Intelligence Review*, v. 53, n. 2, pp. 907–948, 2020.

SPEARMAN, C. "The Proof and Measurement of Association between Two Things." *Amer. Journal of Psychology*, v. 15, n. 1, pp. 72–101, 01 1904.

STÜTZLE, T., DORIGO, M. "ACO algorithms for the traveling salesman problem", *Evolutionary algorithms in engineering and computer science*, v. 4, pp. 163–183, 1999.

STÜTZLE, T., HOOS, H. H. "Improving the Ant System: A detailed report on the MAX–MIN Ant System", *FG Intellektik, FB Informatik, TU Darmstadt, Germany, Tech. Rep. AIDA–96–12*, 1996.

TABAKHI, S., MORADI, P., AKHLAGHIAN, F. "An unsupervised feature selection algorithm based on ant colony optimization", *Engineering Applications of Artificial Intelligence*, v. 32, pp. 112–123, 2014.

ZHANG, Z., FENG, Z. "Two-stage updating pheromone for invariant ant colony optimization algorithm", *Expert Systems with Applications*, v. 39, n. 1, pp. 706–712, 2012.