STANLEY PHILLIPE MACHADO SILVA

APLICAÇÃO DE TÉCNICAS DE CLUSTERIZAÇÃO DE DADOS PARA ANÁLISE DE DESEMPENHO DE ALUNOS DE DISCIPLINAS COM FOCO NO ENSINO DE TÉCNICAS DE PROGRAMAÇÃO.

LAVRAS – MG

2023

STANLEY PHILLIPE MACHADO SILVA

APLICAÇÃO DE TÉCNICAS DE CLUSTERIZAÇÃO DE DADOS PARA ANÁLISE DE DESEMPENHO DE ALUNOS DE DISCIPLINAS COM FOCO NO ENSINO DE TÉCNICAS DE PROGRAMAÇÃO.

Trabalho de Conclusão de Curso apresentado ao Instituto de Ciências Exatas e Tecnológicas da Universidade Federal de Lavras, como parte das exigências para obtenção do título de bacharel em Sistemas de Informação.

Prof. Dr. Joaquim Quinteiro Uchôa
Orientador

Prof. Dr. Renato Ramos da Silva
Coorientador

LAVRAS – MG

2023

STANLEY PHILLIPE MACHADO SILVA

APLICAÇÃO DE TÉCNICAS DE CLUSTERIZAÇÃO DE DADOS PARA ANÁLISE DE DESEMPENHO DE ALUNOS DE DISCIPLINAS COM FOCO NO ENSINO DE TÉCNICAS DE PROGRAMAÇÃO.
APPLICATION OF DATA CLUSTERING TECHNIQUES FOR THE PERFORMANCE ANALYSIS OF STUDENTS IN COURSES FOCUSED ON TEACHING PROGRAMMING TECHNIQUES.

Trabalho de Conclusão de Curso apresentado ao Instituto de Ciências Exatas e Tecnológicas da Universidade Federal de Lavras, como parte das exigências para obtenção do título de bacharel em Sistemas de Informação.

APROVADA em 30 de Novembro de 2023.

Prof. Dr. Joaquim Quinteiro Uchôa	DAC ICET
Prof. Dr. Renato Ramos Da Silva	DAC ICET
Profa. Dra. Juliana Galvani Greghi	DAC ICET
Esp. Lívia Rosa Souza	INMETRICS

Prof. Dr. Joaquim Quinteiro Uchôa
Orientador

Prof. Dr. Renato Ramos da Silva
Co-Orientador

LAVRAS – MG
2023

Aos que não desistem.

AGRADECIMENTOS

Primeiramente, expresso minha profunda gratidão aos meus pais, cujo apoio tem sido fundamental em toda a minha jornada. O apoio e a confiança deles foram essenciais. Ao meu irmão, meu fiel companheiro de jornada, por ter me acompanhado e ajudado - "Tamo junto!". Aos meus tios, que sempre me incentivaram a nunca desistir. Agradeço a Andressa, minha amada namorada, por ser o maior encanto de meus pensamentos. Aos amigos, tanto aos que reencontrei quanto aos novos que conquistei ao longo dessa jornada, expresso minha gratidão. Agradeço também aos meus professores, que com extrema paciência, guiaram esta criatura pela trilha de aprendizado, aprovações e reprovações. Agradeço ao meu amigo Samuel, que me mostrou como pavimentar o caminho a ser percorrido. E, por fim, agradeço a Deus por me permitir chegar até aqui, apesar dos pesares, com vida e intelecto para continuar irritando e sendo irritante. Expresso minha mais profunda gratidão por tudo e a todos.

"A inteligência artificial é a nova eletricidade."
Andrew Ng.

RESUMO

Observa-se uma tendência de modernização das práticas educacionais, especialmente no campo da programação. A avaliação é reconhecida como um instrumento essencial e popular para mensurar o progresso do ensino, e consequentemente o nível de conhecimento do aluno. Entretanto, a avaliação do próprio sistema de avaliação se torna imperativa para aprimorar a qualidade do ensino e obter métricas mais precisas em relação ao desenvolvimento dos estudantes. O Departamento de Computação Aplicada da Universidade Federal de Lavras utiliza a ferramenta *Dredd* para avaliar os alunos, mas enfrenta limitações estruturais, o que instigou o desenvolvimento do sistema *Vesperto*. Este novo sistema busca integrar os aspectos positivos do *Dredd* e incorporar tecnologias contemporâneas para enriquecer o processo de ensino-aprendizagem. Além disso, o potencial de oferecer aos educadores um leque mais amplo de opções para análise de desempenho, adaptações didáticas ou até mesmo intervenção pedagógica. O estudo investigativo contido neste trabalho reflete sobre a potencial aplicação de aprendizado de máquina e seu impacto na produção de resultados mais eficientes nas avaliações de desempenho através da técnica de clusterização, visando a melhoria contínua do ensino de programação. As análises realizadas visam refletir e justificar o potencial do uso de algoritmos de aprendizagem de máquina como ferramenta de avaliação de desempenho. Essa avaliação tem como potencial prever deficiências no ensino de programação que permitirá ao professor intervenções didáticas sem comprometer o andamento do processo pedagógico, mensurar a qualidade de suas avaliações e até mesmo propor alternativas personalizadas de acordo com o discente.

Palavras-chave: Inteligência Artificial. Aprendizagem de Máquina. Clusterização. Análise de Desempenho. Ensino de Programação.

ABSTRACT

There is a noticeable trend towards modernizing educational practices, especially in the field of programming. Evaluation is recognized as an essential and popular tool for measuring the progress of teaching and, consequently, the level of student knowledge. However, evaluating the evaluation system itself becomes imperative to enhance the quality of education and obtain more accurate metrics regarding student development. The Department of Applied Computing at the Federal University of Lavras uses the tool *Dredd* to assess students but faces structural limitations, prompting the development of the *Vesperto* system. This new system seeks to integrate the positive aspects of *Dredd* and incorporate contemporary technologies to enrich the teaching and learning process. Additionally, it has the potential to offer educators a broader range of options for performance analysis, didactic adaptations, or even pedagogical intervention. The investigative study in this work reflects on the potential application of machine learning and its impact on producing more efficient performance evaluations through clustering techniques, aiming for continuous improvement in programming education. The analyses conducted aim to reflect and justify the potential use of machine learning algorithms as a performance evaluation tool. This evaluation has the potential to predict deficiencies in programming education, allowing teachers to make didactic interventions without compromising the progress of the pedagogical process, assess the quality of their evaluations, and even propose personalized alternatives based on the student's needs.

Keywords: Artificial Intelligence. Machine Learning. Clustering. Performance Analysis. Programming Education.

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

A essência da educação está intrinsecamente ligada ao enfrentamento dos desafios diários no contexto do ensino e da aprendizagem, superando obstáculos que podem dificultar a transferência de conhecimento do professor para o aluno, favorecendo assim a construção do conhecimento (FREIRE, 1974).

O ensino de programação não foge dessa realidade, quando observamos em conversas de corredores as sonoras reclamações em relação as dificuldades de determinadas matérias. Esses lamúrios ecoam pela instituição, que por sua vez, representada pelos professores, busca alternativas didáticas para estas disciplinas. Tal sentimento não pode passar batido visto que a motivação é fator chave que influencia na Psicologia Escolar e Educacional (BZUNECK; BORUCHOVITCH, 2016).

Essa reflexão se torna relevante ao considerar que diferentes metodologias de ensino motivam diversos perfis de alunos, sendo essenciais também para a formulação de políticas públicas. A realidade local comprova que ainda existem excluídos que não têm contato com Tecnologias de Comunicação e Informação (TICs), como observado durante a implementação do Programa Educação Conectada no município de Lavras, que conta com a participação da UFLA (SANTOS, 2020).

A combinação do uso de ferramentas computacionais com métodos tradicionais para compreender e interpretar algoritmos representa um exemplo do "melhor dos dois mundos". Enquanto para alguns, a abordagem do "lápis e papel" facilita a compreensão da abstração lógica matemática e suas aplicações em diferentes contextos, para outros, especialmente no contexto da universidade pública, o computador serve como meio para informatizar a população, proporcionando as primeiras experiências computacionais aos estudantes (SOUZA, 2009).

Quando se analisa o contexto educacional, observa-se um crescente interesse na modernização das práticas de ensino que pode mascarar problemas inerentes à capacidade de aprendizagem de cada indivíduo. A utilização do lápis e papel como meio de representação do pseudocódigo, citada anteriormente, pode evidenciar por exemplo a dificuldade que os estudantes podem enfrentar quanto na saída de dados e não associar adequadamente o terminal como parte integrante do processo de programação. Morgado (SOUZA, 2009), por exemplo, demonstrou que o *Visualg* desempenha um papel relevante no apoio à didática neste sentido, ao tratar-se de um ambiente de desenvolvimento que proporciona aos educadores uma ampla vari-

edade de recursos, incluindo a apresentação passo a passo do código e assistência na depuração do pseudocódigo.

No contexto do processo de ensino-aprendizagem, a avaliação emerge como um instrumento fundamental para identificar essas fragilidades e mensurar o progresso do ensino. O filósofo e pedagogo Paulo Freire faz lembrar que a prática de ensinar transcende a mera transmissão de conhecimento, devendo primordialmente possibilitar a construção ativa desse saber pelo aprendiz (FREIRE, 1974). Nesse contexto, surge a necessidade de refletir sobre como efetivamente avaliar o ensino de programação, abordando tanto a perspectiva do aluno quanto o papel do professor.

A avaliação, enquanto componente que responde ao empenho e à dedicação do aluno, viabiliza a mensuração e a conceituação do processo de formação acadêmica, podendo até mesmo servir para prever possíveis resultados, como demonstra Pereira et al. (2020). Considerando uma abordagem mais ativa, é possível perceber o poder da intervenção pedagógica ao utilizar ambientes virtuais de aprendizagem. Esse potencial se torna evidente com a implementação do *SisREA* (AMARAL et al., 2021), que considera o interesse do aluno através do uso de questionário para recomendar Estratégias de Aprendizagem.

Nesse contexto, o Departamento de Computação Aplicada da Universidade Federal de Lavras, por exemplo, faz uso da ferramenta denominada *Dredd* para a avaliação do desempenho de seus alunos, um juiz online. O funcionamento desse sistema permite ao docente realizar uma pré-configuração para a correção, análise do código e fornecimento de realimentação ao aluno, com base em respostas padronizadas.

Contudo, a avaliação sobre a metodologia didática e sua eficiência quanto ao saber transmitido e saber adquirido também precisa ser avaliada, e é aqui que o *Dredd* apresenta limitações estruturais que motivaram o departamento a iniciar o desenvolvimento de um novo sistema, denominado *Vesperto*. No âmbito do *Vesperto*, serão incorporados os aspectos positivos do *Dredd*, além da introdução de tecnologias mais contemporâneas, como a implementação de técnicas de aprendizagem máquina, e a disponibilização de possibilidades mais amplas para desenvolvimentos futuros, sendo uma delas a possibilidade de avaliar as metodologias didáticas do processo de ensino aprendizagem.

A utilização de inteligência artificial para analisar o comportamento dos alunos e prever deficiências no ensino-aprendizado demonstra ter potencial de intervenções metodológicas que mitiguem essas deficiências e promovam a eficiência, com resultados de maiores índices de

aprovação (AMARAL et al., 2021). O presente trabalho é o resultado de uma reflexão sobre os elementos que podem ser agregados ao *Vesperto*, de forma a enriquecer o processo de ensino-aprendizagem, bem como a oferecer aos docentes um maior leque de opções para a análise de desempenho e eventuais adaptações na abordagem didática, caso se mostrem necessárias.

1.1 Objetivos Gerais e Específicos

O presente trabalho visa justificar o uso de Inteligência Artificial através de emprego de algoritmos de Aprendizado de Máquina, direcionando sua aplicação para a análise do desempenho de alunos em disciplinas voltadas ao ensino de técnicas de programação.

A primeira dimensão abordada neste estudo concentra-se na capacidade intrínseca dos algoritmos de Aprendizado de Máquina em agrupar dados de forma coesa e eficiente. A aplicação de técnicas de clusterização possibilita a identificação de padrões latentes no desempenho dos alunos em disciplinas de programação. Ao agrupar estudantes com características semelhantes, é possível extrair informações valiosas sobre as abordagens de aprendizado mais eficazes, permitindo uma adaptação mais precisa e personalizada das estratégias pedagógicas.

Uma consideração essencial nesta pesquisa reside na análise da complexidade linguística selecionada para a implementação dos algoritmos de Aprendizado de Máquina, juntamente com a complexidade intrínseca dos próprios algoritmos. Torna-se imperativo avaliar o impacto desses conceitos no potencial desenvolvimento e implementação dessas técnicas na realidade do *Vesperto*.

Por fim, a análise se estende à capacidade dos dados agrupados em fornecer informações sobre o perfil comportamental dos alunos, possibilitando uma compreensão mais profunda além do desempenho didático. A busca por padrões comportamentais nos agrupamentos dos dados permite identificar características específicas que influenciam o aprendizado em disciplinas de programação.

Dessa forma, a escolha por algoritmos de Aprendizado de Máquina e técnicas de clusterização específicas fundamenta-se na busca por uma compreensão mais aprofundada e eficiente do desempenho dos alunos em disciplinas de programação. A implementação dessas abordagens visa não apenas à análise retrospectiva, mas também à projeção e ao aprimoramento contínuo do processo educacional, visando entender os benefícios dessa implementação no ambiente do *Vesperto*, para identificar deficiências no ensino e previsão de desempenho baseado nas metodologias aplicadas.

1.2 Estrutura do Trabalho

O presente trabalho propõe-se a fornecer uma base investigativa que justifique a implementação de Inteligência Artificial (IA) no ambiente educacional, especificamente no contexto do *Vesperto*, que será desenvolvido pelo DCA da UFLA. Este trabalho está estruturado em capítulos que abordam diferentes aspectos relacionados à IA e seu potencial impacto no aprimoramento do processo educacional. A busca por fundamentos teóricos e práticos visa não apenas compreender as vantagens da IA, mas também orientar futuras ações para a efetiva incorporação dessa tecnologia no *Vesperto*. Cada capítulo é desenvolvido de forma a contribuir para a construção de um argumento consistente que respalde a necessidade e os benefícios da adoção da IA no contexto educacional proposto.

O Capítulo 2 desempenha um papel fundamental ao oferecer o referencial teórico que norteia os pensamentos e análises dos dados desta pesquisa. Nele, são apresentados temas cruciais como Aprendizado de Máquina, Mineração de Dados e Clusterização. Explora-se a fundo os algoritmos utilizados nesse contexto, promovendo comparações entre eles e destacando as medidas de avaliação aplicadas. A abordagem cuidadosa desses elementos proporciona uma base sólida para a compreensão do arcabouço teórico que sustenta a análise dos dados ao longo deste estudo. Este capítulo visa, assim, estabelecer as bases conceituais necessárias para a compreensão aprofundada das metodologias e técnicas empregadas na implementação de IA no *Vesperto*.

O Capítulo 3 detalha as ferramentas essenciais empregadas nesta pesquisa. Ele fornece uma visão abrangente das amostras utilizadas como base, elucidando os procedimentos de coleta de dados, assim como as etapas envolvidas no processamento e transformação desses dados. Além disso, são apresentadas justificativas detalhadas para a escolha do algoritmo adotado, com ênfase nos motivos que o embasaram. A decisão sobre a quantidade de clusters também é justificada, evidenciando o raciocínio por trás dessa escolha. Este capítulo, assim, oferece uma visão abrangente e fundamentada dos métodos empregados, proporcionando um arcabouço metodológico para a condução da pesquisa no contexto de possível implementação.

O Capítulo 4 expõe detalhadamente as análises realizadas e o desenvolvimento do estudo utilizando a linguagem de programação Python. Nele, são apresentados os resultados provenientes das análises dos agrupamentos realizadas, fornecendo uma visão abrangente das respostas obtidas e conclusões tiradas a partir dos dados examinados. O uso dos algoritmos como ferramenta para o desenvolvimento é discutida, destacando a sua relevância e importân-

cia no contexto das análises de aprendizado de máquina e clusterização realizadas. Este capítulo busca oferecer não apenas um registro objetivo dos resultados, mas também uma interpretação das implicações desses resultados para o propósito geral da pesquisa.

O Capítulo 5 encerra este trabalho, expondo as conclusões derivadas da análise dos resultados obtidos. Neste contexto, será destacado como as deficiências identificadas nos sistemas legados impactaram diretamente neste estudo, evidenciando de que maneira essas limitações influenciaram as conclusões apresentadas. Além disso, serão exploradas possíveis abordagens para superar ou mitigar tais deficiências, contribuindo para uma visão mais abrangente sobre a aplicabilidade dos métodos estudados.

Na sequência, serão propostos estudos futuros que têm o potencial não apenas de complementar este trabalho, mas também de otimizar significativamente o processo de desenvolvimento.

2 REFERENCIAL TEÓRICO

No cenário atual da era digital, a tecnologia de *Aprendizagem de Máquina* representa um campo fundamental que revoluciona a forma como lidamos com dados complexos e volumes massivos de informações. Em paralelo, a *Mineração de Dados*, outro conceito-chave nesta área, envolve a exploração e análise de grandes conjuntos de dados para descobrir padrões, correlações e informações úteis. Uma das técnicas mais poderosas dentro da mineração de dados é a *clusterização*, que agrupa dados similares em clusters ou grupos, facilitando a compreensão dos padrões subjacentes nos dados (WITTEN; FRANK; HALL, 2011).

Este capítulo terá como finalidade a apresentação desse conceitos e aprofundamento dos temas relacionados a *clusterização* e seus algoritmos para definir as referências bases para estudo e implementação das atividades desse trabalho.

2.1 Aprendizagem de Máquina

Assim como em Witten, Frank e Hall (2011), não temos como foco definir o conceito de aprendizagem ou a capacidade do computador de aprender, visto que tal árdua tarefa é do campo da filosofia. Assim seguiu-se pela abordagem prática do uso de técnicas para encontrar e descrever padrões encontrados em conjuntos de dados.

Em um contexto prático, Aprendizagem de Máquina (AM) ou *Machine Learning* (termo em inglês) refere-se à capacidade dos sistemas de computadores de aprender padrões complexos a partir de dados e melhorar seu desempenho em uma tarefa específica ao longo do tempo, sem serem explicitamente programados. Em vez de depender de regras de programação fixas, os algoritmos de AM usam dados para treinar modelos e fazer previsões ou tomar decisões com base nessas informações.

Por exemplo, em um sistema de recomendação de filmes, AM pode analisar o histórico de preferências de um usuário, como os filmes que ele assistiu e gostou, para prever quais outros filmes ele poderia gostar no futuro baseando-se em informações de autor, gênero, atores ou até mesmo de duração da película. Da mesma forma, em aplicações de reconhecimento de voz, algoritmos de AM podem ser treinados com enormes conjuntos de dados de áudio para entender e transcrever fala humana com precisão.

Em resumo, em um contexto prático, AM capacita sistemas a aprender com dados passados, identificar padrões e tomar decisões ou fazer previsões com base nesses padrões, tornando-

o essencial em uma variedade de aplicações do mundo real, desde assistentes virtuais até diagnósticos médicos.

2.2 Mineração de dados

A Mineração de Dados (MD) ou *Data Mining* (termo em inglês), também conhecida como Descoberta de Conhecimento em Bancos de Dados (KDD, do inglês Knowledge Discovery in Databases), é um processo de encontrar padrões, informações significativas e conhecimento útil a partir de grandes volumes de dados. É uma disciplina interdisciplinar que combina técnicas e teorias da estatística, inteligência artificial, aprendizado de máquina, reconhecimento de padrões e banco de dados para analisar e interpretar dados.

O processo de mineração de dados geralmente envolve várias etapas tal como exemplifica o trabalho (WITTEN; FRANK; HALL, 2011). Inicialmente, os dados brutos são coletados de diferentes fontes, como bancos de dados corporativos, redes sociais, sensores, entre outros. Em seguida, esses dados são pré-processados para limpar ruídos, tratar valores ausentes e transformar os dados em um formato adequado para análise.

Após a preparação dos dados, várias técnicas de mineração são aplicadas para descobrir padrões. Estas incluem técnicas de clusterização, que agrupam dados similares em clusters para identificar relações entre eles; técnicas de classificação, que categorizam os dados em classes ou rótulos predefinidos com base em suas características; técnicas de regressão, que modelam a relação entre variáveis; e técnicas de associação, que encontram relações interessantes entre diferentes variáveis. Todas estas técnicas podem ser encontradas no livro de Han, Kamber e Pei (2012).

A tecnologia de Mineração de Dados, ao extrair informações valiosas de bases de dados e oferecer uma nova perspectiva analítica, desempenha um papel essencial em diversos setores, incluindo o *Business Intelligence* (BI). O BI utiliza métodos e conceitos implementados por meio de softwares para transformar dados organizacionais em conhecimentos que auxiliam na tomada de decisões estratégicas. Ao reunir dados cruciais em um único local, o BI transforma informações em conhecimento, proporcionando vantagens em um mercado competitivo (GIULIANO, 2012).

Além disso, a tecnologia OLAP (On-Line Analytical Processing), ou Processamento Analítico On-Line, complementa essa abordagem ao permitir uma visão multidimensional dos dados da organização. Ao oferecer consultas que disponibilizam dados relacionados a medidas,

desmembrados em diversas dimensões, o OLAP possibilita que as informações sejam visualizadas e analisadas de várias perspectivas, mantendo a estrutura de dados de forma adequada e flexível para os usuários.

2.3 Clusterização

A Clusterização de Dados no âmbito da Mineração de Dados (MD), também conhecida apenas como Clusterização, desempenha um papel crucial ao agrupar dados semelhantes em conjuntos distintos, conhecidos como clusters, com base em suas características comuns. O propósito subjacente a essa abordagem, fundamentado no Aprendizado de Máquina, reside na organização significativa de conjuntos de dados complexos, proporcionando uma compreensão mais profunda das estruturas inerentes a esses conjuntos (JAIN; MURTY; FLYNN, 1999).

A diversidade das técnicas de clusterização é evidenciada na apresentação de modelos distintos para a organização dos dados. A diferenciação dos pontos dentro de cada cluster, assim como o modelo de cálculo de distância, exemplifica as características distintivas de cada método. Métodos como *K-means*, *K-medoids*, e *DBSCAN* desempenham um papel fundamental na análise de dados, permitindo a identificação de padrões e estruturas nos conjuntos de dados. Cada método adota uma abordagem singular para agrupar os dados, sendo aplicável a diferentes problemas e estruturas de dados (HAN; KAMBER; PEI, 2012).

Os objetivos primordiais da clusterização de dados englobam a representação mais compacta e informativa dos dados, facilitando a compreensão e análise por meio da redução da dimensionalidade. Além disso, a identificação de padrões e estruturas nos dados é viabilizada, proporcionando conhecimentos valiosos sobre os conjuntos de dados. A segmentação dos dados com base em características comuns possibilita aplicações diversas, como em estratégias de marketing e publicidade (DAM; DINH; MENVIELLE, 2019) ou suporte à saúde (REDDY; AGGARWAL, 2015).

2.3.1 K-means

O algoritmo *K-means* é um método de clusterização com foco nos centróides. Como conceituada e traduzida do trabalho (HAN; KAMBER; PEI, 2012):

> Uma técnica de particionamento baseada em *centróides* utiliza o centróide de um cluster, *Ci*, para representar esse cluster. Conceitualmente, o centróide de um cluster é seu ponto central. O centróide pode ser definido de várias

> maneiras, como pela média ou mediana dos objetos (ou pontos) atribuídos ao cluster.

Ele agrupa os dados em K clusters, onde cada cluster é representado por um centróide, um ponto médio que minimiza a soma das distâncias euclidianas entre os pontos do cluster e o centróide. O *K-means* é eficaz para dados onde os clusters têm uma forma esférica e tamanho semelhante. Ele é iterativo e requer a definição prévia do número de clusters (K), sendo sensível à inicialização dos centróides.

2.3.2 K-medoids

A outra técnica utilizada de clusterização é centrada em medoids. O medoid é o ponto que minimiza a soma das distâncias para todos os outros pontos no mesmo cluster. Em vez de usar centróides, o K-medoids usa pontos de dados reais como representantes dos clusters, tornando-o mais robusto em relação a valores atípicos, são pontos de dados que se diferenciam significativamente do restante do conjunto de dados. O K-medoids é útil quando os dados possuem pontos fora da curva ou quando a forma dos clusters não é esférica, pois ele é menos sensível a pontos extremos (HAN; KAMBER; PEI, 2012).

2.3.3 DBSCAN

O *DBSCAN* (Density-Based Spatial Clustering of Applications with Noise) é um método de clusterização baseado em densidade (HAN; KAMBER; PEI, 2012). Ele agrupa os dados com base na densidade local, identificando áreas onde as observações são densas e semelhantes. Os clusters no *DBSCAN* podem ter diferentes formas e tamanhos, e ele é capaz de detectar pontos de dados que não pertencem a nenhum cluster específico, conhecidos como ruídos, tornando o *DBSCAN* é vantajoso neste contexto.

2.3.4 Comparações, Vantagens e Desvantagens

Em resumo, enquanto o *K-means* é eficaz para clusters esféricos e requer a definição prévia do número de clusters, o *K-medoids* é mais robusto em relação a valores atípicos e não depende da forma dos clusters. Os dois modelos fazem parte dos grupos de de algoritmos que metodizam a divisão da base em partições, os conhecidos Métodos Particionais (HAN; KAMBER; PEI, 2012).

Por outro lado, o *DBSCAN* é adequado para clusters de formas variadas e é capaz de identificar pontos fora da curva automaticamente. A escolha do método de clusterização depende da natureza dos dados e dos objetivos da análise.

A Clusterização opera por meio de algoritmos que analisam as semelhanças entre os dados e os agrupam de acordo com essas similaridades. Algoritmos como *K-means*, *K-medoids* e *DBSCAN* são amplamente empregados para identificar padrões intrínsecos nos dados. Por exemplo, em um conjunto de dados de clientes de um comércio eletrônico, a clusterização pode agrupar clientes com comportamentos de compra semelhantes, permitindo que as empresas personalizem estratégias de marketing para cada grupo de clientes de maneira eficaz.

Esse processo é crucial no contexto acadêmico, sendo amplamente discutido em diversos trabalhos científicos e estudos de pesquisa. Referências teóricas (JAIN; DUBES, 1988) fornecem uma base sólida para o desenvolvimento de novos métodos e técnicas na área da Mineração de Dados. Além disso, a contínua pesquisa e aplicação da clusterização contribuem significativamente para o avanço do conhecimento, oferecendo interessantes estratégias para resolver problemas complexos em diversos setores.

A clusterização encontra aplicação em diversas áreas, desde marketing e finanças até biologia e medicina. Na bioinformática, por exemplo, é utilizada para agrupar sequências genéticas semelhantes, auxiliando na classificação de organismos. No âmbito deste trabalho, há inúmeras possibilidades de implementação. No entanto, a análise e previsão do comportamento do aluno com base em seu histórico acadêmico se destacam como uma abordagem promissora. Considerando que o novo sistema poderia armazenar dados, como o tipo de atividade, poderia utilizar o histórico dessas atividades pelos alunos para obter informações, como a quantidade de alunos que apresentariam desempenho inferior em tipos específicos de questões, como as questões abertas.

2.4 Medidas para Avaliação da Clusterização

A aplicação exclusiva do modelo de clusterização não resulta em saídas otimizadas ou prontas para estudo. É crucial incorporar uma orientação para avaliação da qualidade interna dos clusters a fim de superar o desafio de extrair informações relevantes da vastidão de dados passíveis de avaliação. O trabalho de Oliveira (2018) destaca a existência de três tipos comuns de avaliação: Avaliação Interna, Avaliação Externa e Avaliação Relativa, as quais serão detalhadas nas seções subsequentes 2.4.1, 2.4.2 e 2.4.3, respectivamente. A seção 2.4.3 também

apresenta detalhes sobre o método de avaliação escolhido e sua aplicação ao contexto deste trabalho.

2.4.1 Avaliação Externa

A avaliação externa avalia a clusterização utilizando informações externas ou rótulos predefinidos. Esses rótulos indicam a verdade conhecida sobre a pertinência dos pontos de dados a determinados clusters ou grupos. Em outras palavras, a avaliação externa compara os clusters produzidos pelo algoritmo com uma referência ou *ground truth*, como classes reais ou categorias conhecidas. Essa abordagem permite medir a precisão da clusterização ao verificar o quanto os clusters obtidos coincidem com as categorias ou grupos verdadeiros dos dados. Métricas de avaliação externa, como índice de Rand Ajustado (ARI) e índice de Fowlkes-Mallows (FMI), comparam a semelhança entre as partições produzidas pelo algoritmo de clusterização com as categorias de referência. A avaliação externa é importante quando os dados possuem rótulos verdadeiros disponíveis e ajuda a determinar a eficácia do algoritmo de clusterização em reproduzir as verdadeiras estruturas de grupos presentes nos dados (OLIVEIRA, 2018).

2.4.2 Avaliação Relativa

A avaliação relativa é a comparação e avaliação de desempenho de diferentes algoritmos de clusterização ou configurações de parâmetros usando métricas internas ou externas, sem depender de uma *ground truth* ou rótulos predefinidos. Em outras palavras, a avaliação relativa envolve comparar a qualidade dos clusters obtidos por diferentes métodos ou configurações, geralmente em um contexto de análise exploratória, para determinar qual abordagem produz os clusters mais significativos e bem-estruturados para um determinado conjunto de dados. Ao comparar algoritmos ou configurações de parâmetros usando medidas relativas, pode-se tomar decisões informadas sobre qual técnica de clusterização é mais adequada para os dados em questão. Métricas como índice de Davies-Bouldin são frequentemente usadas para essa avaliação comparativa, proporcionando uma visão objetiva da qualidade dos clusters sem depender de uma verdade absoluta ou rótulos prévios. A avaliação relativa é valiosa para selecionar a melhor abordagem de clusterização em situações onde não há categorias de referência conhecidas (OLIVEIRA, 2018).

2.4.3 Avaliação Interna

A avaliação interna, no contexto de clusterização de dados, refere-se à avaliação da qualidade dos clusters formados pelo algoritmo de clusterização com base apenas nas informações dos dados, sem depender de rótulos ou categorias predefinidas. Em outras palavras, a avaliação interna utiliza medidas e métricas intrínsecas que avaliam a coesão e a separação dos clusters com base nas características dos dados em si. Medidas internas como a inércia (soma dos quadrados intra-cluster), índice de Davies-Bouldin e índice de silhueta, permitem quantificar a qualidade da clusterização observando a compactação dos pontos dentro de cada cluster e a separação entre os clusters. Ao utilizar a avaliação interna, os pesquisadores podem determinar o número ideal de clusters e comparar diferentes algoritmos de clusterização sem depender de informações externas ou rótulos verdadeiros, tornando-a uma abordagem objetiva e independente para avaliar a qualidade dos resultados da clusterização (OLIVEIRA, 2018).

A inércia é um conceito fundamental utilizado na determinação do número ideal de clusters em algoritmos de clusterização, como o *K-means*. A inércia, também conhecida como soma dos quadrados intra-cluster, mede a dispersão dos pontos de dados dentro de cada cluster. Em outras palavras, representa a soma das distâncias ao quadrado entre cada ponto de dados e o centróide do cluster ao qual pertence. Quanto menor a inércia, mais compactos e coesos são os clusters, indicando uma melhor separação dos grupos de dados.

Ao utilizar a inércia, pode-se realizar análises exploratórias para determinar o número apropriado de clusters. Geralmente, calcula-se a inércia para diferentes valores de K (número de clusters) e observa-se como a inércia diminui à medida que o número de clusters aumenta, o que também ocorre neste trabalho. O ponto em que a diminuição na inércia começa a ser marginal, ou seja, começa a ser mínima, é frequentemente considerado o número ideal de clusters para o conjunto de dados em questão (OLIVEIRA, 2018).

Este critério é amplamente discutido na literatura acadêmica, sendo utilizado em muitos estudos de pesquisa para encontrar uma estrutura de clusterização significativa e interpretável. A análise cuidadosa da inércia ajuda a tomar decisões informadas sobre o número ótimo de clusters, garantindo uma clusterização mais eficaz e relevante para suas análises buscando otimizar a coerência interna dos clusters (ARTHUR; VASSILVITSKII, 2007).

3 METODOLOGIA

Nesta seção, serão delineados os pormenores da metodologia empregada na pesquisa investigativa do potencial uso de algoritmos de clusterização.

É essencial salientar que, para preservar a confidencialidade e garantir a objetividade do estudo, os dados originais foram anonimizados e transformados em informações que desconsideram a individualidade dos participantes. Esta medida visa assegurar a imparcialidade e a neutralidade na análise dos resultados.

Ressalta-se também que as análises realizadas têm como foco a avaliação dos métodos de avaliação como uma ferramenta para promover o desenvolvimento acadêmico do aluno. Nesse sentido, as análises foram conduzidas com base em premissas pré-estabelecidas, empregando uma perspectiva tendenciosa que se alinha com o ponto de vista do professor.

As etapas do presente trabalho foram desenvolvidas com base nas metodologias de estudo propostas por Witten, Frank e Hall (2011). Mais especificamente, as seções de 3.1 a 3.4 são embasadas no conteúdo do Capítulo 2 do referido livro. Já as seções subsequentes têm como fundamento o Capítulo 3 do mesmo livro. A escolha do principal algoritmo de estudo, o *k-means*, foi feita seguindo a sugestão do autor no Capítulo 4, enquanto que os algoritmos secundários foram selecionados através de pesquisa em blogs, com foco no estudo de Inteligência Artificial utilizando bibliotecas para programação em Python, levando em consideração sua popularidade. Dessa forma, foram escolhidos o *k-medoids* e o *DBSCAN* como algoritmos secundários.

3.1 Ferramentas Utilizadas

Para a manipulação e análise dos dados neste estudo, a linguagem de programação Python foi amplamente empregada, juntamente com suas bibliotecas especializadas em processamento de dados, com o intuito de aplicar algoritmos de clusterização. O software *LibreOffice - Calc* também foi empregado para leitura e confirmação dos dados em CSV. Além desses, o sistema *ChatGPT*[1] da OpenAI foi utilizado para auxiliar na interpretação dos scripts e facilitar a extração de dados.

A escolha de utilizar o *ChatGPT* para esse propósito foi justificada pela natureza central da atividade do estudo, que envolve a análise dos dados produzidos. O script extraído

[1] Disponível em <https://chat.openai.com/> e acessado entre os meses de Agosto e Dezembro de 2023.

do HTML apresentava peculiaridades em sua arquitetura, o que complicou o desenvolvimento de uma técnica de extração apropriada. Portanto, o uso do *ChatGPT* para criar um script em Python proporcionou economia de tempo e agilidade no processo de extração de dados, além de contribuir para a identificação e correção de erros nos códigos Python desenvolvidos.

Finalmente, para converter os documentos PDF em um formato editável (.ODS), foi utilizada a ferramenta online gratuita do Adobe Acrobat.

Ressalta-se que o desenvolvimento do trabalho foi na aplicação dos três algoritmos e suas bibliotecas para gerar dados de análise e comparação, sendo eles *k-means, k-medoids* e o *DBSCAN*.

3.2 Descrição da Amostra

A amostra utilizada neste estudo consistiu em dados extraídos do juiz online *Dredd* e dos registros dos Diários de Nota (que será referenciado apenas como Diário daqui adiante) dos professores de uma turma específica de Estrutura de Dados no ano de 2023, na Universidade Federal de Lavras. Esses dados foram extraídos de arquivos HTML e em documentos PDF que continham informações sobre o desempenho final dos alunos.

Estes arquivos no formato HTML foram obtidos do site oficial do sistema *Dredd*, acessado por meio do perfil do docente responsável, utilizado para administrar e, quando necessário, corrigir manualmente as atividades propostas para a turma.

Os documentos em formato PDF correspondem aos Diários, que contêm informações relevantes sobre o contexto das avaliações dos discentes. As amostras incluíam informações apenas referentes ao intervalo semestral que se encerra na primeira prova. Em outras palavras, os dados das atividades preparatórias e dos diários até a primeira prova foram considerados. Em termos de conteúdo, a primeira prova abordou os temas de *Heap, Fila e Pilha*.

3.3 Procedimentos de Coleta de Dados

A fase inicial da coleta de dados envolveu a extração das informações contidas no script HTML gerado pelo sistema *Dredd*. Este script apresentava características que dificultaram a compreensão inicial devido à presença de uma combinação de tags HTML, principalmente a tag ‘<table>‘, tornando desafiadora a tarefa de separar os dados para posterior análise em formato CSV.

Para superar essa dificuldade, foram identificados padrões de alternância entre as diferentes tabelas geradas no script. A fim de automatizar esse processo, recorreu-se ao *ChatGPT* para localizar os intervalos específicos pertinentes. A intervenção do *ChatGPT* revelou-se instrumental, pois facilitou a identificação dos trechos comuns de variação, permitindo o isolamento das entidades relevantes contendo os dados necessários, ao mesmo tempo em que eliminava caracteres irrelevantes e estruturava as informações de forma mais compreensível para posterior análise.

Na fase subsequente, o arquivo CSV gerado foi utilizado por um script em Python para modificar o atributo "Tentativa". Este script converteu a string originada do HTML em um valor de ponto flutuante, indicando a quantidade de tentativas necessárias para obter a nota da atividade. Outro atributo alterado nessa etapa foi relacionado às notas pertinentes das atividades. Após a conversão para valores de ponto flutuante, foi calculada a média aritmética das tentativas, multiplicando-a pelas notas para obter o valor de desempenho. Esse atributo foi criado com o propósito de substituir o atributo de nota original, equilibrando todas as atividades em relação às dificuldades dos alunos.

Os documentos PDF obtidos dos diários do perfil dos professores foram processados pela ferramenta online da Adobe, permitindo a conversão para o formato '.ods', o que facilitou a manipulação dos dados. Nesse contexto, a utilização do LibreOffice, uma suíte de aplicativos de código aberto, foi fundamental. O LibreOffice inclui o programa Calc, capaz de manipular arquivos no formato ODS (OpenDocument Spreadsheet). Essa etapa foi necessária para garantir a preservação das configurações básicas do arquivo durante o processo de conversão para o formato ODS, assegurando, assim, a integridade dos dados.

Com os dados extraídos, foi realizado o processo de remoção de informações irrelevantes e o filtro das entidades pertinentes e seus atributos mais relevantes. Duas premissas fundamentaram esse procedimento: primeiro, que o atributo "atividade" fosse registrada como "resposta submetida", indicando que o aluno tinha efetivamente completado a tarefa; segundo, que o aluno que entregou a atividade demonstrasse intenção de utilizá-la como ferramenta de estudo. Ao considerar a marcação "resposta submetida", foram excluídos os alunos que não enviou a atividade proposta para correção. Além disso, ao levar em conta a disposição do aluno, validou-se a premissa de que houve esforço por parte dos discentes.

3.4 Processamento e Transformação dos Dados

Na presente solução, foi adotada uma abordagem metodológica rigorosa para processar dados brutos provenientes de fontes heterogêneas. O método inicia-se com a leitura de dois conjuntos de dados, um em formato CSV e outro em ODS, que representam informações sobre atividades realizadas. Utilizando uma abordagem de orientação a objetos no desenvolvimento, foram criadas as classes *DadosCSVObject* e *DiarioODSObject*, permitindo a estruturação e associação dos dados e a criação de objetos para cada linha dos conjuntos de dados originais. É fundamental destacar que a escolha de estruturar os dados em objetos não apenas proporciona uma organização eficiente, mas também reflete a aplicação de princípios orientados a objetos, contribuindo para uma arquitetura de código mais modular e reutilizável.

O processo de correspondência entre os objetos foi cuidadosamente delineado, priorizando a precisão e a consistência. A estratégia adotada consiste em utilizar a primeira posição da *string* da coluna 'Tentativa', pois na origem dos dados temos uma expressão no padrão "*n* de *n* tentativas". Esta decisão foi fundamentada na natureza dos dados, uma vez que, para este trabalho, não foi possível obter os valores precisos presentes no banco de dados, pois o sistema não possui fácil acesso que permita a extração desses dados.

As notas dos alunos extraídas do documento "Diário" fornecido foram convertidas para o atributo "conceito", representando um valor binário de "Aprovado" ou "Reprovado". Isso permitiu a comparação dos dados processados com o desempenho final dos alunos nas provas. Em termos computacionais, os valores interpretados pelos scripts foram definidos como "True" para "Aprovado" e "False" para "Reprovado".

Para determinar os casos de reprovação, foram considerados os alunos que obtiveram o conceito 'A' no Diário, o que a instituição caracteriza como "ausente" na realização da avaliação, equivalente a "False". Ou seja, por motivos diversos, o aluno não pôde comparecer e de acordo com as normas da instituição foi reprovado na avaliação.

Para o atributo 'valorDesempenho', foi calculada a média de tentativas multiplicada pela nota obtida nas atividades de estudo. Esse valor foi necessário para tentar equiparar os pesos das notas para diminuir a discrepância de uma nota muito alta numa atividade considerada fácil e outra nota baixa numa atividade complexa. É relevante mencionar que a limitação do valor na coluna 'valorDesempenho' a até 5 casas decimais é uma prática que não apenas aprimora a legibilidade e a compreensão dos dados, mas também evita problemas de precisão numérica em cálculos subsequentes.

Após a obtenção dos dados relevantes, foi realizado um processo de anonimização para eliminar informações que pudessem identificar os discentes. O primeiro passo envolveu a substituição dos nomes dos alunos por números randomizados, gerados no momento da execução do código e concatenados ao final de uma string.

3.5 Análise dos Dados Obtidos

Os dados obtidos consistem em informações processadas, contendo apenas os dados relevantes. Essas informações referem-se à entidade aluno no contexto do sistema *Dredd* e dos diários. A origem desses dados foi eliminada após a manipulação e processamento, garantindo que os dados apresentados não possam ser rastreados até sua fonte original.

Para análise de viabilidade o primeiro cenário de aplicação do algoritmo foi implementar o *k-means*, *k-medoids* e *DBSCAN* para clusterizar os dados processados a partir dos atributos de questão e valor de desempenho.

Os dados também passaram por análises que tinham como foco manter a lógica de uso de um Sistema Gerenciador de Banco de Dados. Os SGBDs são softwares que permitem organizar, armazenar, recuperar e manipular dados de forma eficiente em bancos de dados e são amplamente utilizados em aplicativos que requerem armazenamento e recuperação de grandes volumes de informações, facilitando a gestão e a interação com os dados de forma rápida e segura. A opção por realizar as análises com essa técnica foi motivada devido a natureza do projeto, logo considerou-se que os dados teriam semelhanças com dados oriundos de SGDBs de modelo relacional (SQL), e assim facilitar o uso futuro do material aqui desenvolvido para a aplicação real no *Vesperto*.

3.6 Escolha do Algoritmo

O *K-means* foi escolhido como algoritmo base devido à sua ampla popularidade e à facilidade de compreensão que oferece. Esses atributos são cruciais, especialmente quando se considera a necessidade de explicar o processo e os resultados a uma audiência diversificada, incluindo aqueles que podem não ter um profundo conhecimento em aprendizado de máquina e inteligência artificial.

Ressalta-se que o *Vesperto* é um projeto do Departamento de Computação Aplicada e que os alunos que continuarão o trabalho podem não estar no momento do curso em que

verão conceitos de inteligência artificial. A simplicidade do *K-means* facilita a interpretação dos resultados e a comunicação efetiva das descobertas, tornando-o uma escolha pragmática em contextos nos quais a clareza é fundamental. Além disso, a implementação do *K-means* é relativamente direta e eficiente, sendo uma opção eficaz para análises exploratórias iniciais e para casos em que a interpretabilidade é tão importante quanto a precisão.

Embora *K-medoids* e *DBSCAN* possam oferecer vantagens específicas em determinados cenários, a escolha do *K-means* como algoritmo base é respaldada pela sua facilidade de uso e compreensão, fatores que desempenham um papel crucial na disseminação efetiva dos resultados e na aceitação do método escolhido.

3.7 Escolha da Quantidade de Cluster

Buscando fundamentar ainda mais os estudos no contexto dos métodos de clusterização, especificamente com o K-means, a inércia, como mencionado anteriormente, serve como uma métrica de avaliação interna. Esta métrica representa a soma das distâncias quadradas entre cada ponto de dados e o centro do cluster ao qual foi atribuído, também conhecido como centróide. Em termos simples, a inércia mede a compactação dos clusters, indicando que quanto menor a inércia, mais compactos e bem definidos são os clusters.

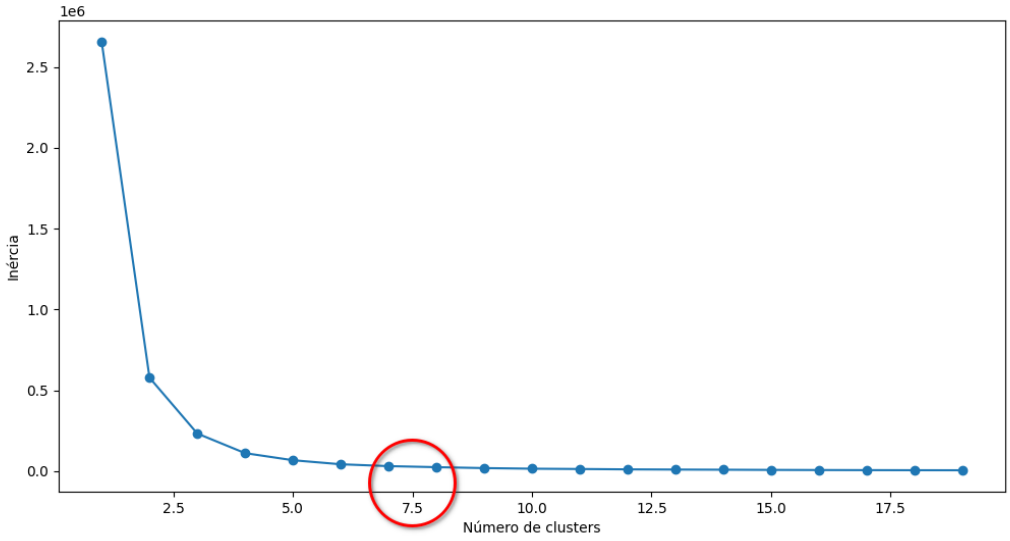
Este conceito foi então aplicado ao trabalho com o objetivo de minimizar a distância dos centróides, buscando assim encontrar o valor ideal para o total de clusters que proporcionaria maior coesão entre eles. Isto é relevante visto a necessidade de deixar os agrupamentos mais coesos. Ao observar a figura 3.1, nota-se que a variação do valor de inércia pode ser considerada irrelevante. Mais especificamente, após o total de 7 clusters, não se observa uma curva de impacto significativo no trabalho e portanto sem melhoria substancial na coesão entre eles. Essa informação é crucial para a definição do número ótimo de clusters no contexto do K-means, proporcionando entendimento valioso para a análise dos dados.

A determinação do número de clusters para os algoritmos de aprendizado de máquina foi realizada com base em critérios específicos. No caso do algoritmo *K-means*, a escolha recaiu sobre o total de 7 clusters, fundamentada no ponto de inflexão identificado no gráfico da curva de inércia, conforme ilustrado na figura 3.1. O algoritmo *K-medoids*, devido à sua similaridade com o *K-means*, também foi configurado para trabalhar com 7 clusters. Já o *DBSCAN*, por sua natureza conceitualmente distinta em relação aos demais algoritmos, teve sua quantidade de clusters determinada de maneira apropriada ao seu contexto específico.

No contexto da implementação do algoritmo *DBSCAN*, a definição das características do agrupamento é conduzida através dos parâmetros `eps` (raio da vizinhança) e `min_samples` (número mínimo de amostras em uma vizinhança para formar um *cluster*). Assim ao aplicar o algoritmo sobre o conjunto de dados, cada ponto é atribuído a um *cluster*. Diferentemente de métodos como *K-means*, o *DBSCAN* não requer uma predefinição do número de *clusters*, afinal o mesmo opera com base na densidade de pontos, formando *clusters* onde a densidade é alta e considerando pontos isolados como possíveis ruídos. A quantidade exata de *clusters* não é especificada antecipadamente, sendo uma propriedade intrínseca do algoritmo.

Devido a determinados comportamentos dos dados e dos gráficos no decorrer do projeto, foram utilizados valores diferentes para as quantidades de clusters. Essa mudança foi necessária para adaptar-se à realidade dos dados e em como as informações foram geradas. Na situação de busca por padrão de comportamento baseado no perfil do aluno – fase de estudo relacionado a análise dos agrupamentos dos rótulos IDs, Conceitos e ValorDesempenho – a quantidade de clusters definida foi 5 para forçar com que cada cluster tivesse mais elementos. Nessa situação, é necessário se fazer saber que o total de elementos para se analisar os dados foi de 53 "Ids". Portanto foi deliberadamente ignorado a avaliação interna afim de se observar o potencial do algoritmo para avaliar comportamento nessas condições de menor quantidade de elementos.

Figura 3.1 – Gráfico dos valores de inércia calculados para o algoritmo *K-means* para estudo dos elementos agrupados pelos rótulo Questões X ValorDesempenho



Fonte: Gráfico plotado pela biblioteca Matplotlib do Python

4 RESULTADOS E ANÁLISES

Neste capítulo, serão apresentados os resultados obtidos por meio da análise das entradas que foram trabalhadas, processadas e interpretadas em saídas, que foram descritas nos capítulos anteriores. O foco deste capítulo é refletir sobre as afirmações e hipóteses, buscando justificar como os dados obtidos descrevem os comportamentos dos alunos nas aulas, nas atividades e, por fim, nas avaliações. Além disso, será investigado como esse comportamento pode impactar positiva ou negativamente na qualidade do ensino e da aprendizagem e consequentemente seu impacto na taxa de "aprovados" e "reprovados".

Os dados apresentados fazem parte da análise no mesmo contexto em diferentes algoritmos, e assim as imagens se seguiram com o título evidenciando cada um dos algoritmos. Contudo, ressalta-se que os dados tem como foco o uso do *K-means* sendo as outras imagens apenas uma métrica de avaliação complementar.

4.1 Análise do desenvolvimento do trabalho utilizando Python

A escolha do algoritmo *K-means* foi fundamentada na simplicidade de implementação, em seu amplo potencial de aplicação, e também na consideração da acessibilidade para estudantes que ainda não cursaram disciplinas de inteligência artificial ou possuem conhecimento limitado sobre o assunto. A opção por implementar o algoritmo usando bibliotecas em Python proporcionou uma abordagem amigável, o que permitiria com que alunos em diferentes estágios da graduação possam se envolver ativamente no processo de desenvolvimento do *Vesperto*.

Ao utilizar o *K-means* por meio de bibliotecas em Python, observou-se que o algoritmo também se destaca pelo potencial de aprendizado mais acelerado e pela simplicidade de compreensão, características que facilitam sua assimilação por estudantes menos familiarizados com conceitos avançados de inteligência artificial. Essa abordagem acessível cria um ambiente de aprendizado inclusivo, o que permitiria aos alunos dedicarem mais tempo à análise dos dados em si, uma vez que não se perderiam nas complexidades algorítmicas. A simplicidade do *K-means* não apenas simplifica a implementação prática, mas também se traduz em uma experiência educacional mais intuitiva, sendo valiosa neste contexto de potencial uso.

Como observado na Figura 4.1, o algoritmo K-means foi aplicado a um conjunto de dados representando o desempenho em questões de avaliação. O número de clusters foi fixado em 5 para esta avaliação. Após a execução do algoritmo, os resultados foram analisados, revelando informações valiosas sobre a distribuição dos participantes em clusters distintos.

Por mais que o conceito de "facilidade" seja subjetivo, a utilização de Python como linguagem favoreceu desenvolvimento deste trabalho observando a Figura 4.2. Para mudarmos as referências para outro contexto de clusterização foi necessário apenas mudar a linha onde se tem a construção do objeto "dataKmeans". Esse fato demonstra que um aluno tendo um básico conhecimento na linguagem ou tendo domínio dos conceitos de Orientação a Objetos, poderia desenvolver scripts em Python para o *Vesperto*.

Figura 4.1 – Código desenvolvido para clusterização de dados a partir dos atributos de *id X valorDesempenho X conceito*

```
# Autor: Stanley P. M. Silva
# Objetivo: Este programa realiza a clusterização usando o algoritmo K-Means.
# Data: 10/10/2023

import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

dataKmeans = pd.read_csv("DADOSCSVPROCESSADOS.csv")
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

dataKmeans["questao"] = pd.to_numeric(dataKmeans["questao"],
    errors='coerce').fillna(0)
dataKmeans["valorDesempenho"] = pd.to_numeric(dataKmeans["valorDesempenho"],
    errors='coerce').fillna(0)

# Incluindo o "id" na clusterização
X = dataKmeans[["id", "valorDesempenho", "conceito"]]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

num_clusters = 7
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(X_scaled)

dataKmeans["Cluster"] = kmeans.labels_
dados_clusterizados = dataKmeans[["id", "conceito",
    "questao", "valorDesempenho", "Cluster"]]
nome_arquivo = f"k_means_idXvalorDesempenhoXconceito"
dados_clusterizados.to_csv(f"{nome_arquivo}.csv", index=False)
```

Fonte: Código desenvolvido no contexto da realização deste trabalho

Figura 4.2 – Código desenvolvido para clusterização de dados a partir dos atributos de *questao X valor-Desempenho*

```
# Autor: Stanley P. M. Silva
# Objetivo: Este programa realiza a clusterização usando o algoritmo K-Means.
# Data: 13/10/2023

import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

dataKmeans = pd.read_csv("DADOSCSVPROCESSADOS.csv")
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

dataKmeans["questao"] = pd.to_numeric(dataKmeans["questao"],
    errors='coerce').fillna(0)
dataKmeans["valorDesempenho"] = pd.to_numeric(dataKmeans["valorDesempenho"],
    errors='coerce').fillna(0)

X = dataKmeans[["questao", "valorDesempenho"]]
num_clusters = 7
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(X)

dataKmeans["Cluster"] = kmeans.labels_
dados_clusterizados = dataKmeans[["id", "conceito", "questao",
    "valorDesempenho", "Cluster"]]
nome_arquivo = f"k_means_questaoXdesempenho"
dados_clusterizados.to_csv(f"{nome_arquivo}.csv", index=False)
```

Fonte: Código desenvolvido no contexto da realização deste trabalho

4.2 Análise: Questões X ValorDesempenho

Como ponto de partida para esta investigação, o primeiro passo foi refletir sobre o processo de ensino e como as estratégias educacionais podem impactar os discentes em seu processo de aprendizado. É comum ouvir afirmações como "faça as todas as questões do *Dredd* e a prova é fácil" como forma de motivação dentro do ambiente de sala de aula. Esse será o ponto de partida para as análises, ou seja, o objetivo é tentar encontrar provas de que os dados das atividades têm referência com a aprovação ou reprovação do aluno.

Num primeiro momento, a primeira investigação realizada sobre os dados clusterizados consistirá na análise da relevância das notas em relação às questões. Portanto, pretende-se investigar quais clusters apresentam uma proporção mais alta de questões respondidas. Essa análise precisa ser compreendida levando em consideração alguns pontos essenciais: em primeiro lugar, os alunos não eram obrigados a responder todas as questões; dentro desse conjunto de questões, diferentes alunos podiam encontrar questões repetidas entre eles, questões com dificuldades semelhantes para o mesmo aluno e até mesmo questões completamente diferentes umas das outras; um número de questões era sorteado entre os alunos, sendo que essas questões foram usadas como avaliação, no entanto, o uso dessas questões por outros alunos para fins de estudo era opcional.

Na Figura 4.3, a análise da saída é representada, ilustrando a quantidade de questões por cluster usando o algoritmo *K-means*. As Figuras 4.4 e 4.5 são as representações das mesmas medidas usando os algoritmos *k-medoids* e *DBSCAN*.

Figura 4.3 – Saída do terminal com dados sobre as quantidades de questões por cluster - Algoritmo *K-means*

```
K-means: Quantidade de questões por Cluster:
Cluster
2    194
0    150
1    105
3     90
6     88
4     71
5     69
```

Fonte: Texto extraído do terminal de saída

Figura 4.4 – Saída do terminal com dados sobre as quantidades de questões por cluster - Algoritmo *K-medoids*

```
K-medoids: Quantidade de questões por Cluster:
Cluster
4    183
6    151
1    105
2    100
3     84
5     82
0     62
```

Fonte: Texto extraído do terminal de saída

Figura 4.5 – Saída do terminal com dados sobre as quantidades de questões por cluster - Algoritmo *DBSCAN*

```
    DBSCAN: Quantidade de questões por Cluster:
Cluster
-1     357
 0      66
 12     45
 11     38
 9      25
 5      22
 1      19
 2      18
 13     16
 3      13
 10     12
 6      12
 14     12
 19     10
 15     10
 16     10
 4       9
 8       8
 24      8
 18      8
 7       6
 23      6
 17      6
 25      6
 20      5
 21      5
 22      5
 26      5
 27      5
```

Fonte: Texto extraído do terminal de saída

A análise subsequente foi conduzida com o propósito de identificar clusters em relação ao desempenho dos alunos e suas correlações com os estados de "aprovação" e "reprovação", tal informação é caracterizada no atributo `conceito` com o valor de "True" ou "False"(exemplo pode ser visto na Figura 4.6). A abordagem adotada visa compreender se existem informações passíveis de análise para avaliar o impacto das variáveis em questão no desempenho final da avaliação. As Figuras 4.6, 4.7, e 4.8 apresentam os resultados das análises utilizando cada algoritmo, demandando uma interpretação meticulosa.

Cumpre salientar que a soma das reprovações em cada cluster resulta em valores discrepantes que não condizem com a realidade das reprovações observadas. Nesse contexto, a intenção é observar se existe alguma relação entre as variáveis referentes ao total de questões respondidas e seu valor de desempenho em conjunto com a aprovação ou reprovação do aluno.

Este enfoque metodológico busca não apenas identificar padrões nos dados, mas também discernir a natureza das relações entre as variáveis investigadas, especialmente aquelas relacionadas aos resultados de aprovação e reprovação. A interpretação cuidadosa desses resultados é crucial para extrair significativas informações no contexto de previsão de desempenho.

Figura 4.6 – Saída do terminal com dados sobre as quantidades de aprovações e reprovações por questão - Algoritmo *K-means*

```
K-means: Soma dos conceitos por Cluster:
conceito  False  True
Cluster
0            42    108
1            17     88
2            43    151
3             8     82
4            12     59
5            12     57
6            22     66
```

Fonte: Texto extraído do terminal de saída

Figura 4.7 – Saída do terminal com dados sobre as quantidades de aprovações e reprovações por questão - Algoritmo *K-medoids*

```
K-medoids: Soma dos conceitos por Cluster:
conceito  False  True
Cluster
0            12     50
1            17     88
2            24     76
3             6     78
4            41    142
5            14     68
6            42    109
```

Fonte: Texto extraído do terminal de saída

Figura 4.8 – Saída do terminal com dados sobre as quantidades de aprovações e reprovações por questão - Algoritmo *DBSCAN*

```
DBSCAN: Soma do conceito por Cluster:
conceito  False  True
Cluster
-1           76    281
 0           17     49
 1            4     15
 2            6     12
 3            3     10
 4            1      8
 5            4     18
 6            8      4
 7            4      2
 8            0      8
 9            3     22
 10           0     12
 11           6     32
 12           2     43
 13           0     16
 14           2     10
 15           2      8
 16           0     10
 17           4      2
 18           4      4
 19           6      4
 20           2      3
 21           0      5
 22           1      4
 23           0      6
 24           0      8
 25           1      5
 26           0      5
 27           0      5
```

Fonte: Texto extraído do terminal de saída

Esta observação visa quantificar se as questões, considerando esses alunos e o contexto específico, contribuíram positivamente para o processo de aprendizado. É crucial ressaltar que o critério de reprovação utilizado reflete o resultado ao final do período correspondente à avaliação "P1", compreendendo um intervalo médio de três semanas.

É evidente que esse resultado apresenta limitações, uma vez que há lacunas quanto à natureza intrínseca das próprias questões. Questões fundamentais, como se são abertas ou fechadas, se foram realizadas ou se eram obrigatórias, permanecem sem resposta nesta base de dados proveniente do *Dredd*. No entanto, essa abordagem revela potencial, especialmente considerando a possibilidade de implementação no *Vesperto*. Uma caracterização mais detalhada dessa base poderia aprimorar significativamente o mapeamento das questões, tornando a análise mais promissora.

Não obstante essas limitações, os resultados obtidos permanecem promissores. Ao analisarmos as figuras 4.3 e 4.6, é possível observar a seguinte correlação: os clusters (0, 1, 2 e 3), que apresentam a maior quantidade de questões respondidas, também exibem os maiores índices de aprovação (108, 88, 151 e 82, respectivamente). Isso permite concluir que há uma correlação entre o engajamento na resolução de questões e a taxa de aprovação.

A carência de informações mencionada anteriormente também destaca aspectos relevantes nos resultados associados ao algoritmo *DBSCAN* (conferir Figuras 4.5 e 4.8). A complexidade inerente a esse algoritmo sugere uma estrutura mais intrincada e um grau de dificuldade mais elevado, os quais poderiam ser mais profundamente explorados mediante a disponibilidade de dados adicionais que respondam a essas questões. A obtenção de informações mais detalhadas, portanto, teria o potencial de proporcionar uma interpretação mais refinada dos dados filtrados.

O último conjunto de dados com potencial para análise é o "valorDesempenho", associado ao conceito de aprovação ou reprovação. Ao utilizar os valores mínimo e máximo, obtemos, respectivamente, os valores `0.00000` e `167.14472`, como pode ser observado nas três figuras (Imagens 4.9, 4.10 e 4.11). Esses valores correspondem ao percentual mínimo e máximo de pontos da questão que varia de 0 a 100%.

Ressalta-se que, de acordo com as normas institucionais, a aprovação do aluno é determinada ao atingir 60% do valor total da nota na atividade. A nota da atividade foi previamente ajustada multiplicando a média de tentativas pela pontuação obtida na correção do *Dredd*. Nesse contexto, foi necessário calcular a média correspondente ao "valorDesempenho", a qual equi-

vale ao total (100% da nota da questão) multiplicado por 0.6 (representando 60%):

$$167.14472 \times 0.6 = 100.28683$$

Com a média calculada, surge a necessidade de investigar a média proporcional presente em cada cluster, obtido por meio da agrupação dos dados de "questões"e "valorDesempenho". Compreender os limites de valores dentro desses clusters torna-se uma análise potencialmente relevante. O foco agora reside na tentativa de elucidar se existe influência das atividades realizadas na percepção do aluno como preparação para a primeira prova.

Como evidenciado nas Figuras 4.9, 4.10 e 4.11, apresentam-se os clusters e seus intervalos, fundamentados nos valores do impacto dessas questões na prática dos discentes.

Figura 4.9 – Saída do terminal com os intervalos de valorDesempenho de cada cluster - Algoritmo*K-means*

```
K-means: Valores maximos e minimos:
                min        max
Cluster
0           0.00000   20.05737
1          93.60104  117.66988
2         150.43025  167.14472
3          47.97053   68.52934
4         120.34420  147.58879
5          70.86936   91.92960
6          21.72881   46.80052
```

Fonte: Texto extraído do terminal de saída

Figura 4.10 – Saída do terminal com os intervalos de valorDesempenho de cada Cluster - Algoritmo*K-medoids*

```
K-medoids: Valores maximos e minimos:
                min         max
Cluster
0          73.54368   91.92960
1          93.60104  117.66988
2          22.56454   50.14342
3          51.81486   72.54081
4         152.10169  167.14472
5         120.34420  150.43025
6           0.00000   21.72881
```

Fonte: Texto extraído do terminal de saída

Figura 4.11 – Saída do terminal com os intervalos de valorDesempenho de cada cluster- Algoritmo*DBSCANM*

```
DBSCAN: Valores maximos e minimos:
                min         max
Cluster
-1          0.00000  167.14472
 0        167.14472  167.14472
 1        167.14472  167.14472
 2          0.00000    0.00000
 3        167.14472  167.14472
 4        142.07301  142.07301
 5        167.14472  167.14472
 6        112.65554  113.15698
 7          0.00000    0.00000
 8         73.54368   73.54368
 9          5.01434    5.01434
 10       122.01565  122.01565
 11       103.62973  103.62973
 12        62.67927   62.67927
 13       113.15698  113.15698
 14        40.44902   40.44902
 15       136.55724  136.55724
 16         4.84720    5.01434
 17        74.37940   74.37940
 18        83.07093   83.07093
 19         0.00000    0.00000
 20        39.11186   39.27901
 21        72.37366   72.54081
 22         5.68292    5.85007
 23       158.78748  158.78748
 24       167.14472  167.14472
 25       158.78748  158.78748
 26       158.78748  158.78748
 27       150.43025  150.43025
```

Fonte: Texto extraído do terminal de saída

Agora, ao consolidar as três informações obtidas e correlacionar os clusters, os conceitos e os valores de desempenho, surge uma observação crucial que contradiz as análises anteriores. Notavelmente, os clusters 0 e 3, com maiores índices de aprovação, apresentam valores mínimos e máximos (0.00000 e 20.05737 para o Cluster 0; 47.97053 e 68.52934 para o Cluster 3) abaixo da média.

Refinando a análise com base na Figura 4.6, destaca-se que os clusters 0 e 3 têm taxas de reprovação de 42 (a segunda maior taxa) e 8 (a menor taxa), respectivamente.

Para uma compreensão mais aprofundada, as Figuras 4.12 e 4.15 (geradas com a biblioteca *matplotlib.pyplot* do Python, assim como as Figuras 4.16 e 4.17) revelam a presença de repetições nos números referentes às questões, especialmente ao interagir com o valor de desempenho. Esse tipo de resultado ocorreu devido à deficiência do *Dredd* em organizar corretamente as questões das atividades e a qual atividade está relacionada.

Vale notar que as Figuras 4.13 e 4.16 mostram um comportamento semelhante para o algoritmo *K-medoids*. No contexto do *DBSCAN*, as Figuras 4.14 e 4.17 apresentam informações coesas, apesar da complexidade visual do gráfico.

No conjunto de dados da Imagem 4.14, a análise comparativa entre o *DBSCAN* e o *K-means* revela um potencial superior do *DBSCAN*. Os dados agrupados por meio desse algoritmo exibiram maior coesão, menor dispersão e menor repetitividade. Destaca-se que, devido à natureza do *DBSCAN*, os dados podem incluir valores com significativo potencial. Entretanto, é crucial não ignorar o fato de que a implementação do *DBSCAN* é mais adequada para conjuntos de dados extensos, o que diverge do contexto atual.

Em relação ao algoritmo do *K-medoids*, os dados extraídos (Figuras 4.4, 4.7, 4.10 e 4.13) os agrupamentos obtidos foram bem parecidos com os do *K-means*, o que não acrescentou muito no contexto deste trabalho.

Figura 4.12 – Saída do terminal com as listas das respectivas questões agrupadas em cada Cluster pelo *K-means*

```
K-means: Valores de 'questao' contidos em cada cluster:
Cluster
0    [2, 3, 4, 3, 3, 3, 3, 1, 4, 1, 1, 1, 2, 2, 2, ...
1    [1, 1, 2, 1, 1, 3, 3, 1, 2, 2, 3, 2, 2, 2, 2, ...
2    [1, 2, 3, 2, 4, 4, 4, 3, 1, 1, 1, 2, 2, 2, 2, ...
3    [3, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, ...
4    [2, 3, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, ...
5    [2, 3, 3, 3, 4, 3, 2, 3, 2, 2, 2, 1, 1, 2, 2, ...
6    [4, 4, 4, 3, 4, 4, 4, 3, 1, 1, 1, 1, 1, 1, 1, ...
```

Fonte: Texto extraído do terminal de saída

Figura 4.13 – Saída do terminal com as listas das respectivas questões agrupadas em cada Cluster pelo *K-medoids*

```
K-medoids: Valores de 'questao' contidos em cada cluster:
Cluster
0    [2, 3, 3, 3, 4, 3, 2, 3, 2, 2, 2, 1, 1, 2, 2, ...
1    [1, 1, 2, 1, 1, 3, 3, 1, 2, 2, 3, 2, 2, 2, 2, ...
2    [4, 4, 4, 3, 4, 4, 4, 1, 1, 1, 1, 1, 1, 1, 1, ...
3    [3, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, ...
4    [1, 2, 3, 2, 4, 4, 3, 1, 1, 1, 2, 2, 2, 2, 4, ...
5    [2, 3, 4, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, ...
6    [2, 3, 4, 3, 3, 3, 3, 1, 4, 3, 1, 1, 1, 2, 2, ...
```

Fonte: Texto extraído do terminal de saída

Figura 4.14 – Saída do terminal com as listas das respectivas questões agrupadas em cada Cluster pelo *DBSCAN*

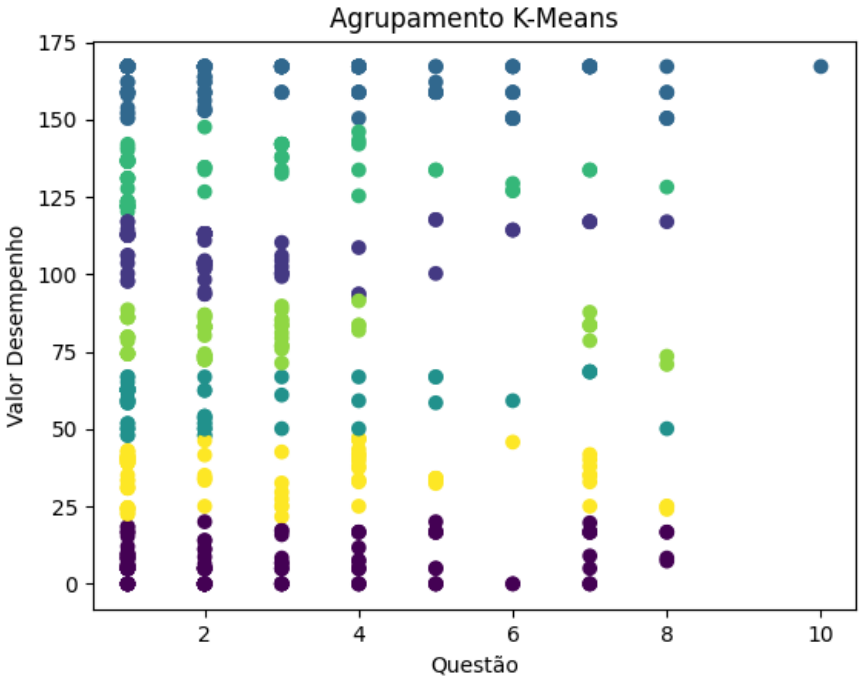
```
DBSCAN: Valores de 'questao' contidos em cada cluster:
Cluster
-1     [2, 4, 4, 3, 1, 1, 1, 2, 1, 1, 2, 2, 2, 4, 2, ...
 0     [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
 1     [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ...
 2     [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ...
 3               [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]
 4                           [3, 3, 3, 3, 3, 3, 3, 3, 3]
 5     [4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, ...
 6                  [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
 7                                    [3, 3, 3, 3, 3, 3]
 8                                [2, 2, 2, 2, 2, 2, 2, 2]
 9     [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
 10                 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
 11    [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ...
 12    [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
 13     [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
 14                 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
 15                       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
 16                       [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
 17                                    [1, 1, 1, 1, 1, 1]
 18                             [2, 2, 2, 2, 2, 2, 2, 2, 2]
 19                       [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
 20                                       [1, 1, 1, 1, 1]
 21                                       [2, 2, 2, 2, 2]
 22                                       [1, 1, 1, 1, 1]
 23                                    [1, 1, 1, 1, 1, 1]
 24                                [7, 7, 7, 7, 7, 7, 7, 7]
 25                                   [5, 5, 5, 5, 5, 5, 5]
 26                                       [4, 4, 4, 4, 4]
 27                                       [6, 6, 6, 6, 6]
```

Fonte: Texto extraído do terminal de saída

Os gráficos a seguir ilustram a relação dos agrupamentos em cada algoritmo. Cada cor representa um cluster, sendo cada ponto a entidade aluno conforme a relação dos atributos de valor de desempenho por questão. As Figuras 4.15 e 4.16 facilitam a análise visual e delineiam os limites de cada cluster em relação ao atributo de valor de desempenho.

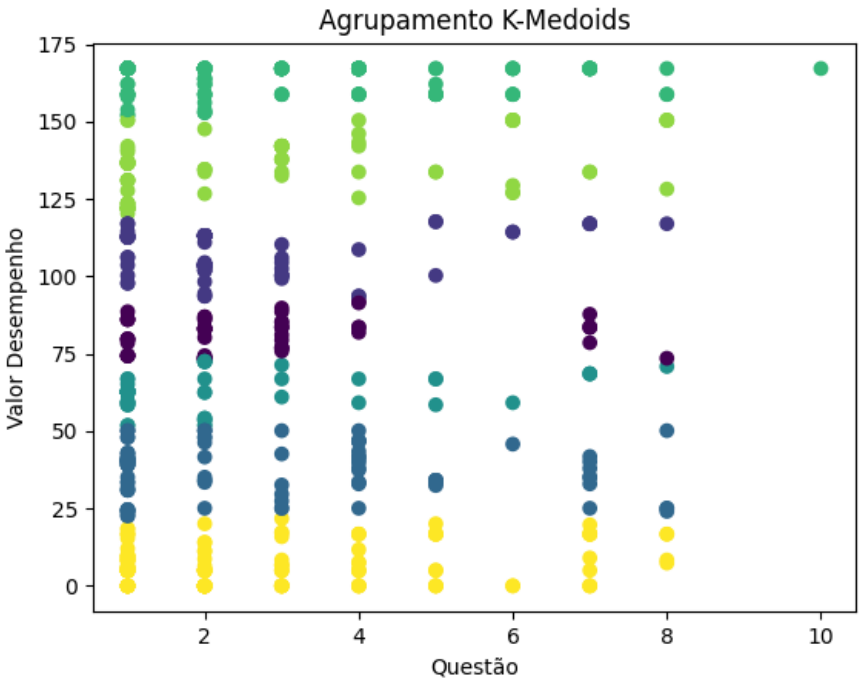
A Figura 4.17, à primeira vista, apresenta-se confusa e não proporciona uma representação visual eficaz. Além disso, ela gera uma análise mais opaca dos dados, obscurecendo o potencial de agrupamento já mencionado do algoritmo DBSCAN. Em outras palavras, essa representação gráfica não é uma escolha adequada para este algoritmo e nestas condições de trabalho.

Figura 4.15 – Representação ValorDesempenho por Questão por dispersão do *K-means*.



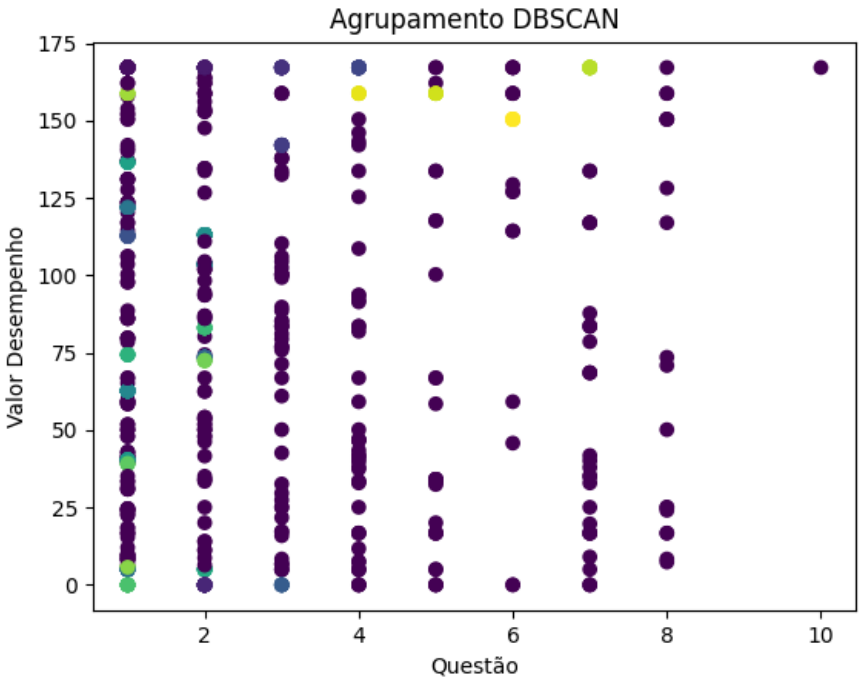
Fonte: Gerada pelo autor utilizando o matplotlib.pyplot

Figura 4.16 – Representação ValorDesempenho por Questão por dispersão *K-medoids*.



Fonte: Gerada pelo autor utilizando o matplotlib.pyplot

Figura 4.17 – Representação ValorDesempenho por Questão por dispersão *DBSCAN*.



Fonte: Gerada pelo autor utilizando o matplotlib.pyplot

É crucial destacar, no entanto, que um mesmo discente pode ser agrupado várias vezes no mesmo conjunto, introduzindo nuances na análise das reprovações. Para esclarecer melhor esse efeito, é possível observar, na Figura 4.12, que o número da questão é repetido diversas vezes. Ao analisar a estrutura do objeto "Aluno", cada elemento é composto pelos seguintes atributos: `id`, `questao`, `tentativa`, `valorDesempenho`, `conceito` e `Cluster`.

Os atributos `id` e `conceito` estão relacionados ao aluno, enquanto os atributos `questao` e `valorDesempenho` representam dados relativos à questão. Utilizando a ferramenta Libcalc do LibreOffice, pode-se obter o exemplo da Tabela 4.1, onde se nota, nas linhas [2, 4, 11, 13], que o mesmo aluno, com conceito reprovado, foi adicionado ao cluster 6.

Comportamentos como a colaboração entre estudantes ou a partilha de respostas também impactam diretamente nos resultados, proporcionando uma justificativa para tais disparidades. Na mesma Tabela 4.1, observa-se, nas linhas [7, 8, 9, 10], que o referido aluno obteve desempenho máximo em questões distintas – `valorDesempenho` igual a 167.14472. Este problema em referenciar cada uma das questões no sistema unicamente pelo número de ordem na lista (questão 1, questão 2, etc.), além deste tipo de impacto, também impossibilita a extração de conclusões mais abrangentes, tais como qual o tema da questão com menores notas, por exemplo.

De todo modo, esses problemas impactam quando se tenta detalhar mais precisamente o perfil do aluno, mas, de modo geral, demonstraram ser muito produtivos para uma análise inicial.

Quadro 4.1 – Exemplo da entidade Aluno depois de realizado o agrupamento

linha	id	questao	tentativa	valorDesempenho	conceito	Cluster
1	251023787	4	1	150.43025	False	1
2	251023787	1	1	74.3794	False	6
3	251023787	2	1	0.0	False	0
4	251023787	1	2	74.3794	False	6
5	251023787	2	2	103.62973	False	2
6	251023787	2	3	103.62973	False	2
7	251023787	1	1	167.14472	False	1
8	251023787	1	1	167.14472	False	1
9	251023787	4	1	167.14472	False	1
10	251023787	3	1	167.14472	False	1
11	251023787	1	1	74.3794	False	6
12	251023787	2	1	0.0	False	0
13	251023787	1	2	74.3794	False	6
14	251023787	2	2	103.62973	False	2
15	251023787	2	3	103.62973	False	2
16	251023787	1	1	0.0	False	0
17	251023787	5	1	0.0	False	0
18	251023787	1	1	0.0	False	0
19	251023787	1	2	16.71447	False	0
20	251023787	7	1	16.71447	False	0

Fonte: Amostra dos resultados obtidos do arquivo k_means_valorDesempenhoXconceito.csv gerado pelo algoritmo *K-means* e filtrado pelo *Libre Office Calc*.

4.3 Análise: IDs X Conceitos X ValorDesempenho

Esta seção tem como propósito apresentar os resultados com base nas entidades alunos. Aqui, procuramos investigar se há uma relação entre os desempenhos das personas e os conceitos, ou seja, se os alunos aprovados na primeira prova (P1) apresentaram um desempenho acima da média previamente calculada e estimada em **100.28683**. Essa análise foi restrita ao uso do algoritmo *k-means*, justificada pela significativa redução de informações geradas nesta etapa ao considerar o identificador único do aluno nos filtros. Como indicado na subseção 2.3.3, o *DBSCAN* não é recomendado para análises com baixo volume de dados. Além disso, conforme mencionado na seção anterior (4.2), o *K-medoids* não foi utilizado devido ao seu comportamento muito semelhante, o que não acrescentaria análises relevantes ao trabalho.

Ao buscar a quantidade de "aprovação"e "reprovação"com base na quantidade de personas, representadas aqui pelo ID, após filtragem e clusterização, obtivemos 53 personas discentes nos agrupamentos. Em outras palavras, foram analisadas informações de 53 perfis comportamentais de alunos que puderam ser extraídas do *Dredd*.

Na Figura 4.18, é possível observar a distribuição da quantidade de dados em cada cluster, enquanto na Figura 4.19, são apresentados os totais de aprovações e reprovações. No contexto das personas estudadas, a Figura 4.20 ilustra os valores mínimos e máximos alcançados por essas personas.

Figura 4.18 – Saída do terminal com as listas das respectivas questões agrupadas em cada Cluster pelo k-means

```
K-means: Quantidade de ids por Cluster:
Cluster
2    16
1    13
3    10
4    10
0     4
```

Fonte: Texto extraído do terminal de saída

Figura 4.19 – Saída do terminal com as listas das respectivas questões agrupadas em cada Cluster pelo k-means

```
K-means: Soma do conceito por Cluster:
conceito  False  True
Cluster
0             2     2
1             2    11
2             6    10
3             2     8
4             2     8
```

Fonte: Texto extraído do terminal de saída

Figura 4.20 – Saída do terminal com as listas das respectivas questões agrupadas em cada Cluster pelo k-means

```
K-means: Valores maximos e minimos:
              min        max
Cluster
0        33.42894  167.14472
1         0.00000  167.14472
2         0.00000  167.14472
3         5.85007  167.14472
4         4.84720  167.14472
```

Fonte: Texto extraído do terminal de saída

O método K-means revelou valores mínimos e máximos significativos nos diversos clusters. A Saída 4.20 apresenta os resultados, destacando os extremos mínimos e máximos para cada cluster específico. Ao examinar os dados gerados, ressalta-se a potencial relevância da análise do Cluster 0, no qual foi observada uma equiparação entre os valores de aprovação e reprovação, ao mesmo tempo em que não foram registrados os extremos mínimos. Nesse contexto, o valor mínimo fixado em zero indica que um aluno pode ter obtido uma média devido à falta de participação em atividades ou ter recebido o conceito de abandono, o que inviabiliza a análise nos Clusters 1 e 2. Nos Clusters 3 e 4, registram-se valores de aprovação superiores aos de reprovação; no entanto, tal observação não contribui de maneira significativa para a análise, pois não apresenta padrões comportamentais claros.

Portanto, o potencial do Cluster 0 depende de mais informações para definir padrões comportamentais, não permitindo afirmar se foi uma análise satisfatória ou insatisfatória.

Além disso, ao examinar os padrões presentes nos clusters, observa-se uma prevalência de valores mais elevados em relação ao conceito máximo. Essa tendência sugere que determinados grupos de alunos consistentemente apresentam desempenhos acima da média. Uma análise mais aprofundada desses clusters específicos poderia proporcionar a identificação de características comportamentais ou acadêmicas compartilhadas, fornecendo informações valiosas para estratégias de ensino personalizadas ou intervenções educacionais.

5 CONCLUSÃO E TRABALHOS FUTUROS

O objetivo deste trabalho foi pesquisar e investigar possíveis usos de algoritmos de inteligência artificial como ferramenta para aprimorar a prática didática e promover uma intervenção mais pontual e correta do corpo docente nas atividades pedagógicas voltadas para o ensino de disciplinas com foco em programação e utilização dessas técnicas pelo *Vesperto*.

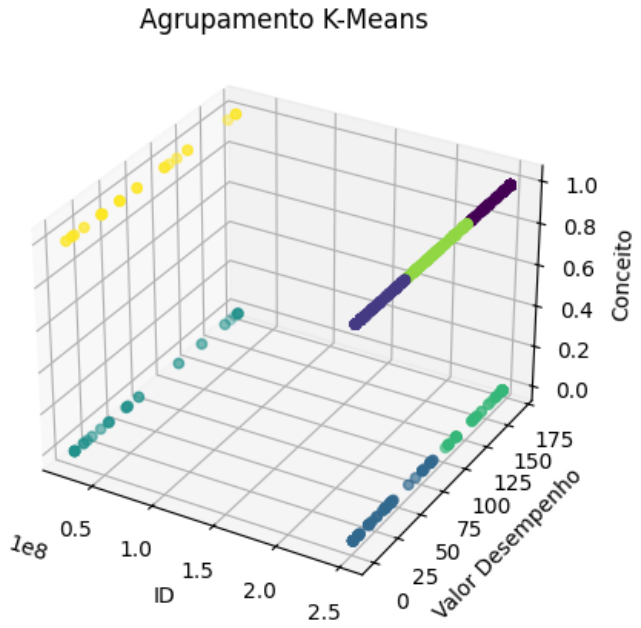
Os resultados obtidos mostraram-se promissores ao evidenciar que é possível obter com certa clareza dados sobre o desempenho dos alunos e mensurar suas atividades acadêmicas, gerando perfis que classificam potenciais deficiências e possíveis reprovações. Também foi interessante comprovar que, com base no agrupamento dos dados, a ideia de que é necessário realizar atividades pedagógicas com foco no treino para uma boa avaliação é um contexto realmente válido, mesmo que ocorram pontos fora da curva que comprovem que a mesma estratégia não funciona para todos os discentes.

Mesmo nos casos de exceções, os algoritmos mostraram potencial ao apontar comportamentos de risco, como diversas questões iguais com os mesmos valores para alunos diferentes, o que permite pensar na possibilidade de troca de respostas entre os alunos, prejudicando obviamente o desenvolvimento individual.

Entretanto, este trabalho também apresentou limitações, principalmente estruturais, o que comprova a necessidade de uma nova ferramenta que possibilite a implementação de algoritmos de inteligência artificial. A incapacidade do *Dredd* de armazenar dados das questões ou permitir questões com metodologias diferentes, por exemplo, é um fator prejudicial a este trabalho, quando na coleta de dados os dados já são incompletos. A capacidade de classificar as questões entre múltipla escolha ou questões de lógica, atividade de diagramação ou programação em pares (*pair programming*) faz com que a atividade de mapeamento do potencial do aluno de acordo com a metodologia aplicada seja impossível dentro do contexto deste trabalho.

Como demonstra Amaral et al. (2021), a possibilidade de mapeamento seria extremamente importante, pois retira do professor o problema de andar às cegas dentro de tantas possibilidades. Como ilustra a imagem 5.1 é possível realizar um estudo baseado em múltiplos rótulos, gerando perfis que permite rastreamento e acompanhamento. Uma melhor descrição da base de dados, com identificadores (todas as entidades possuindo *Primary Keys*) capazes de serem rastreados, permitiria implementar mais recursos e ferramentas de IA, podendo transcender o objetivo primário do *Vesperto* e ajudar até mesmo professores de disciplinas não ligadas à programação.

Figura 5.1 – Representação da dispersão dos dados agrupados.



Fonte: matplotlib.pyplot

Ressalta-se também que a própria quantidade de dados foi um fator limitante para este trabalho. Devido à natureza da forma como as atividades são realizadas atualmente, a quantidade de informações processáveis é baixa. Os algoritmos utilizados são os mais populares de agrupamento em aprendizado de máquina, mas podem enfrentar desafios ao lidar com conjuntos de dados pequenos. O K-means e o K-medoids são sensíveis à inicialização dos centróides, o que pode resultar em soluções de agrupamento distintas com escolhas iniciais diferentes. Esses algoritmos assumem que os clusters têm formas esféricas e tamanhos aproximadamente iguais, o que pode não ser adequado para conjuntos de dados pequenos e complexos, nos quais os agrupamentos podem ser irregulares. O DBSCAN pode ser menos eficiente em conjuntos de dados pequenos devido à sensibilidade aos parâmetros, como o raio de busca e o número mínimo de pontos necessários para formar um cluster. Escolhas inadequadas desses parâmetros podem levar a clusters indesejados ou à incapacidade de identificar agrupamentos significativos.

O *Vesperto* tem potencial para ser um sistema mais seguro, moderno e adaptável do que os sistemas atualmente em uso. Contudo, este trabalho demonstra que o desenvolvimento do mesmo precisa passar por análises cuidadosas desses sistemas e entender a origem dos erros presentes, a fim de corrigir, evoluir e permitir não apenas a manutenção no futuro.

O principal fator que pode ser apontado aqui, como características potenciais de análises de dados que forem coletados pelo *Vesperto*, seria a capacidade de mensurar o tempo gasto em cada questão. Isso pode indicar o nível de complexidade do exercício ou do tema de estudo e complementar as informações já obtidas pelo *Dredd*.

Supondo melhorias em cima de erros já existentes, pode-se pensar num contexto onde as avaliações documentadas na seção 4.2 poderiam ser complementadas com algoritmos de Processamento de Linguagem Natural (PLN). O *Dredd* já possui as informações do código do exercício e também devolve determinadas respostas padronizadas. Com a adição desses algoritmos de PLN, o sistema poderia não apenas apontar mais precisamente os erros cometidos, como também personalizar as respostas com sugestões de melhoria no código entregue para correção, através de uma abordagem mais didática aonde desafiaria o aluno a compreender o erro e a corrigi-lo.

Finalmente, a correção do maior problema identificado torna-se imperativa: a carência de informações relativas a cada questão e sua interconexão com as atividades realizadas. A ausência de abstração no contexto didático da questão, assim como na forma como foi empregada dentro da atividade, aliada à falta de dados para monitorar individualmente o desempenho do aluno, não possibilita uma abordagem mais aprimorada. As figuras 4.12, 4.13 e 4.14 destacam essa deficiência atual no sistema *Dredd*. Com uma maior quantidade de dados referentes às questões, seria possível considerar o desenvolvimento de métricas mais precisas para equalizar as notas obtidas ou atribuir maior peso a exercícios obrigatórios.

Essa correção é crucial para aperfeiçoar a capacidade do *Dredd* em avaliar o desempenho dos alunos, proporcionando uma visão mais abrangente e precisa das competências adquiridas em cada questão e sua aplicação nas atividades propostas. Isso, por sua vez, poderia resultar em uma avaliação mais justa e eficiente do progresso dos alunos, contribuindo para uma abordagem mais eficaz no contexto educacional.

Considerando já o estudo para melhorias é imperativo que o banco de dados tenha um desenho que aborde uma abstração mais detalhista sobre o armazenamento das questões e das atividades e uma Interface de Programação de Aplicação (API) que permita a extração desses dados. Este armazenamento detalhista passa por um design que seja capaz de considerar que a entidade "atividade" tem referência forte com a entidade "questão", o que permitiria lastrear os dados analisados. Isto é necessário pois a maior deficiência encontrada no *Vesperto* foi a incapacidade de extrair informações das atividades realizadas.

A conclusão deste trabalho é que os algoritmos de Aprendizado de Máquina podem ser implementados no *Vesperto* e tem um potencial significativo na capacidade de melhoria contínua na didática não apenas de disciplinas com foco em programação, como foi evidenciado nas análises realizadas (Cap. 4), mas também em disciplinas de diversas áreas por ter capacidade de mapear o comportamento dos discentes.

Como recomendação para futuras pesquisas, sugere-se a análise da implementação de algoritmos de Processamento de Linguagem Natural (PLN) sobre os dados obtidos por meio das técnicas aqui descritas. Isso permitiria uma identificação mais precisa das deficiências nos sistemas existentes e indicaria a abordagem mais apropriada para a aplicação desses algoritmos de Aprendizado de Máquina no contexto de intervenção pedagógica.

REFERÊNCIAS

AMARAL, G. de S. et al. Um sistema de recomendação de estratégias de aprendizagem baseado no perfil de motivação do aluno: Sisrea. In: SBC. **Anais do XXXII Simpósio Brasileiro de Informática na Educação**. [S.l.], 2021. p. 718–727.

ARTHUR, D.; VASSILVITSKII, S. K-means++ the advantages of careful seeding. In: **Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms**. [S.l.: s.n.], 2007. p. 1027–1035.

BZUNECK, J. A.; BORUCHOVITCH, E. Motivação e autorregulação da motivação no contexto educativo. **Psicologia Ensino & Formação**, Associação Brasileira de Ensino de Psicologia, v. 7, n. 2, p. 73–84, 2016.

DAM, N. A. K.; DINH, T. L.; MENVIELLE, W. Marketing intelligence from data mining perspective. **International Journal of Innovation, Management and Technology**, v. 10, n. 5, p. 184–190, 2019.

FREIRE, P. **Pedagogia do oprimido**. São Paulo: Paz e Terra, 1974.

GIULIANO. Mineração de dados - data warehouse, data mining, bi e olap. **Revista ClubeDelphi 146**, 2012. Disponível em: <https://www.devmedia.com.br/mineracao-de-dados-data-warehouse-data-mining-bi-e-olap-revista-clubedelphi-146/26537>.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. [S.l.]: Morgan Kaufmann Publishers, 2012.

JAIN, A. K.; DUBES, R. C. **Algorithms for clustering data**. [S.l.]: Prentice-Hall, Inc., 1988.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: A review. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 31, n. 3, p. 264–323, sep 1999. ISSN 0360-0300.

OLIVEIRA, M. de S. Metodologia de seleção de features não-supervisionada para clustering em conjunto de dados de alta dimensionalidade. **Trabalho de Conclusão de Curso, Universidade Federal de Pernambuco**, 2018.

PEREIRA, F. D. et al. Predição de desempenho em ambientes computacionais para turmas de programação: um mapeamento sistemático da literatura. In: SBC. **Anais do XXXI Simpósio Brasileiro de Informática na Educação**. [S.l.], 2020. p. 1673–1682.

REDDY, C. K.; AGGARWAL, C. C. **Healthcare data analytics**. [S.l.]: CRC Press, 2015. v. 36.

SANTOS, I. N. Políticas públicas de inclusão digital: o caso do programa educação conectada em lavras mg. Universidade Federal de Lavras, 2020.

SOUZA, C. M. de. Visualg-ferramenta de apoio ao ensino de programação. **Revista Eletrônica TECCEN**, v. 2, n. 2, p. 01–09, 2009.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining: Practical Machine Learning Tools and Techniques**. 3. ed. [S.l.]: Morgan Kaufmann, 2011. ISBN 978-0-12-374856-0.