CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA

Fiscalização de Compras Públicas: Uma Abordagem com Processamento de Linguagem Natural

Vinicius Gonçalves Paraizo Borges

Prof. Orientadores:
Eduardo Bezerra, D.Sc.
Wellington Souza Amaral, M.Sc.

Rio de Janeiro,
2025

CENTRO FEDERAL DE EDUCACAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA

Fiscalização de Compras Públicas: Uma Abordagem com Processamento de Linguagem Natural

Vinicius Gonçalves Paraizo Borges

Projeto final apresentado em cumprimento às normas do Departamento de Educação Superior do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação.

Prof. Orientadores:
Eduardo Bezerra, D.Sc.
Wellington Souza Amaral, M.Sc.

Rio de Janeiro,
2025

DEDICATÓRIA

Dedico este trabalho aos meus pais, que sempre acreditaram em mim mesmo nos momentos mais difíceis.

AGRADECIMENTOS

Agradeço, primeiramente, à professora Myrna, por sua orientação e incentivo constantes ao longo desta jornada no CEFET. Seu apoio foi essencial para o meu crescimento.

Ao professor Eduardo, meu sincero agradecimento por estar sempre presente, orientando e propondo desafios que me fizeram evoluir. Seu apoio foi essencial.

À minha família, meu porto seguro, agradeço por todo amor, compreensão e incentivo incondicional. Sem vocês, nada disso seria possível.

RESUMO

Este trabalho aborda a análise crítica e a fiscalização de empenhos no Tribunal de Contas do Estado do Rio de Janeiro (TCE/RJ), com foco na otimização da gestão de recursos governamentais por meio de tecnologias de aprendizado de máquina. Observa-se que há desafios significativos relacionados à inconsistência e à falta de padronização nas notas de empenho, o que compromete a fiscalização eficaz e a identificação de padrões. Propõe-se uma metodologia que integra técnicas avançadas de Processamento de Linguagem Natural (PLN) e algoritmos de aprendizado profundo para clusterização, visando superar as limitações de dados fragmentados e incompletos. O estudo é elaborado sobre um modelo capaz de lidar com a clusterização de dados complexos e volumosos, como os dados públicos, proporcionando uma análise mais robusta das relações semânticas entre os registros. O modelo desenvolvido permite uma gestão mais eficiente dos recursos públicos, com base na identificação de padrões e percepções valiosas.

Palavras-chave: Fiscalização de Empenhos, Tribunal de Contas, Processamento de Linguagem Natural, Aprendizado Profundo, Clusterização de Dados, Gestão de Recursos Governamentais, Análise de Dados Públicos.

ABSTRACT

This study addresses the critical analysis and auditing of public invoices at the Court of Accounts of the State of Rio de Janeiro (TCE/RJ), focusing on optimizing the management of government resources through machine learning technologies. It is observed that significant challenges arise from inconsistencies and the lack of standardization in the public spending invoices, which hinder effective oversight and pattern identification. This work proposes an innovative methodology that integrates advanced techniques of Natural Language Processing (NLP) and deep learning algorithms for clustering, aiming to overcome the limitations of fragmented and incomplete data. The study is based on a model capable of handling the clustering of complex and large-scale data, such as public records, providing a more robust analysis of semantic relationships between entries. The developed model enables more efficient management of public resources by identifying patterns and generating valuable insights.

Keywords: Invoices Auditing, Court of Accounts, Natural Language Processing, Deep Learning, Data Clustering, Government Resource Management, Public Data Analysis.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIAÇÕES

AM	Aprendizado De Máquina	13, 14, 16, 21
AP	Aprendizado Profundo	14, 34, 37, 39, 59
DEC	Deep Embedded Clustering	15, 29, 33, 35, 36, 37, 38, 39, 40, 41, 59, 60, 68, 71
DNNS	Deep Neural Networks	24, 26, 29
MSE	Erro Médio Quadrático	31, 66
PLN	Processamento De Linguagem Natural	15, 16, 17, 21, 24, 29
SAE	Autoencoders Empilhados	29, 60
SBERT	*Sentence BERT*	24, 58
SGD	*Stochastic Gradient Descent*	25
TCE/RJ	Tribunal De Contas Do Estado Do Rio De Janeiro	11, 12, 14, 17, 20, 41, 77

Capítulo 1

Introdução

1.1 Contextualização

A transparência e a eficácia na fiscalização das despesas públicas são elementos essenciais para garantir a boa gestão dos recursos governamentais. No contexto do Tribunal de Contas do Estado do Rio de Janeiro (TCE/RJ), a análise de empenhos é um passo crítico para entender como os recursos são utilizados pelos jurisdicionados, abrangendo diversos órgãos e entes municipais.

Embora a legislação permita que certos bens de pronta entrega sejam adquiridos sem a formalização de contratos, a natureza dessas transações ainda mantém uma relação semântica que pode ser identificada e analisada. Essa conexão semântica decorre de padrões recorrentes nas descrições dos empenhos, nos fornecedores envolvidos, nos valores e frequências das aquisições, além do contexto orçamentário em que estão inseridos. Por exemplo, compras regulares de gêneros alimentícios para merenda escolar, mesmo que fragmentadas ao longo do tempo e realizadas sem contrato formal, compartilham características que indicam um propósito comum e um vínculo lógico com outras despesas similares.

Além disso, a ausência de um contrato associado não impede que determinados grupos de registros de Itens de Empenhos apresentem indícios de planejamento conjunto ou de dependência funcional entre si. O uso de técnicas avançadas de análise, como aprendizado de máquina e processamento de linguagem natural, permite identificar essas correlações implícitas, agrupando empenhos por similaridade semântica e auxiliando na detecção de possíveis estratégias de fracionamento indevido. Dessa forma, é possível aprimorar os mecanismos de controle e fiscalização, garantindo maior transparência na execução orçamentária e mitigando riscos associados à fragmentação intencional de despesas públicas.

Logo, identificar e agrupar essas relações de forma sistemática e inteligente é essencial para uma gestão pública eficiente. Essa abordagem permite obter uma visão consolidada das despesas públicas e identificar possíveis fragmentações de empenhos, que poderiam passar despercebidas em análises tradicionais, comprometendo a transparência e a eficácia na fiscalização

dos recursos.

1.2 Motivação

O desafio desta pesquisa reside na necessidade de identificar padrões e possíveis agrupamentos das despesas, levando em consideração a complexidade e o grande volume de dados disponíveis. O estudo fundamenta-se em uma base de dados do TCE/RJ, que contém informações essenciais sobre os Itens de Empenho de Jurisdicionados das cidades do Estado do Rio de Janeiro, exceto a capital. Dentre os campos relevantes desta análise, destacam-se:

- `IdContrato`: Identificador do contrato associado ao empenho, quando existente, permitindo a vinculação de despesas a um instrumento formalizado;
- `Histórico`: Descrição detalhada da despesa, contendo informações sobre a natureza da aquisição e sua justificativa;
- `Elemento da Despesa do TCE`: Classificação contábil que categoriza a despesa, facilitando sua identificação e análise;
- `Unidade`: Órgão ou entidade responsável pela realização do empenho;
- `Credor`: Fornecedor ou prestador de serviço beneficiado pelo pagamento.
- `Vlr_Empenhado`: Valor reservado no orçamento para cobrir uma despesa pública prevista.

O agrupamento desses empenhos é essencial para identificar padrões de gastos e detectar possíveis irregularidades. Em situações ideais, um serviço contratado é precedido por um contrato formal, com empenhos vinculados diretamente ao seu respectivo contrato. No entanto, há alguns cenários em que essa relação não está claramente definida, tornando a análise mais complexa.

Logo, destacam-se os desafios principais deste estudo:

- **Empenhos sem contrato associado para a mesma prestação de serviço**
  - Embora as despesas associadas a um mesmo tipo de prestação de serviço geralmente compartilhem características comuns, como a recorrência do mesmo fornecedor ou credor, descrições similares no campo `Histórico` e a correspondência

com o mesmo elemento de despesa, observa-se que muitas delas não possuem um contrato formalmente registrado.

- **Empenhos de um mesmo tipo de serviço, mas com inconsistências nos registros**
  - Em casos como folha de pagamento, a estrutura dos empenhos pode variar devido à forma como o credor é registrado. Em alguns registros, utiliza-se o CPF de um funcionário listado, enquanto em outros é informado o CNPJ da unidade administrativa, dificultando a identificação dos empenhos que pertencem ao mesmo grupo de despesas.
- **Empenhos referentes à mesma licitação, porém em lotes diferentes**
  - Em compras governamentais, um mesmo processo licitatório pode ser dividido em lotes, permitindo que credores diferentes sejam vencedoras de diferentes segmentos da mesma licitação. Isso é comum na aquisição de medicamentos, onde o `Histórico` dos empenhos pode indicar a mesma licitação e o mesmo ente administrativo, mas com credores distintos.

Além dessas dificuldades, a falta de um campo padronizado que correlacione empenhos semelhantes exige abordagens avançadas para garantir uma identificação precisa e a correta associação das despesas. Dessa forma, é necessário desenvolver métodos que permitam a identificação automática dessas relações ocultas, possibilitando ao controle externo uma visão consolidada das despesas e aumentando a eficiência da fiscalização pública.

A relevância deste tema é evidente tanto do ponto de vista acadêmico quanto prático. No campo acadêmico, ele impulsiona o desenvolvimento de métodos avançados de Aprendizado de Máquina (AM) voltados para aplicações no setor público, destacando o potencial da inteligência artificial para resolver problemas complexos relacionados à fiscalização e gestão de recursos. Além disso, o grande volume de dados públicos e a necessidade de processamento em alta velocidade demandam abordagens computacionalmente eficientes e escaláveis, incentivando o desenvolvimento de técnicas mais sofisticadas e eficazes.

Na esfera prática, os benefícios incluem maior eficácia na alocação de recursos, prevenção de desperdícios e aumento da transparência pública. Esses avanços promovem uma administração mais responsável e comprometida com os princípios democráticos, fortalecendo a confiança entre os cidadãos e o poder público.

1.3 Objetivos

O objetivo primário deste trabalho é aprimorar e facilitar a fiscalização das contas públicas, tornando-a mais precisa, automatizada e escalável, com menor dependência de inspeções manuais. Para isso, utilizam-se técnicas de AM, com ênfase em Aprendizado Profundo (AP) e agrupamento de dados (*clusterização*), a fim de agrupar registros de forma sistemática e eficiente.

Mais especificamente, este estudo propõe a aplicação de técnicas de agrupamento em grande escala sobre os itens de empenho da base de dados do TCE/RJ, com o intuito de identificar automaticamente padrões, grupos e subgrupos de comportamento ou natureza similar entre os registros analisados. Para isso, os seguintes objetivos secundários foram estabelecidos:

- Realizar o pré-processamento da base de dados de itens de empenho, incluindo a transformação do campo textual `Histórico`, que contém descrições detalhadas dos empenhos, em *embeddings* semânticos. Além disso, codificar os campos `IdContrato`, `Histórico`, `Unidade`, `ElemDespesaTCE` e `Credor`, garantindo que suas informações sejam preservadas e adequadas para a etapa de processamento, o processo de agrupamento de dados.

- Desenvolver e implementar um algoritmo de agrupamento de dados eficiente, capaz de processar o grande volume de dados de alta dimensionalidade, proporcionando agrupamentos precisos e uma visão consolidada e estruturada dos itens de empenho.

- Propor e aplicar métodos para avaliar a qualidade dos agrupamentos gerados, utilizando métricas apropriadas e adaptadas ao contexto de despesas públicas. O objetivo é identificar os modelos e métodos que oferecem os melhores resultados e garantir a robustez da abordagem proposta.

Adicionalmente, este trabalho enfatiza a importância de garantir uma análise transparente e interpretável. A interpretabilidade do modelo é crucial, especialmente em aplicações financeiras públicas, onde a confiança e os valores democráticos estão em jogo. Assim, assegurar que o modelo de agrupamento de dados seja compreensível e transparente é um fator essencial para promover uma gestão pública mais responsável e transparente.

1.4 Metodologia

Esta seção descreve a metodologia adotada neste trabalho, a qual combina técnicas avançadas de Processamento de Linguagem Natural (PLN) para o pré-processamento dos campos de texto com abordagens de aprendizado profundo para o agrupamento dos itens de empenho.

O processo de agrupamento dos itens de empenho foi realizado por meio do algoritmo não-supervisionado Deep Embedded Clustering (DEC), proposto por Xie et al. [2016], que utiliza os campos `IdContrato`, `Historico`, `Unidade`, `elemDespesaTCE` e `Credor` como variáveis de entrada.

A seleção dessas variáveis foi baseada em sua relevância para a identificação e agrupamento dos itens de empenho conforme o tipo de serviço prestado. Além disso, a escolha de um subconjunto específico de atributos visa garantir a eficiência computacional do algoritmo, evitando sobrecarga no processamento.

No entanto, antes de serem utilizados pelo modelo de agrupamento de dados, os campos foram passados por um processo de pré-processamento para garantir sua adequada integração no modelo. A transformação de dados textuais em representações numéricas foi um passo essencial nesse processo.

Para isso, é essencial aplicar técnicas de PLN no campo textual `Histórico` para gerar *embeddings*. Estes são representações numéricas em um espaço vetorial, onde palavras e frases semanticamente similares são mapeadas para vetores próximos uns dos outros. Esse processo foi realizado utilizando um modelo de PLN pré-treinado, isto é, que já foi treinado em grandes volumes de dados textuais previamente. Consequentemente, ao utilizar o modelo, os textos serão processados e convertidos em vetores que capturam tanto o significado individual das palavras quanto o contexto em que estão inseridas. Essa abordagem resultou em um agrupamento mais preciso, graças às representações semânticas estruturadas e ricas geradas a partir do texto.

Desta forma, a abordagem proposta permite que os agrupamentos gerados reflitam de maneira mais robusta e precisa as relações subjacentes entre os itens de empenho. Ao integrar diversas fontes de informação e transformar os dados em representações numéricas adequadas, o modelo proporciona uma análise mais aprofundada e consistente das despesas públicas, permitindo uma visão mais clara e confiável dos padrões dos itens de empenho.

Por fim, a métrica de avaliação adotada foi o *Silhouette Score*, que avalia a coerência interna dos agrupamentos gerados, medindo o quão semelhantes os dados são em um mesmo grupo

em comparação com os demais. A escolha do número k de grupos foi realizada por meio do algoritmo de agrupamento de dados *KMeans*, testando-se um intervalo de valores para k e selecionando aquele que obteve o maior valor médio de *Silhouette Score*.

1.5 Organização dos Capítulos

Além desta introdução, o trabalho está organizado em mais cinco capítulos e um apêndice. O Capítulo 2 apresenta a fundamentação teórica necessária para a compreensão do tema, incluindo conceitos relacionados à computação, como AM e PLN e técnicas de agrupamento de dados. O Capítulo 3 discute os trabalhos correlatos à proposta deste estudo, destacando pesquisas anteriores que aplicam técnicas semelhantes na análise de dados públicos. O Capítulo 4 detalha o desenvolvimento do algoritmo proposto, incluindo a descrição do conjunto de dados, os pré-processamentos realizados, as arquiteturas de modelos utilizadas e os critérios de avaliação definidos. No Capítulo 5, são apresentados os resultados das avaliações experimentais realizadas para validar a abordagem proposta. Esse capítulo discute os agrupamentos obtidos, a qualidade dos grupos e as análises realizadas com base nos resultados. O Capítulo 6 apresenta as considerações finais do trabalho, destacando as contribuições obtidas até o momento, apontando os desafios enfrentados durante o desenvolvimento da solução, bem como os próximos passos a serem seguidos. O Apêndice A apresenta o desenvolvimento de uma aplicação web construída como complemento a esta pesquisa, cujo objetivo é facilitar a consulta e exploração dos itens de empenho por meio de uma interface interativa e mecanismos de busca semântica.

Capítulo 2

Fundamentação Teórica

Neste capítulo vamos nos aprofundar nas definições dos tópicos importantes para a pesquisa. Começando com os conceitos financeiros, na Seção 2.1, onde foi abordada a definição de conceitos relevantes ao tema e o funcionamento desse documento no contexto da gestão pública. Depois, passamos para os tópicos de computação, explicando conceitos básicos de *embeddings* de dados utilizando PLN na Seção 2.2 e conceitos fundamentais sobre *Autoencoders* na Seção 2.3, sobre Redes Neurais Profundas. Existem conceitos importantes sobre o algoritmo utilizado neste presente estudo na Seção 2.4, sobre agrupamento de dados. Por fim, conceitos fundamentais sobre a métricas relevantes para este estudo na Seção 2.5

2.1 Itens de Empenho

O conceito central deste estudo são os **Itens de Empenho**, um dos principais instrumentos da gestão orçamentária pública. Seu objetivo é regular as finanças públicas no Brasil, sendo um ato formal emitido por uma autoridade competente, no qual o Estado assume a obrigação de pagamento, seja pendente ou condicionada ao cumprimento de determinada condição. Esse documento oficial, emitido por órgãos públicos, registra o compromisso de despesa com um fornecedor ou prestador de serviço, fazendo parte do processo de execução orçamentária e garantindo a disponibilidade de recursos no orçamento para cobrir a despesa.

O empenho de despesa é um instrumento essencial para assegurar a correta execução orçamentária e financeira, constituindo elemento fundamental para a transparência e o controle das finanças públicas. O TCE/RJ fiscaliza os órgãos públicos do Estado do Rio de Janeiro, com exceção da capital, por meio da análise dos Itens de Empenho, disponibilizados na base de dados de Itens de Empenho em **??**.

O Item de Empenho não apenas formaliza o compromisso do governo com o pagamento de determinada despesa, mas também oferece garantias e previsões, tanto para os gestores públicos quanto para a sociedade, no que se refere à aplicação dos recursos públicos.

2.1.1 Finalidades do Empenho

As finalidades do empenho são múltiplas e essenciais para o bom andamento da administração pública. Entre os principais objetivos do empenho, destacam-se:

- Firmar o compromisso de aquisição e pagamento futuro entre a administração pública e o fornecedor, garantindo que a despesa será cumprida em um momento posterior.

- Justificar a necessidade do gasto. Ele é uma prova de que o gasto é necessário e foi aprovado segundo as necessidades do orçamento e as prioridades da gestão pública. Este processo ajuda a assegurar que os recursos públicos estão sendo utilizados de forma racional e eficiente.

- Identificar o responsável pela aprovação da despesa, o que é fundamental para a transparência e prestação de contas. Esse registro facilita a rastreabilidade das decisões financeiras, permitindo, no contexto do e-Governo, o uso de sistemas eletrônicos como o SIAFI, que proporcionam maior controle, rastreabilidade e auditoria das despesas públicas.

- Garantir que os recursos de determinada classificação orçamentária sejam devidamente alocados às despesas, funcionando como um mecanismo de controle para evitar o uso indevido dos recursos e assegurar o cumprimento do orçamento conforme planejado.

- Assegurar que o crédito disponível seja suficiente para cobrir a despesa, verificando se existem recursos financeiros disponíveis para atender à necessidade de pagamento, evitando o comprometimento de fundos além do que foi autorizado.

- Servir de referência à liquidação da despesa. A liquidação é o reconhecimento formal de que a despesa foi efetivamente realizada e é diretamente vinculada ao empenho, garantindo que o pagamento seja realizado segundo o compromisso assumido.

- Assegurar a validade dos contratos, convênios e ajustes financeiros, garantindo que as despesas estejam devidamente documentadas e autorizadas, o que contribui para a conformidade legal dos processos financeiros, conforme estabelecido pela Lei nº 4.320/1964. Isso garante que os gastos sejam planejados dentro das previsões orçamentárias, contribuindo para o equilíbrio das contas públicas.

Em suma, o item de empenho é uma peça chave no mecanismo de governança financeira do Estado, essencial para a manutenção da responsabilidade fiscal, do controle interno e da transparência na gestão pública. Seu uso adequado é fundamental para assegurar que os recursos públicos sejam utilizados de maneira eficiente e conforme os interesses da sociedade.

2.1.2 Tipos de Empenho

A existência de diferentes tipos de empenho visam adequar a formalização das despesas às suas características específicas. A escolha do tipo de empenho a ser utilizado depende da natureza da despesa e da sua previsibilidade. Existem três tipos de empenho, a saber:

- **Ordinário:** Utilizado quando o valor exato da despesa é conhecido, e o pagamento será feito de uma única vez. Este tipo de empenho é mais simples e direto, sendo aplicado para despesas mais previsíveis e de valores fixos.
- **Estimativo:** Aplica-se quando não é possível determinar com exatidão o valor total da despesa, como nos casos de contas de consumo (por exemplo, energia elétrica e água), onde o valor exato pode variar ao longo do tempo. Embora a quantia exata não seja conhecida, o valor é estimado para fins de previsão orçamentária.
- **Global:** Usado para despesas contratuais que envolvem pagamentos parcelados ao longo do tempo. Esse tipo de empenho é comum em contratos de longo prazo, como os de construção de obras públicas, fornecimento de serviços contínuos ou outras contratações de longo prazo. Ele permite o controle adequado das obrigações financeiras, divididas em parcelas conforme o andamento do contrato.

A utilização desses tipos de empenho busca atender à necessidade de flexibilidade no processo orçamentário, garantindo que o governo possa se adaptar a diferentes situações financeiras e de execução de projetos. Cada tipo de empenho tem características que permitem sua aplicação em diferentes contextos, assegurando que o planejamento orçamentário e a execução das despesas estejam segundo a legislação vigente e as previsões financeiras.

2.1.3 Estrutura de um Item de Empenho

A estrutura típica de um Item de Empenho segue um formato padronizado, que visa garantir a transparência, o controle e a rastreabilidade das despesas públicas. Embora possam existir

variações entre os sistemas de gestão orçamentária de diferentes órgãos, os principais elementos presentes em um Item de Empenho são:

- **Identificação do Item de Empenho:** Inclui o `IdContrato`, que identifica de forma única cada contrato, garantindo que cada empenho esteja relacionado a uma transação específica, a data de emissão (que indica o momento da formalização do compromisso financeiro), o campo `Unidade`, que se refere ao órgão ou entidade responsável pela execução do contrato.

- **Dados do Beneficiário (Fornecedor ou Prestador de Serviço):** O campo `Credor`, que identifica o nome e informações complementares do fornecedor relacionado ao empenho.

- **Informações Orçamentárias:** Inclui o campo `ElemDespesaTCE`, com base nas categorias predefinidas pela legislação e pelo TCE/RJ (ex.: material de consumo, serviços de terceiros), a fonte de recurso, que indica a origem dos recursos utilizados para a despesa, o programa e ação orçamentária, que vincula a despesa ao planejamento estratégico e ao orçamento público, e o plano interno, que é um código usado para rastreamento contábil e gerencial.

- **Descrição da Despesa:** O campo `Histórico` descreve de forma detalhada o serviço ou item relacionado ao empenho, proporcionando informações adicionais sobre a finalidade ou natureza do gasto, como serviços financeiros para arrecadação ou a contratação de empresas especializadas.

- **Informações Adicionais:** Inclui a vinculação a contrato ou licitação, o número do processo licitatório correspondente, se aplicável, o prazo para execução ou entrega e observações complementares relevantes.

- **Autorizações e Assinaturas:** Refere-se à identificação do responsável pela emissão do empenho, à autorização do ordenador de despesas e à assinatura ou carimbo digital, quando aplicável, para validação.

Um exemplo ilustrativo dessa estrutura pode ser visualizado na Figura 2.1.

Campo	**Informação**
Número do Empenho	2025NE0001
Data	05/02/2025
Órgão Emitente	Ministério da Educação
Unidade Gestora	Secretaria de Tecnologia
Fornecedor	Empresa XYZ Ltda
CNPJ	12.345.678/0001-90
Elemento de Despesa	339030 (Material de Consumo)
Fonte de Recurso	100 (Tesouro Nacional)
Objeto da Despesa	Aquisição de notebooks para escolas públicas
Quantidade	50 unidades
Valor Unitário	R$ 5.000,00
Valor Total	R$ 250.000,00
Tipo de Empenho	Ordinário
Vinculação a Licitação	Pregão Eletrônico 001/2025
Prazo de Entrega	30 dias
Assinatura do Ordenador	João da Silva

Figura 2.1: Exemplo de um Item de Empenho, documento utilizado na administração pública para formalizar a reserva de recursos orçamentários destinados à realização de despesas previamente autorizadas.

2.2 Processamento de Linguagem Natural

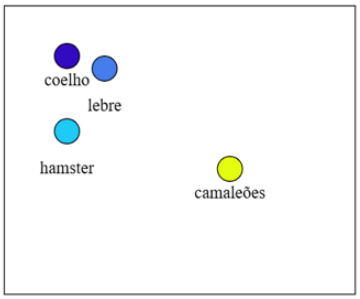
O Processamento de Linguagem Natural, é um tipo de AM que se dedica a desenvolver modelos e algoritmos capazes de entender, interpretar e gerar linguagem humana de maneira que seja útil para diversas aplicações, como tradução automática, análise de sentimentos, pesquisa de similaridade (*similarity search*), chatbots, e resumo de textos.

O PLN é capaz de lidar com as relações de sintaxe, semântica e pragmática da linguagem natural. Uma das tarefas fundamentais do PLN é a conversão de textos em representações numéricas vetoriais, que podem ser interpretadas de forma mais eficiente pelos sistemas computacionais. Esse processo é realizado por meio de modelos de *embedding* de texto, os quais convertem palavras, frases ou até sentenças inteiras em vetores de dimensão fixa, conhecidos como *embeddings*.

Os *embeddings* são representações vetoriais contínuas de palavras ou outras unidades de dados, como frases ou documentos. Em vez de representar as palavras como valores binários ou categóricos, os *embeddings* mapeiam as palavras para um espaço vetorial de alta dimensão, onde palavras semanticamente semelhantes ou relacionadas ficam localizadas próximas umas das outras. No caso dos itens de empenho, essas representações podem ser usadas para padronizar e agrupar descrições semelhantes, mesmo quando expressas de formas diferentes. Por exemplo, termos como “compra de materiais” e “aquisição de insumos” poderiam ser reconhecidos como

semanticamente próximos. Um exemplo ilustrativo dessa estrutura pode ser visualizado na Figura 2.2.

	vivo	mamífero	bípede	roedor	gênero	plural
lebre	0,9	0,7	-0,1	-0,1	0,2	-0,3
coelho	0,9	0,8	-0,1	-0,1	0,2	-0,3
hamster	0,9	0,6	-0,3	-0,4	-0,4	-0,5
camaleões	0,9	-0,2	-0,2	-0,5	-0,2	0,8



mãe	0,9	0,8	0,9	-0,2	0,8	-0,2
pai	0,9	0,9	0,9	-0,3	-0,8	-0,3
mulher	0,8	0,7	0,9	-0,1	0,9	-0,1
homem	0,8	0,8	0,8	-0,2	-0,9	-0,3

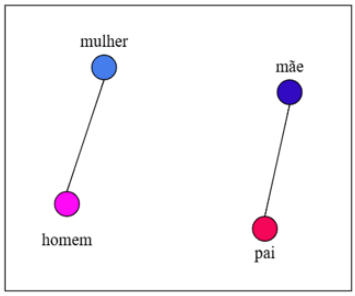


Figura 2.2: Exemplo ilustrativo de uma estrutura de *embedding*, representando a transformação de dados em um espaço vetorial de menor dimensão para capturar relações semânticas e padrões subjacentes.

Cada dimensão de um *embedding* representa uma característica latente associada a uma palavra ou entidade, permitindo a captura de relações semânticas, contextuais e sintáticas durante o processo de treinamento. Inicialmente, os vetores de palavras são atribuídos de maneira aleatória e, subsequentemente, ajustados com base nos contextos em que essas palavras ocorrem. Quando se trabalha com *embeddings* em espaços vetoriais de 300 dimensões, por exemplo, cada palavra é representada por um vetor de 300 números, onde cada dimensão reflete uma faceta distinta do significado da palavra. As principais características representadas nas dimensões de um *embedding* incluem:

- **Semântica**: Aspectos do significado das palavras. Por exemplo, a palavra "rei" pode estar mais próxima de "rainha", "príncipe" e "monarquia", enquanto "cachorro" está mais próxima de "gato", "animal" e "pet". Cada uma dessas dimensões pode refletir uma faceta do significado.

- **Contexto**: O contexto de uma palavra, que pode ser representado por várias dimensões, dependendo de como ela é usada em frases diferentes. Por exemplo, a palavra "banco" pode ser representada de maneira diferente quando se refere a uma instituição financeira

ou a um assento. As dimensões podem capturar esses diferentes usos baseados no contexto.

- **Sintaxe**: A estrutura gramatical das palavras também pode ser representada nas dimensões de um *embedding*. Por exemplo, as palavras "correr" e "correndo" podem ser representadas de forma similar em várias dimensões devido à sua relação gramatical.

- **Relações de proximidade**: algumas dimensões podem representar relações específicas entre palavras, como pluralidade, tempo verbal, gênero, etc. Por exemplo, se "homem" e "mulher" estão representados como vetores próximos no espaço vetorial, a diferença entre eles pode capturar um aspecto de gênero.

2.2.1 Modelos de *Embedding* de Dados

O conceito central por trás dos modelos de *embedding* é a ideia de aprender representações semânticas, de tal forma que a similaridade entre palavras ou unidades de texto possa ser medida por distâncias vetoriais, como a distância euclidiana ou o cosseno de similaridade. Isso permite que o modelo capture relações complexas entre palavras, como sinonímia. Entre os principais modelos de *embedding*, destacam-se:

- **Word2Vec**: um dos modelos mais conhecidos para geração de *embeddings* de palavras, foi proposto por Mikolov et al. (2013). Ele utiliza uma rede neural simples para mapear palavras em um espaço vetorial de alta dimensão, para maximizar a probabilidade de prever uma palavra a partir de seu contexto (ou vice-versa). O Word2Vec possui duas abordagens principais: o Continuous Bag of Words (CBOW) e o Skip-gram, sendo o Skip-gram mais eficaz para grandes volumes de dados e palavras menos frequentes.

- **GloVe** (Global Vectors for Word Representation): Desenvolvido por Pennington et al. (2014), o GloVe é um modelo de *embedding* que combina a contagem de ocorrência de palavras no corpus com uma fatoração matricial. O objetivo é aprender representações vetoriais de palavras que preservem as relações de coocorrência global do corpus, ao contrário do Word2Vec, que foca nas coocorrências locais.

- **BERT** (Bidirectional Encoder Representations from Transformers): BERT, proposto por Devlin et al. (2018), é um modelo pré-treinado baseado na arquitetura Transformer que captura o contexto de uma palavra em ambas as direções (da esquerda para a direita e

vice-versa). Ao contrário dos modelos anteriores, o BERT pode ser utilizado para gerar *embeddings* contextualizados, onde a representação de uma palavra varia conforme o contexto em que ela aparece. Esse modelo revolucionou muitas tarefas de *PLN*, incluindo a tradução automática, a análise de sentimentos e a resposta a perguntas.

- ***Sentence BERT* (SBERT)**: uma extensão do BERT, o Sentence Transformers (conhecido como SBERT) é otimizado para gerar *embeddings* de sentenças e parágrafos inteiros, permitindo medir a similaridade semântica entre textos mais longos. É particularmente eficaz para tarefas de comparação entre sentenças, como busca semântica e recuperação de informações.

Esses modelos de *embeddings* são fundamentais para tarefas de PLN, pois fornecem uma representação eficiente e densa das palavras ou textos, facilitando a modelagem e análise de grandes volumes de dados textuais.

2.3 Redes Neurais Profundas

As **Redes Neurais Profundas** (ou *Deep Neural Networks* - Deep Neural Networks (DNNs)) são um tipo de arquitetura de rede neural composta por múltiplas camadas entre a entrada e a saída. Elas se destacam por sua capacidade de modelar relações complexas e extrair características de alto nível a partir de dados brutos, sendo amplamente utilizadas em áreas como visão computacional e PLN. Esta estrutura é composta por:

- **Camada de entrada (*input layer*)**: responsável por receber os dados de entrada, convertendo-os em um formato que pode ser processado pelas camadas seguintes.

- **Camadas ocultas (*hidden layers*)**: constituem o núcleo da rede, onde ocorrem as transformações não lineares dos dados. Cada camada oculta é composta por múltiplos neurônios (ou perceptrons), que processam as informações por meio de pesos, *bias* e funções de ativação.

- **Camada de saída (*output layer*)**: gera a resposta final da rede, como uma classificação, uma regressão ou outra forma de predição, dependendo da tarefa.

Embora compartilhem a mesma base conceitual das Redes Neurais Artificiais simples, que possuem apenas uma camada oculta, as DNNs se diferenciam pela profundidade, isto é, pelo

número maior de camadas ocultas, o que as permitem aprender representações mais abstratas e hierárquicas dos dados. Uma representação de uma Rede Neural Profunda pode ser observada na Figura 2.3.

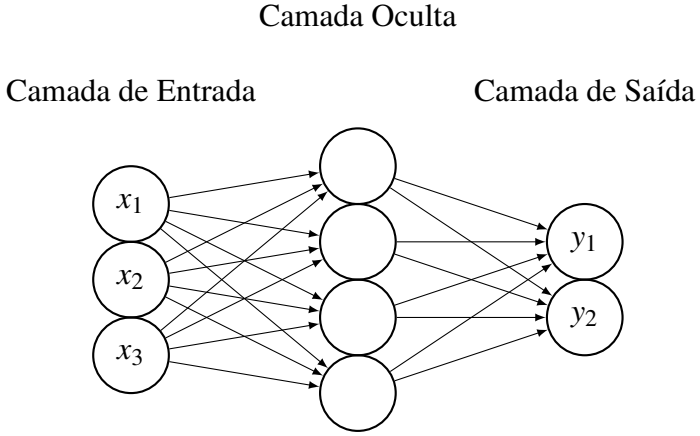


Figura 2.3: Este exemplo de Rede Neural Profunda possui uma camada oculta. A camada de entrada, possuirá a dimensão dos dados de entrada e a de saída, possuirá o número de saídas desejadas (podendo ser o número de agrupamentos procurados).

Cada camada é formada por vários *perceptrons*, que realizam operações lineares sobre os dados de entrada (multiplicações por pesos e somas com bias). Após essa etapa, uma função de ativação é aplicada, introduzindo não-linearidade e permitindo que a rede aprenda funções complexas.

Antes do treinamento, os pesos e bias são inicializados aleatoriamente. Durante o processo de aprendizagem, utiliza-se a técnica de minimização da função de custo (via algoritmos como *backpropagation* combinados com otimizadores como o gradiente descendente, como o *Stochastic Gradient Descent* (SGD)) para ajustar esses parâmetros, de modo que a rede melhore progressivamente seu desempenho na tarefa.

2.3.1 Denoising Autoencoder

Autoencoders são tipos de arquiteturas de redes neurais utilizadas para aprendizado não supervisionado. Essa arquitetura é definida por um codificador (*encoder*) e um decodificador (*decoder*). O primeiro transforma os dados de entrada em uma representação de menor dimensão, enquanto o decodificador reconstrói os dados codificados de volta para o formato original. A rede é treinada para minimizar a diferença entre os dados decodificados e os dados de entrada.

No entanto, os *Autoencoders* arriscam se tornarem uma função identidade, ou seja, produzirem uma saída idêntica à entrada, o que torna a rede neural ineficaz. Quando a estrutura possui

muitos neurônios nas camadas ocultas, mais do que na camada de entrada, o que é comum nas DNNs, então ela não precisa aprender características relevantes. Ela pode somente memorizar o input e copiar diretamente para o output.

Um *Denoising Autoencoder* é uma modificação da estrutura original. Em vez de fornecer os dados originais como entrada, é fornecida uma versão corrompida ou com ruído dos dados ao codificador. No entanto, a função de perda de reconstrução continua sendo calculada com base nos dados originais.

Isso leva o algoritmo a ter que aprender características significativas dos dados, sendo obrigado a aprender a imputar ou preencher informações faltantes durante o processo de reconstrução. Desta forma, levando a um aprendizado mais eficiente e reduzindo significativamente o risco de o *Autoencoder* aprender apenas a função identidade. É possível visualizar as etapas do *Autoencoder* através da Figura 2.4.

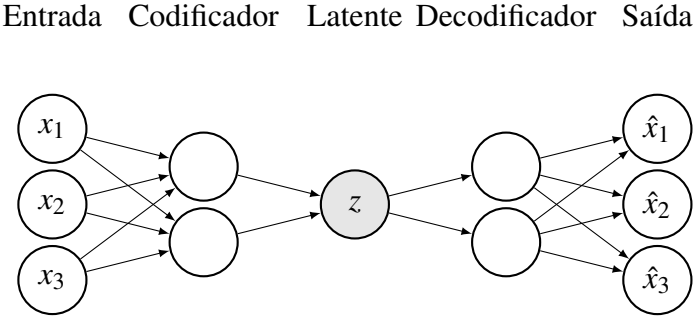


Figura 2.4: Esquema ilustrativo de um Autoencoder. Os dados de entrada x são comprimidos em uma representação latente z por meio do codificador, e reconstruídos como $\hat{x}$ pelo decodificador.

Dessa forma, podemos encarar a estrutura geral como a composição de duas redes neurais distintas conectadas: o codificador e o decodificador, conforme definido na Equação 2.1. Cada dado de entrada x_i é primeiramente corrompido por ruído e, em seguida, processado pelo codificador $E(\cdot)$, que gera uma representação latente. O decodificador $D(\cdot)$ então tenta reconstruir a entrada original a partir dessa representação.

$$\hat{x}_i = D(E(x_i + \text{ruído})) \tag{2.1}$$

Após a reconstrução, calcula-se a função de perda, que mede o erro entre os dados originais x_i e os dados reconstruídos $\hat{x}_i$. Com base nesse erro, os pesos dos neurônios são ajustados, visando minimizar a perda e, consequentemente, aprimorar a qualidade da reconstrução ao longo do treinamento.

2.4 Algoritmos de Agrupamento

O agrupamento de dados é uma técnica que visa organizar dados em grupos, ou agrupamentos, com base em sua similaridade. Naturalmente, os modelos de agrupamento de dados são não-supervisionados. Isso significa que o processo ocorre sem o uso de conhecimento prévio sobre os grupos ou suas características. De fato, em muitos casos, pode-se sequer saber quantos grupos são necessários identificar.

O que distingue os modelos de agrupamento de dados de outras técnicas de aprendizado de máquina é que não há uma saída ou campo alvo pré-definido para o modelo prever. Ao invés disso, o foco está em descobrir padrões ou estruturas ocultas nos dados, o que torna esses algoritmos uma poderosa ferramenta para explorar e entender dados não rotulados. Esse processo pode ser visualizado na Figura 2.5.

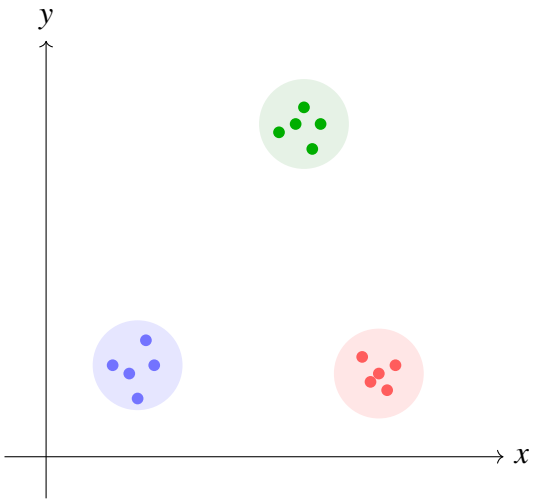


Figura 2.5: Representação visual de três agrupamentos no plano cartesiano, cada um com pontos próximos ao seu respectivo centroide. A figura demonstra como um algoritmo de agrupamento, como o *KMeans*, pode agrupar dados com base em similaridade espacial.

Essa técnica pode ser aplicada para segmentar clientes de uma empresa com base em seu comportamento de compra, por exemplo. Algoritmos como o *KMeans* podem ser usados para agrupar clientes em agrupamentos, com base em características como frequência de compra, valor gasto, categorias de produtos comprados, entre outras. Essa abordagem permite a segmentação de clientes de forma não supervisionada, sem a necessidade de rótulos prévios, facilitando a identificação de grupos de consumidores com comportamentos semelhantes, o que pode ser útil para estratégias de marketing personalizadas.

2.4.1 Algoritmo *KMeans*

O algoritmo *KMeans* é uma técnica de aprendizado não supervisionado amplamente utilizada para tarefas de agrupamento (*clusterização*). Seu objetivo principal é particionar um conjunto de dados em *k* agrupamentos, de forma que os dados em um mesmo grupo sejam mais semelhantes entre si do que em relação a dados de outros grupos.

O funcionamento do *KMeans* pode ser descrito pelas seguintes etapas principais:

1. **Inicialização:** selecionam-se k centroides iniciais, que podem ser escolhidos aleatoriamente a partir dos dados ou utilizando técnicas mais robustas, como o *KMeans++*, que busca distribuir melhor os centroides iniciais para evitar más convergências.

2. **Atribuição de agrupamentos:** cada ponto de dado é atribuído ao agrupamento cujo centróide está mais próximo, utilizando geralmente a distância euclidiana como métrica de proximidade.

3. **Atualização dos centroides:** os centroides são recalculados como a média dos pontos atribuídos a cada agrupamento.

4. **Convergência:** as etapas de atribuição e atualização são repetidas até que os centroides não se alterem significativamente entre as iterações, ou até que um número máximo de iterações seja atingido.

Matematicamente, o objetivo do *KMeans* é minimizar a soma das distâncias quadradas entre os pontos e seus respectivos centroides, dada pela Equação 2.2.

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{2.2}$$

Onde C_i representa o conjunto de pontos atribuídos ao i-ésimo agrupamento e μ_i é o centróide desse agrupamento.

Apesar de sua simplicidade, o *KMeans* é eficiente e escalável para grandes volumes de dados, sendo amplamente utilizado em problemas de análise de padrões. No entanto, o número de agrupamento k deve ser definido previamente, e o desempenho do algoritmo pode ser afetado pela escolha inicial dos centroides.

2.4.2 Deep Embedded Clustering

Em dados de alta dimensionalidade, como no PLN, o número de características pode ser muito elevado. Nessas situações, abordagens tradicionais de agrupamento de dados não-supervisionada tendem a perder eficiência. Para mitigar esse problema, surgiram variantes do *KMeans* que realizam redução conjunta de dimensionalidade e agrupamento. No entanto, essas abordagens são limitadas a incorporações lineares.

Desenvolvido por Xie et al. [2016], o método DEC aplica DNNs para lidar com dados complexos e não-lineares. Por meio de um *Denoising Autoencoder*, aprende-se uma representação latente Z dos dados, mais compacta e estruturada, o que torna o agrupamento mais eficaz em espaços de alta dimensão.

O algoritmo realiza dois processos fundamentais. Inicialmente, ajusta um conjunto de k centroides com base nos vetores do espaço latente Z. Em seguida, treina os parâmetros θ da rede neural profunda, que projeta os dados nesse espaço.

O primeiro processo tem início assim que os centroides iniciais são definidos. Esses centroides são obtidos por meio de um agrupamento inicial dos pontos em Z utilizando o algoritmo *KMeans*. A partir desse momento, inicia-se a otimização dos parâmetros da rede, refinando tanto a representação dos dados no espaço latente quanto a própria atribuição dos agrupamentos.

Processo de treinamento

O treinamento tem início com a transformação das variáveis de entrada em representações latentes Z, utilizando *Denoising Autoencoders* empilhados. Esse vetor latente é uma versão condensada dos dados originais, mantendo suas características mais relevantes em um espaço de menor dimensionalidade, o que torna o processo subsequente mais eficiente. O funcionamento detalhado do Autoencoders Empilhados (SAE) é apresentado na Seção 2.3.1.

Na etapa de agrupamento de dados, as representações latentes Z obtidas são utilizadas como entrada para o método de agrupamento proposto. Inicialmente, o algoritmo *KMeans* é aplicado para a definição inicial dos centróides dos agrupamentos. Em seguida, o processo de refinamento é conduzido em duas subetapas principais: a atribuição suave dos pontos aos agrupamentos e o ajuste iterativo dos agrupamentos por meio da minimização da divergência de Kullback-Leibler (KL). Esse procedimento é repetido ao longo de um número predefinido de

épocas, permitindo a convergência dos agrupamentos a uma estrutura mais coerente com a distribuição dos dados.

1. **Atribuição suave dos pontos aos agrupamentos.** Nesta etapa, os pontos no espaço latente são atribuídos a agrupamentos de forma probabilística. Para isso, utiliza-se a distribuição t de Student, com grau de liberdade igual a 1, para calcular a similaridade entre cada ponto embutido z_i ($z_i \in Z$) e os centroides μ_j dos agrupamentos, conforme a Equação 2.3. O resultado é uma matriz de atribuição suave Q, onde cada entrada Q_{ij} representa a probabilidade do ponto i pertencer ao agrupamento j.

$$q_{ij} = \frac{(1+||z_i - \mu_j||^2)^{-1}}{\sum_k (1+||z_i - \mu_k||^2)^{-1}} \tag{2.3}$$

   Nessa formulação, q_{ij} representa a probabilidade de que o ponto z_i pertença ao agrupamento cujo centroide é μ_j. A atribuição final será então $argmax_j q_{ij}$.

2. **Refinamento dos agrupamentos.** Esta segunda etapa consiste em alinhar a atribuição suave definida no último passo com uma distribuição-alvo P. Isso é feito através da perda de Kullback-Leibler (KL) entre a matriz de atribuição suave Q e a distribuição auxiliar P.

   Essa distribuição auxiliar consiste em atribuições de alta confiança, definida conforme a Equação 2.4.

$$P_{ij} = \frac{\left(q_{ij}^2 / f_j\right)}{\sum_{j'} \left(q_{ij'}^2 / f_{j'}\right)}, \quad \text{onde } f_j = \sum_i q_{ij} \tag{2.4}$$

   A função de perda de Kullback-Leibler (KL), usada para aprimorar as previsões dos agrupamentos, mede a diferença entre as distribuições de probabilidade das variáveis aleatórias Q e P. A minimação dessa função, denotada na Equação 2.5, incentiva o modelo a aproximar q_{ij} de p_{ij}, ajustando assim tanto os parâmetros da rede quanto a estrutura dos agrupamentos de forma iterativa.

$$L = KL(P||Q) = \sum_i \sum_J p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{2.5}$$

Conforme destacado por Xie et al. [2016], o processo de aprendizado do DEC pode ser interpretado como uma forma de autoaprendizado (*self-training*), na qual o próprio modelo, a

partir de um classificador inicial, rotula os dados com base em suas previsões de alta confiança, utilizando essas previsões como base para o refinamento progressivo dos agrupamentos.

2.5 Métricas de Avaliação

2.5.1 Silhouette Score

O *Silhouette Score* é uma métrica amplamente utilizada para avaliar a qualidade de agrupamentos gerados pelos algoritmos. Esse índice combina medidas de coesão e separação para indicar o quão bem cada ponto de dado está alocado em seu respectivo grupo, em comparação com os demais grupos.

Para cada amostra, o índice de silhueta $s(i)$ é calculado seguindo a Equação 2.6.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{2.6}$$

onde:

- $a(i)$ é a média das distâncias entre o ponto i e todos os outros pontos dentro do mesmo agrupamento (medida de *coesão*);
- $b(i)$ é a menor média das distâncias entre o ponto i e todos os pontos de qualquer outro agrupamento ao qual i não pertence (medida de *separação*).

O valor de $s(i)$ varia entre -1 e $+1$. Valores próximos de 1 indicam que o ponto está bem ajustado ao seu próprio agrupamento e bem separado dos demais. Valores próximos de 0 sugerem que o ponto está na fronteira entre dois agrupamentos. Já valores negativos indicam que o ponto pode ter sido alocado no agrupamento incorreto.

O *Silhouette Score* global é obtido pela média dos valores $s(i)$ de todos os pontos, sendo útil para comparar diferentes configurações de agrupamento, como diferentes valores de k no algoritmo *k-means*. Essa métrica é particularmente valiosa em cenários de aprendizado não supervisionado, onde não há rótulos verdadeiros disponíveis.

2.5.2 Erro Quadrático Médio

O Erro Quadrático Médio (Erro Médio Quadrático (MSE)) é uma métrica estatística amplamente utilizada para avaliar o desempenho de modelos de regressão. Ele quantifica a diferença

média entre os valores previstos pelo modelo e os valores reais, elevando ao quadrado essas diferenças para penalizar erros maiores de forma mais severa.

Podemos o através da Equação 2.7.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{2.7}$$

onde:

- n representa o número total de observações;
- y_i é o valor real da i-ésima observação;
- $\hat{y}_i$ é o valor previsto pelo modelo para a i-ésima observação.

O *MSE* sempre assume valores não negativos, sendo que quanto mais próximo de zero, melhor é o desempenho do modelo. Por elevar as diferenças ao quadrado, ele é sensível a outliers, o que pode ser desejável em contextos onde erros maiores precisam ser mais penalizados. Essa métrica é amplamente utilizada por sua simplicidade e por fornecer uma medida clara da magnitude dos erros de predição.

Capítulo 3

Trabalhos Relacionados

Neste capítulo, apresentamos os trabalhos relacionados ao tema desta pesquisa. A Seção 3.1 descreve a metodologia utilizada para a seleção dos trabalhos. A Seção 3.2 apresenta um resumo das abordagens adotadas pelos estudos, suas metodologias, principais resultados e conclusões. Por fim, a Seção 3.3 apresenta uma análise comparativa dos trabalhos apresentados.

3.1 Seleção dos Trabalhos

A seleção dos trabalhos relacionados foi orientada por critérios específicos de inclusão e exclusão, de modo a garantir que apenas estudos relevantes e alinhados ao escopo desta pesquisa fossem considerados. Foram definidos dois critérios distintos de inclusão, apresentados nas Seções seguintes.

O primeiro critério de inclusão consistiu na seleção de pesquisas que abordassem o agrupamento de dados financeiros, com ênfase na utilização de dados públicos. Já o segundo critério considerou estudos que explorassem a aplicação do algoritmo DEC em cenários reais.

Para o primeiro critério de inclusão, os critérios de exclusão envolveram a eliminação de trabalhos que tratavam a análise de agrupamento de dados de maneira genérica, sem relação direta com o contexto financeiro. Considerando a ampla aplicação da análise de agrupamentos na literatura, estudos que não apresentavam vínculo explícito com a análise de dados financeiros foram desconsiderados, a fim de manter a coerência temática da revisão.

No que se refere ao segundo critério de inclusão, foram excluídos trabalhos que, embora utilizassem técnicas relacionadas ao DEC, focavam em abordagens derivadas ou modificadas, desviando-se da aplicação direta do algoritmo em sua forma original.

Para garantir uma seleção abrangente, utilizamos uma busca sistemática em bases científicas renomadas, incluindo Google Scholar [2024] e IEEE Xplore [2024]. Além disso, aplicamos a técnica de *snowballing*, analisando as referências citadas em estudos relevantes para identificar novos trabalhos de interesse.

3.2 Apresentação dos Trabalhos

3.2.1 Agrupamento em Dados Financeiros Públicos

Diversos estudos exploram a análise de agrupamento de dados para dados financeiros públicos, aplicando diferentes abordagens e técnicas. A seguir, apresentamos um resumo dos principais trabalhos identificados na revisão da literatura.

Detecção de Fraudes em Dados Financeiros com KMeans

Joksimovic et al. [2023] propõem uma abordagem baseada no algoritmo *KMeans*, utilizando distância Euclidiana e normalização *Z-Score*, visando detectar fraudes em dados financeiros. O conjunto de dados utilizado inclui grandes volumes de transações financeiras, e os experimentos realizados avaliam a eficácia do modelo na identificação de padrões suspeitos.

Os autores descrevem um processo que envolve a normalização dos dados utilizando *Z-Score*, a aplicação do algoritmo KMeans para agrupar transações com base em similaridades e a análise das anomalias detectadas nos agrupamentos formados. Os resultados indicam que o método proposto é eficaz na identificação de inconsistências e padrões atípicos, contribuindo para a análise e monitoramento de atividades financeiras suspeitas.

Clusterização Explicável para Detecção de Fraudes

Min et al. [2021] exploram uma abordagem alternativa baseada em um algoritmo de agrupamento combinado com técnicas de AP, visando alcançar alta eficiência computacional na detecção de fraudes em dados financeiros. O estudo considera dados financeiros transacionais, avaliando a eficácia da técnica proposta em relação a métodos tradicionais de agrupamento.

A avaliação do desempenho foi realizada por meio de métricas de eficiência computacional e interpretabilidade do modelo, sendo observadas melhorias na capacidade de identificar padrões anômalos em comparação com abordagens convencionais. Os autores destacam que o algoritmo *KMeans* apresenta limitações, como a necessidade de definir um número fixo de agrupamentos (k), sua sensibilidade a anomalias (*outliers*) e a dependência das sementes iniciais. Para superar essas limitações, a abordagem adotada incorpora técnicas avançadas, incluindo a combinação de *Bi-directional LSTM (BiLSTM)* e um mecanismo de atenção para o processamento de dados sequenciais. Além disso, o estudo enfatiza a importância da interpretabilidade do modelo, ga-

rantindo que os resultados não sejam apenas precisos, mas também compreensíveis, facilitando a tomada de decisões informadas.

3.2.2 Aplicação do método Deep Embedded Clustering

O método DEC tem sido utilizado em diferentes domínios como uma abordagem eficiente para o agrupamento de dados complexos e de alta dimensionalidade. A técnica combina aprendizado de representações com agrupamento, permitindo identificar padrões latentes nos dados. Nesta seção, são apresentados estudos que aplicam o Deep Embedded Clustering em diferentes contextos.

Clusterização Espaço-Temporal de dados de Tráfego com Deep Embedded Clustering

Asadi and Regan [2019] apresentam uma abordagem baseada em aprendizado profundo para o agrupamento de dados de tráfego, utilizando o método DEC. O objetivo principal do estudo é identificar padrões espaço-temporais em grandes volumes de dados de tráfego, capturando similaridades tanto no domínio temporal quanto espacial. Para isso, os autores aplicam o Deep Embedded Clustering em séries temporais obtidas de sensores de tráfego, com o intuito de gerar agrupamentos que revelem comportamentos similares de tráfego entre diferentes regiões e horários.

O trabalho se destaca por adaptar essa estrutura para lidar com a séries temporais complexas e de alta-dimensionalidade, incorporando a informação geográfica dos sensores para melhorar a coesão dos agrupamentos formados. Os experimentos demonstram que o método é capaz de identificar padrões relevantes para aplicações como detecção de anomalias e planejamento urbano, superando abordagens tradicionais de agrupamento.

Esse estudo é relevante para este trabalho por demonstrar a aplicabilidade do DEC em dados reais e complexos, além de destacar a importância do pré-processamento e da representação latente na qualidade dos agrupamentos gerados. Destaca-se também a metodologia adotada para a escolha dos hiperparâmetros, em especial a definição do número de agrupamento k utilizado na etapa do *KMeans*, onde os autores consideram métricas como a inércia, definida como a soma dos quadrados das distâncias dos pontos aos seus respectivos centroides, como critério de avaliação.

Outro ponto relevante é a análise feita pelos autores sobre a correlação entre a distância euclidiana das representações latentes e a distância DTW (*Dynamic Time Warping*) entre as

séries temporais originais, o que reforça a consistência do espaço latente aprendido em preservar características temporais dos dados.

Aplicação do framework DEC para dados mistos

O trabalho de Lee et al. [2022] aborda o desafio de agrupar conjuntos de dados que contêm tanto características numéricas quanto categóricas — um cenário comum em aplicações do mundo real. Os métodos tradicionais do DEC são projetados principalmente para dados numéricos, o que pode limitar sua aplicabilidade. Para superar essa limitação, os autores propõem algumas alterações na estrutura original da implementação do DEC, integrando técnicas de aprendizado profundo para lidar de forma eficaz com dados mistos.

No caso dos dados categóricos, os autores implementam técnicas que possibilitam sua interpretação pelo modelo, utilizando camadas de *embedding* ou codificação *one-hot* para transformar essas variáveis em representações numéricas compatíveis com o processo de aprendizado. Essas representações, juntamente com as características numéricas originais, que não requerem codificação, são então combinadas e utilizadas como entrada para o modelo DEC.

Além disso, visando aperfeiçoar os resultados, a função de perda *Mean Squared Error (MSE)*, utilizada no *Autoencoder*, no DEC original, foi substituída pela função de perda de *cross-entropy*, aplicada apenas às variáveis categóricas. Essa abordagem trata a reconstrução dessas variáveis, durante a etapa de pré-treinamento, como um problema de classificação multiclasse, o que resultou em um aumento na eficiência do treinamento nessa etapa.

Na etapa de agrupamento de dados, os autores adotam uma estratégia para melhorar a convergência do modelo. Durante a minimização da divergência de Kullback-Leibler (KL) no DEC, é necessário atualizar a matriz de *target distribution* a cada época. No entanto, quando a diferença entre as matrizes de *soft assignment* e *target distribution* é muito elevada, a estabilidade da convergência pode ser comprometida.

Para mitigar esse problema, os autores propõem o uso de uma *target network*, que consiste em uma versão suavizada da matriz de atribuição suave (*soft assignment*), atualizada de forma mais lenta ao longo das épocas. Essa abordagem é utilizada para atualizar a matriz de atribuição suave, contribuindo para uma convergência mais estável e melhor desempenho na etapa de agrupamento.

3.3 Comparação dos Trabalhos

Os estudos revisados demonstram diferentes estratégias para abordar a detecção de fraudes em dados financeiros, colocando em prática métodos variados de agrupamento de dados e AP para otimizar o agrupamento de dados financeiros. De modo geral, observa-se que:

- Agrupamento de dados tradicional, como *KMeans*, tem limitações significativas, incluindo a necessidade de definir um número fixo de agrupamentos (k) e sensibilidade a *outliers*.
- Abordagens híbridas que combinam aprendizado profundo (como BiLSTM) e mecanismos de atenção vêm sendo exploradas para melhorar a capacidade de identificar padrões complexos em dados sequenciais.
- Abordagens de agrupamentos com Deep Embedded Clustering têm se mostrado eficazes em contextos com dados complexos e de alta dimensionalidade.
- Há uma tendência em adaptar o DEC para lidar com tipos de dados específicos, como séries temporais ou dados mistos, ajustando tanto a estrutura do modelo quanto suas funções de perda.
- O pré-processamento dos dados, incluindo a escolha do número de agrupamentos e a transformação de variáveis, impacta significativamente os resultados.
- A eficiência computacional é uma preocupação central, especialmente para lidar com grandes volumes de dados financeiros em tempo real.
- Diferentemente das abordagens anteriores, este trabalho aplica o DEC a um domínio específico de dados governamentais (itens de empenho). Ele propõe adaptações na estrutura do modelo e estratégias customizadas de seleção de k, além de integrar os resultados em uma aplicação prática de busca semântica.

A Tabela 3.1 apresenta um resumo comparativo dos trabalhos analisados:

Com base na comparação, identificamos que técnicas de agrupamento de dados são amplamente utilizadas no agrupamento de dados financeiros e há uma tendência crescente na adoção de métodos baseados em AP para lidar com dados complexos e heterogêneos. Enquanto abordagens tradicionais como o *KMeans* ainda são eficazes para detectar padrões em grandes volumes

de dados numéricos, métodos mais avançados, como o DEC e modelos híbridos com redes neurais, demonstram maior robustez e interpretabilidade em cenários com séries temporais, dados categóricos ou de alta dimensionalidade.

Além disso, destaca-se a importância do pré-processamento e da escolha adequada de métricas, como inércia, divergência KL e funções de perda adaptadas, que influenciam diretamente na qualidade dos agrupamentos. Técnicas como *embeddings*, redes BiLSTM e mecanismos de atenção têm se mostrado eficazes para enriquecer a representação dos dados antes da etapa de agrupamento, contribuindo para resultados mais precisos e aplicáveis em problemas de agrupamento.

3.4 Considerações Finais

Os trabalhos analisados apresentam diferentes abordagens para o agrupamento de dados financeiros, contribuindo para a evolução das técnicas aplicadas na área. Observamos que há uma tendência crescente na adoção de métodos baseados em aprendizado profundo, especialmente aqueles que integram representações latentes com algoritmos de agrupamento, como o Deep Embedded Clustering. Esses métodos têm se mostrado promissores para lidar com dados de alta dimensionalidade, séries temporais e conjuntos com variáveis mistas, superando limitações de técnicas tradicionais como o *KMeans*.

Além disso, estratégias de pré-processamento, escolha criteriosa de métricas de avaliação e adaptações na arquitetura dos modelos desempenham um papel fundamental na obtenção de resultados mais eficientes, robustos e interpretáveis. A análise comparativa evidencia que, ao considerar características específicas dos dados e objetivos da aplicação, é possível alcançar agrupamentos mais coerentes e úteis para tomada de decisão, seja na detecção de padrões anômalos, no suporte a decisões financeiras ou em contextos urbanos complexos.

Com base nessa revisão, esta pesquisa se propõe a desenvolver uma abordagem robusta para o agrupamento de dados financeiros, explorando técnicas de aprendizado profundo que aprimorem a eficiência computacional sem comprometer a explicabilidade dos modelos. Dessa forma, este presente trabalho visa contribuir para a evolução das estratégias de monitoramento financeiro, tornando os sistemas mais confiáveis e eficazes na identificação de atividades suspeitas.

Referência	Técnica	Dados	Métricas	Conclusão
Joksimovic et al. [2023]	KMeans (Distância Euclidiana, Z-Score)	Dados financeiros em grande volume	Detecção de padrões suspeitos	Algoritmo eficaz para identificar inconsistências financeiras
Min et al. [2021]	Agrupamento de dados com AP, BiLSTM + Mecanismo de Atenção	Dados financeiros sequenciais	Eficiência computacional e interpretabilidade	Modelo mais robusto e explicável que KMeans, adequado para padrões complexos
Asadi and Regan [2019]	DEC adaptado para séries temporais	Séries temporais de sensores de tráfego com informações geográficas	Inércia, correlação com DTW	Identifica padrões espaço-temporais relevantes; útil para planejamento urbano e detecção de anomalias
Lee et al. [2022]	DEC adaptado para dados mistos com *embeddings* e função de perda modificada	Dados com variáveis numéricas e categóricas	Cross-entropy, estabilidade de convergência (KL divergence)	Melhor desempenho no agrupamento de dados heterogêneos e maior estabilidade no treinamento
Este presente trabalho	DEC adaptado para dados públicos financeiros com seleção dinâmica de k e integração a sistema de busca semântica	Itens de empenho do setor público (dados textuais)	Silhouette Score e heurísticas baseadas no tamanho do conjunto	Aplicação prática em plataforma interativa e metodologia específica para o agrupamento por categoria de despesa

Tabela 3.1: Comparação dos trabalhos relacionados, destacando as técnicas utilizadas, os conjuntos de dados analisados, as métricas de avaliação e as principais conclusões de cada estudo.

Capítulo 4

Desenvolvimento

Neste capítulo, apresentamos a metodologia adotada neste trabalho, detalhando todas as suas etapas. Na Seção 4.1 é detalhado a visão geral da solução proposta neste trabalho. Na Seção 4.2, descrevemos a principal fonte de dados utilizada, composto por itens de notas de empenho de municípios do estado do Rio de Janeiro. A Seção 4.3 é consagrada ao detalhamento da etapa de pré-processamento dos dados. Por fim, na Seção 4.4 detalhamos as técnicas aplicadas no processamento propriamente dito.

4.1 Visão geral da solução

A solução proposta neste trabalho baseia-se na adaptação do modelo DEC, originalmente implementado por Xie et al. [2016], para a base de dados de Itens de Empenho. Essa adaptação está disponível no repositório *GitHub* do projeto, em Paraizo [2024], e visa realizar o agrupamento dessas notas de maneira precisa, levando em consideração as características específicas dos dados. A seguir, descrevemos as principais etapas do processo, essenciais para garantir a eficácia da solução.

1. **Etapa de pré-processamento da base de dados:**

   (a) Transformação do arquivo de dados de `csv` para `parquet`, visando facilitar a manipulação e o desempenho no processamento dos dados.

   (b) Extração e tratamento dos dados relevantes: a coluna `histórico` será transformada em *embeddings* de dados e as colunas categóricas nominais, `ElemDespesaTCE`, `unidade` e `Credor` serão transformadas em dados numéricos através do método da codificação *frequency encoding*.

2. **Passagem dos Dados para o Modelo DEC:**

   (a) **Adaptação do Modelo para o Conjunto de Dados:** O modelo DEC é adaptado para lidar com o conjunto de dados de Itens de Empenho, levando em consideração as transformações aplicadas aos dados.

(b) **Execução do Treinamento e do Agrupamento:** O treinamento do modelo é realizado utilizando os dados transformados, e o agrupamento é realizado a partir dos *embeddings* e dos dados categóricos.

A Figura 4.1 ilustra os principais passos da solução proposta, fornecendo uma visão geral do processo de adaptação do modelo DEC e de como os dados são preparados e utilizados para o agrupamento.

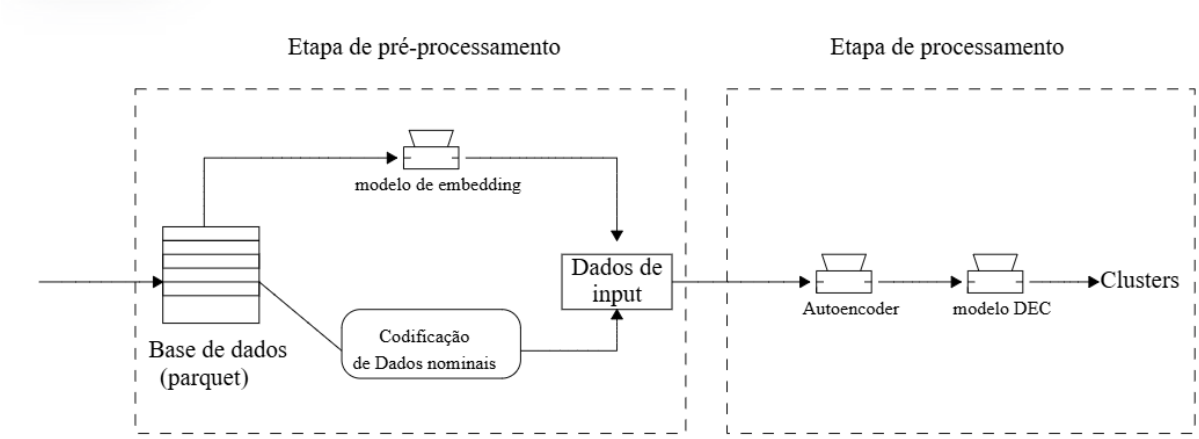


Figura 4.1: Diagrama ilustrativo dos passos da solução proposta, apresentando o fluxo das etapas principais do processo para a análise e segmentação dos dados.

4.2 Base de Dados

A base de dados utilizada neste trabalho é composta por 1.484.918 itens de empenho, com um total de 66 colunas. Essa base contém todos os três tipos de itens de empenho, descritos na Seção 2.1.2. Ela é composta por informações enviadas pelos jurisdicionados do TCE/RJ, abrangendo órgãos e entes de todos os municípios do estado do Rio de Janeiro, exceto a Capital. A análise desta base pelos auditores enfrenta limitações significativas devido a problemas específicos nos dados, o que reduz sua eficácia. Esses problemas são descritos a seguir:

- Empenhos referentes a uma mesma prestação de serviço, mas sem contrato associado, aparecem frequentemente de forma dispersa, dificultando a análise. Ou seja, muitos campos na base de dados sem o seu respectivo `IdContrato`, correspondendo a 87,17% da base total. É possível visualizar o problema descrito pelos exemplos abaixo, retirados da base de dados.

IdContrato	Credor	Valor Empenhado (R$)
0	Play Producoes e Publicidade LTDA	916 493.17
000489096/2018	Play Producoes e Publicidade LTDA	20 607.62

- Inconsistências no preenchimento do campo referente ao `Credor` dificultam a identificação de padrões. Em alguns casos, é usado o CPF do primeiro funcionário listado na folha, enquanto em outros é registrado o CNPJ da unidade ou ente contratante, sem seguir uma regra definida. Essa inconsistência torna o agrupamento mais complexo. É possível visualizar o problema descrito pelos exemplos abaixo, retirados da base de dados.

IdContrato	Credor	Valor Empenhado (R$)
0	Banco do Brasil SA 30 414 056 0001 53D	10 000.00
0	Pedro Asche Cintra Ferreira 17591418874	3 051.09

- Empenhos vinculados a uma única licitação estão frequentemente divididos em diferentes lotes, o que pode resultar em empresas distintas sendo vencedoras de cada lote. Um exemplo comum ocorre em licitações para a compra de medicamentos, onde o campo `histórico` frequentemente faz referência a mesma licitação e ao mesmo órgão ou unidade administrativa. Este problema está representado na Tabela 4.2.

Credor	Histórico	NrLicitacao	Unidade
Luiz Claudio Borgatti	Importância que se empenha para o pagamento dos serviços de transporte escolar automotivo de alunos da rede pública municipal de ensino, a pedido da SEMEEL, processo de nº 2 616/2017, pregão presencial de nº 029/2017	002616/17	Fundo Mun Educacao Bom Jesus Itabapoana
Paulo Jonas Boechat da Silveira	Importância que se empenha para o pagamento dos serviços de transporte escolar automotivo de alunos da rede pública municipal de ensino, a pedido da SEMEEL, processo de nº 2 616/2017, pregão presencial de nº 029/2017	002616/17	Fundo Mun Educacao Bom Jesus Itabapoana

Algumas colunas específicas da base de dados foram selecionadas para o pré-processamento e posterior utilização no algoritmo. A Tabela 4.1 apresenta uma análise detalhada dos conjuntos de dados escolhidos, que serão fundamentais para o desenvolvimento e treinamento do modelo. Essas colunas oferecem informações essenciais que permitem uma compreensão mais profunda dos dados e direcionam a construção de soluções mais eficazes.

IdContrato	Unidade	ElemDespesaTCE	Credor	Histórico
000363087/2020	Pref. Angra dos Reis	Outros serviços de terceiros - PJ	Banco do Brasil SA	Prestação de serviços financeiros para arrecadação de guias de tributos e demais receitas de acordo com o padrão da Febraban com prestação de contas por meio magnético dos valores recebidos na forma do termo de referência e do instrumento convocatório.
000363084/2020	Pref. Angra dos Reis	Outros serviços de terceiros - PJ	Caixa Econômica Federal	Prestação de serviços financeiros para arrecadação de guias de tributos e demais receitas segundo o padrão da Febraban com prestação de contas por meio magnético dos valores recebidos na forma do termo de referência e do instrumento convocatório.
000363084/2020	Pref. Angra dos Reis	Outros serviços de terceiros - PJ	Caixa Econômica Federal	Chamamento público 01 2020 SFI proc 2019021268 serviços financeiros para arrecadação de guias de tributos e demais receitas diversas de acordo com o padrão da Federação Brasileira de Bancos Febraban com prestação de contas por meio magnético.
000363007/2015	Pref. Angra dos Reis	Obrigações tributárias e contributivas	Ministrio da Fazenda	MM 219 2018 SFI DPTES pagamento de contribuição do Pasep referente ao mês de novembro de 2018.
000363011/2010	Pref. Angra dos Reis	Outros serviços de terceiros - PF	Nair Maria Lazaro	Memo nº 001 2018 CGM despesa referente ao complemento de empenho do termo aditivo 010 ao contrato de locação nº 011 2010 com início em 05/02/2017 e término em 04/02/2018 relativo ao imóvel situado na Rua Honório Lima nº 127, Centro, Angra dos Reis

Tabela 4.1: Descrição dos conjuntos de dados de uma Nota de Empenho, apresentando exemplos de contratos, unidades responsáveis, elementos de despesa, credores associados e informações do histórico.

4.2.1 IdContrato

É relevante destacar a importância do campo `IdContrato`, que representa o número identificador de cada contrato associado a uma nota de empenho específica. No entanto, observa-se que a base contém 190.776 itens de empenho com o campo `IdContrato` preenchido. Isso representa apenas 12,84% do total da base. Por esse motivo, optamos por descartar esse campo no treinamento do modelo.

A Figura 4.2 apresenta a distribuição de frequências dos identificadores de contrato na base de dados. Destaca-se que a escala logarítmica foi adotada para melhor visualização e que, a partir da frequência 10, os dados foram agrupados em intervalos progressivos. Observa-se que a maioria dos identificadores de contrato está concentrada no intervalo de 1 a 20 ocorrências, abrangendo a maioria dos Itens de Empenho associados. Além disso, os identificadores com frequências entre 20 e 50 ainda apresentam quantidades significativas, embora, a partir desse ponto, a frequência das ocorrências diminui gradativamente.

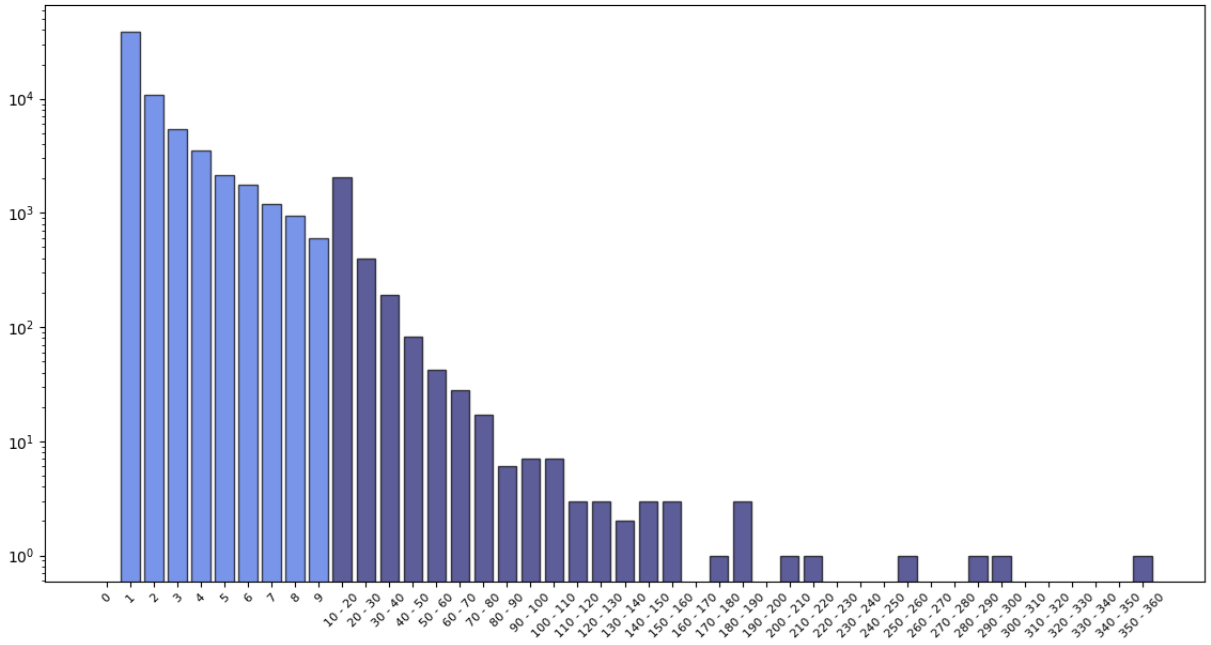


Figura 4.2: Gráfico que representa a distribuição da frequência de ocorrência dos valores de `IdContrato` na base de dados. No eixo x, cada barra indica a quantidade de identificadores que aparecem com uma determinada frequência. A primeira barra, por exemplo, corresponde aos `IdContrato` que ocorrem apenas uma vez, sendo essa a situação mais comum.

A Tabela 4.2 apresenta, de forma numérica, os 10 intervalos mais recorrentes exibidos na Figura 4.2. Cada linha traz a respectiva porcentagem, indicando a frequência em relação ao total de identificadores de contratos válidos na base.

Ocorrências	Quantidade	Porcentagem
1	39.072	57,24%
2	10.775	15,78%
3	5.418	7,94%
4	3.488	5,11%
5	2.145	3,14%
6	1.767	2,59%
7	1.203	1,76%
8	952	1,39%
9	603	0,88%
10-20	2.034	2,98%

Tabela 4.2: Tabela de frequências dos identificadores de contrato na base de dados, apresentando a quantidade de ocorrências em cada intervalo e sua respectiva porcentagem em relação ao total.

4.2.2 Unidade

A variável `Unidade` representa os diferentes órgãos ou entidades responsáveis pelos Itens de Empenho registradas na base de dados. Observa-se que este conjunto de dados não possui dados faltantes, logo, contém um total de **1.484.918** registros. Observa-se, também, que existem **771** categorias para esta variável, indicando que a grande maioria das unidades está associada a 2 ou mais Itens de Empenho.

A Figura 4.3 apresenta a distribuição de frequências das unidades por intervalo. A escala logarítmica foi adotada no eixo y para melhor visualização. Nos intervalos iniciais, agregados de cada 100 (representados pela coloração verde), há mais Itens de Empenho associados. Já nos intervalos intermediários, agregados a cada 500 (representados pela coloração amarela), percebe-se uma variação mais acentuada na quantidade de Itens de Empenho associados, iniciando-se com valores elevados e reduzindo-se progressivamente. Nos intervalos de frequência mais elevados, agregados a cada 1.000 e destacados na coloração vermelha, observa-se uma redução na quantidade de Itens de Empenho associados, um comportamento esperado devido à distribuição dos dados.

Ao analisar os intervalos de frequência mais elevados, destacados em vermelho na Figura 4.3, a Tabela 4.3 apresenta as 10 unidades mais recorrentes na base de dados, evidenciando a sua frequência absoluta e sua respectiva porcentagem em relação ao total de Itens de Empenho. A predominância dessas unidades reforça a importância de avaliar o impacto dessas ocorrências nas análises futuras, uma vez que podem influenciar significativamente a modela-

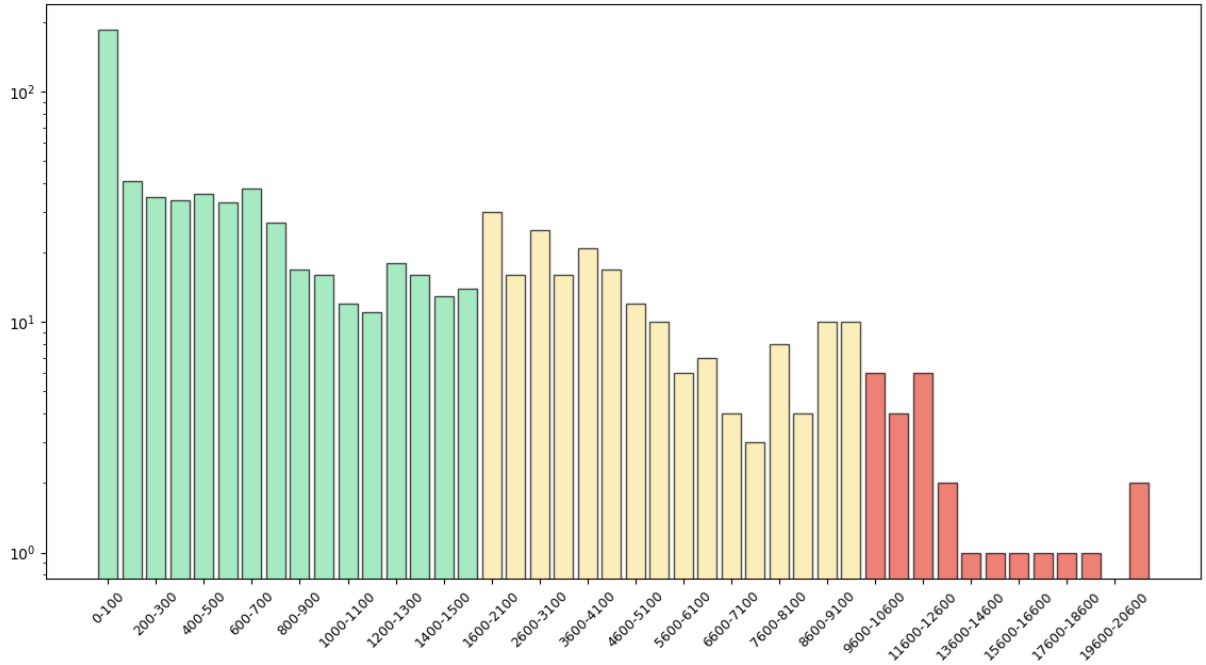


Figura 4.3: A distribuição de frequências das unidades ilustra a quantidade de ocorrências por intervalo definido. No eixo x, são apresentados os intervalos de frequência, agregados de forma diferenciada: para valores menores, utiliza-se uma granularidade mais fina, enquanto para valores mais altos, os intervalos são ampliados. Já o eixo y, em escala logarítmica, representa o número de unidades que pertencem a cada intervalo de frequência na base de dados, permitindo uma melhor visualização da distribuição.

gem dos dados e a interpretação dos resultados.

Unidade	**Frequência**	**Porcentagem**
FUNDO MUN SAÚDE RESENDE	21.108	1,42%
PREFEITURA QUISSAMÃ	20.860	1,40%
PREFEITURA VALENÇA	19.434	1,31%
PREFEITURA RIO DAS OSTRAS	18.001	1,21%
FUNDO MUN SAÚDE BOM JESUS ITABAPOANA	17.208	1,16%
PREFEITURA TRÊS RIOS	16.351	1,10%
FUNDO MUN SAÚDE TRÊS RIOS	15.547	1,05%
PREFEITURA MARICÁ	14.284	0,96%
PREFEITURA SANTO ANTÔNIO DE PÁDUA	13.233	0,89%
FUNDO MUN SAÚDE NOVA IGUAÇU	12.760	0,86%

Tabela 4.3: Top 10 unidades com maior frequência de registros, destacando as entidades com maior volume de ocorrências no conjunto de dados. A Tabela apresenta a quantidade absoluta de registros para cada unidade, bem como a respectiva porcentagem em relação ao total, permitindo uma melhor visualização da representatividade de cada entidade no conjunto de dados

4.2.3 ElemDespesaTCE

O campo `ElemDespesaTCE` representa as diferentes categorias de despesas registradas nos Itens de Empenho da base de dados. Observa-se que o conjunto de dados contém um total de **1.484.918** registros para esta variável, totalizando **129** distintos, ou seja, categorias diferentes para esta variável. Observa-se que não existem valores faltantes, garantindo a completude das informações e evitando a necessidade de imputação ou remoção de dados para análise.

A distribuição de frequências dos Elementos de Despesa do TCE, ilustrada na Figura 4.4, revela um padrão característico. Observa-se, nos intervalos esverdeados, de agregação a cada 100, um pico de frequência inicial, para o primeiro intervalo de até 100 ocorrências e uma grande diminuição nos seguintes. Para os intervalos amarelados, de agregação a cada 1000, nota-se um leve aumento de ocorrências, sem muita variação. Entretanto, nota-se que há um conjunto significativo de elementos que apresentam frequências mais elevadas, destacando-se na faixa alaranjada, de agregação a cada 5000, com ocorrências que variam de entre 11.000 e 31.000 repetições para alguns elementos da despesa. Além disso, identificam-se elementos de despesa com níveis de repetição extremamente altos, representados pelos segmentos em vermelho, de agregação a cada 10.000, que abrangem frequências de 41.000 até 321.000 ocorrências.

A Tabela 4.4 apresenta os 10 elementos de despesa mais recorrentes na base de dados, todos caracterizados por um volume significativo de registros.

Elemento Despesa TCE	Frequência	Porcentagem
Outros serviços de terceiros - Pessoa jurídica	316.956	21,35%
Material de consumo	273.737	18,43%
Outros serviços de terceiros - Pessoa física	152.944	10,30%
Vencimentos e vantagens fixas - Pessoal civil	103.695	6,98%
Diárias - Civil	102.334	6,89%
Vencimentos e vantagens fixas - Pessoal civil	46.881	3,16%
Sentenças judiciais	40.732	2,74%
Contribuição para o Regime Geral de Previdência (INSS)	33.885	2,28%
Equipamentos e material permanente	32.622	2,20%
Contratação por tempo determinado	31.202	2,10%

Tabela 4.4: Top 10 elementos de despesa do TCE mais frequentes, destacando a principal categoria de gastos registrada, sua respectiva frequência e participação percentual no total.

Observa-se que apenas o primeiro elemento de despesa corresponde a mais de 21% do total de registros de Itens de Empenho, representando uma proporção significativamente elevada na base de dados. Esse alto percentual pode impactar consideravelmente a modelagem dos dados e

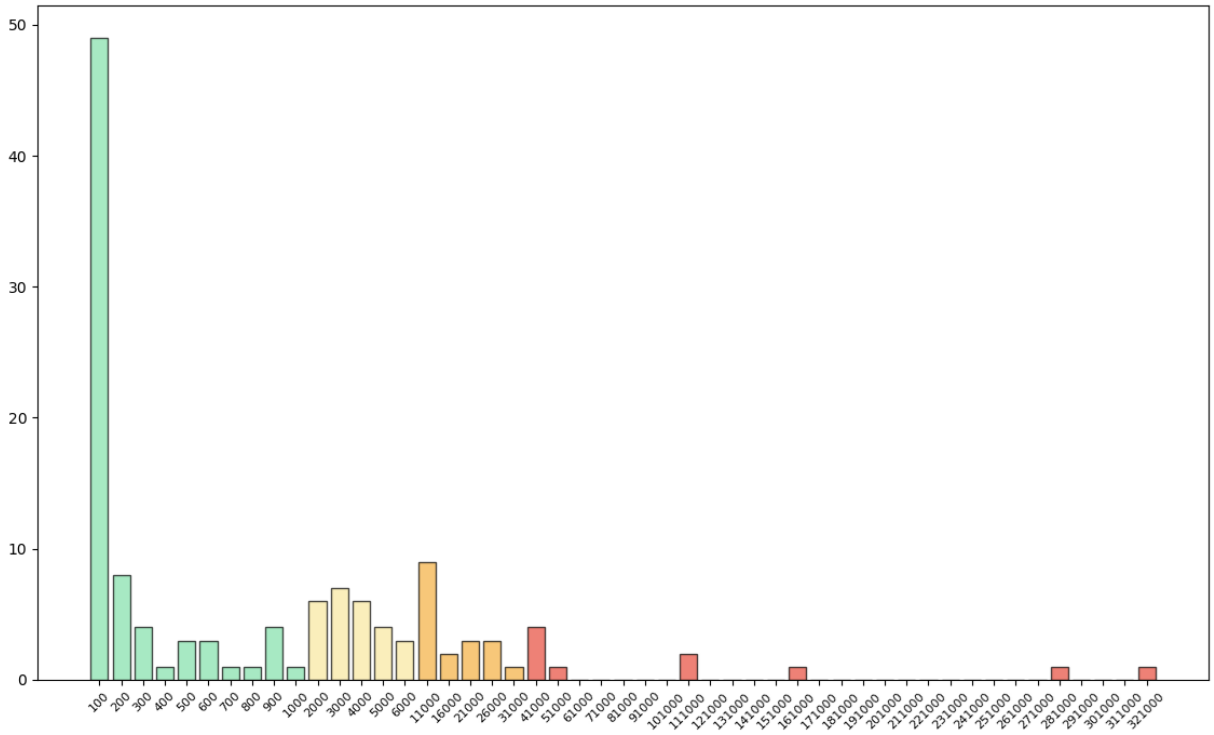


Figura 4.4: Distribuição de frequências dos elementos de despesa do TCE, ilustrando a quantidade de ocorrências em cada intervalo de frequência. No eixo y, é apresentada a quantidade de elementos de despesa que pertencem a cada intervalo. Já o eixo x representa os intervalos de frequência, definidos de acordo com uma segmentação progressiva para melhor visualização da distribuição. No gráfico, quanto mais tendente ao vermelho, maior a segmentação.

a interpretação dos resultados. Além disso, os 10 elementos de despesa mais frequentes somam, conjuntamente, 76,43% dos registros da base, evidenciando uma forte concentração nesses tipos de despesas.

4.2.4 Histórico

O campo `Histórico` descreve cada item de empenho. Esse campo desempenha um papel essencial no processo de agrupamento, sendo utilizado como variável de entrada do modelo. Observa-se que não há valores faltantes nesta variável e que existem 947.140 categorias distintas deste campo, o que indica uma alta variação.

A Figura 4.5 apresenta a distribuição das frequências de ocorrência por intervalo. Observa-se que a grande maioria dos registros concentra-se no intervalo de 1 a 20 ocorrências, enquanto a frequência tende a diminuir conforme os intervalos aumentam. Destaca-se que a escala logarítmica foi adotada no eixo y para melhor visualização da base de dados completa e que, a partir de 20 ocorrências, os dados são agregados em intervalos a cada 10, representados na cor ama-

rela. Para frequências superiores a 50, a agregação ocorre em intervalos a cada 50, sob o tom alaranjado, e para valores acima de 1000, os intervalos passam a ser a cada 1000, destacados em vermelho.

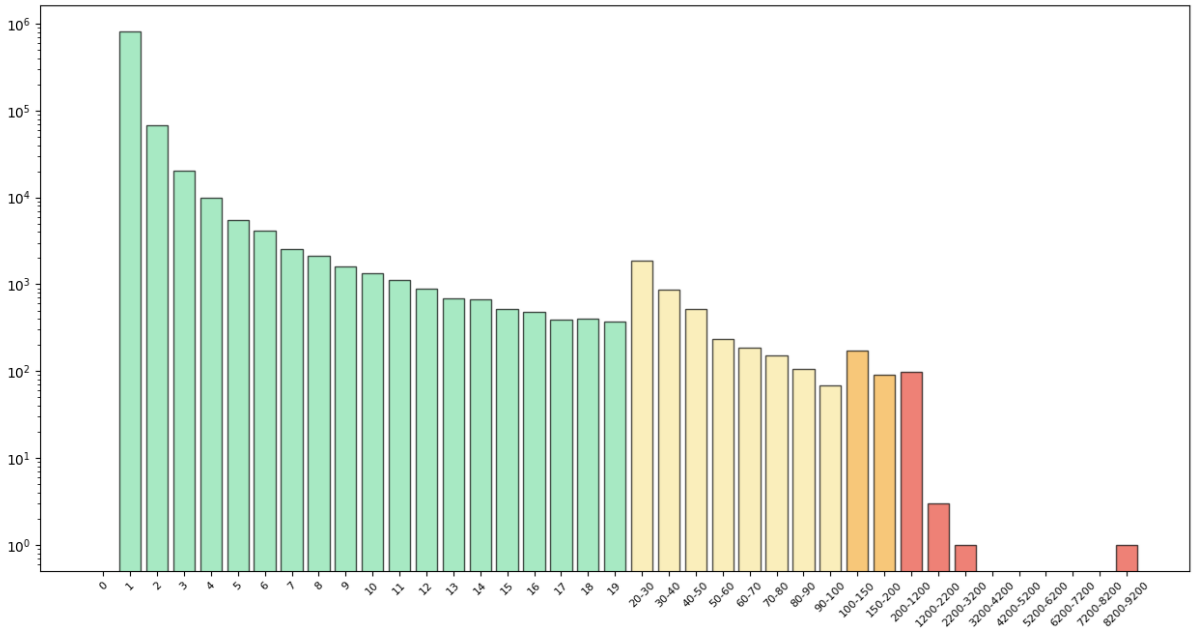


Figura 4.5: Distribuição de frequências das ocorrências do campo `Historico`, ilustrando a quantidade de registros em cada intervalo de frequência. As cores representam diferentes faixas de agregação: amarelo (intervalos de 10), laranja (intervalos de 50) e vermelho (intervalos de 1000).

Nota-se que os intervalos com menor repetição de `Historico` para diferentes Itens de Empenho são os que apresentam maior frequência. A Tabela 4.5 mostra que o primeiro intervalo, que correspondente a ocorrências que repetem somente uma vez, é o mais significativo, totalizando 822.378 registros, o que representa 55,38% da base de dados.

Intervalo	Frequência	Porcentagem
1	822.378	55,38%
2	67.653	4,56%
3	20.078	1,35%
4	9.843	0,66%
5	5.506	0,37%
6	4.108	0,28%
7	2.544	0,17%
8	2.132	0,14%
9	1.608	0,11%
10	1.341	0,09%

Tabela 4.5: Tabela apresentando a distribuição das repetições na base de dados, com intervalos, frequência absoluta e porcentagem relativa.

Além disso, verifica-se que alguns valores do campo `Historico` apresentam frequências consideravelmente altas. Para uma análise mais detalhada, a Tabela 4.6 apresenta as categorias mais recorrentes, juntamente com sua frequência absoluta e a porcentagem de ocorrência em relação ao total da base de dados.

Descrição	Frequência	Porcentagem
Folha de pagamento	8.227	0,55%
Verba escolar para aplicação em até 60 dias	2.647	0,18%
Prestação de serviços como autônomo para a Secretaria Municipal de Saúde	1.962	0,13%
Diária conforme Lei Municipal n.º 368/1996 e decretos posteriores	1.520	0,10%
Diária conforme Lei Municipal n.º 368/1996, decreto n.º 041/1996 e alterações	1.332	0,09%

Tabela 4.6: Tabela das 5 descrições mais frequentes na base de dados, apresentando a frequência absoluta e a participação percentual no total.

A Figura 4.6 apresenta a distribuição da frequência em função do número total de palavras nesse campo. Observa-se que a distribuição apresenta um pico entre 11 e 14 palavras, seguido de uma redução gradual até aproximadamente 34 palavras. Curiosamente, a frequência volta a aumentar até atingir 38 palavras, antes de sofrer uma nova redução progressiva até o final da distribuição. Além disso, a análise revela uma média aritmética de 22,71 palavras, com um desvio padrão de 11,65 palavras.

4.2.5 Credor

O campo `Credor` representa o fornecedor específico dado um Item de Empenho. Observa-se que neste campo não existem dados faltantes, isto é, todos os Itens de Empenho possuem o seu respectivo fornecedor preenchido. Nota-se que a base possui 127.243 credores distintos.

AFigura 4.7 ilustra a distribuição das frequências de ocorrência dos credores na base de dados. No eixo x, estão representadas as diferentes faixas de frequência, agrupadas em intervalos agregados, enquanto no eixo y é exibida a quantidade de credores correspondente a cada intervalo. Para facilitar a análise e lidar com a grande variação entre os valores mais e menos frequentes, foi aplicada uma escala logarítmica ao eixo y. Na figura, os 10 primeiros intervalos são destacados em azul, sem agregação. Como os credores mais frequentes são aqueles que menos se repetem na base de dados, eles se concentram nesse intervalo, não havendo necessidade de agrupamento intervalar. A partir desse ponto, os dados passam a ser agregados em intervalos

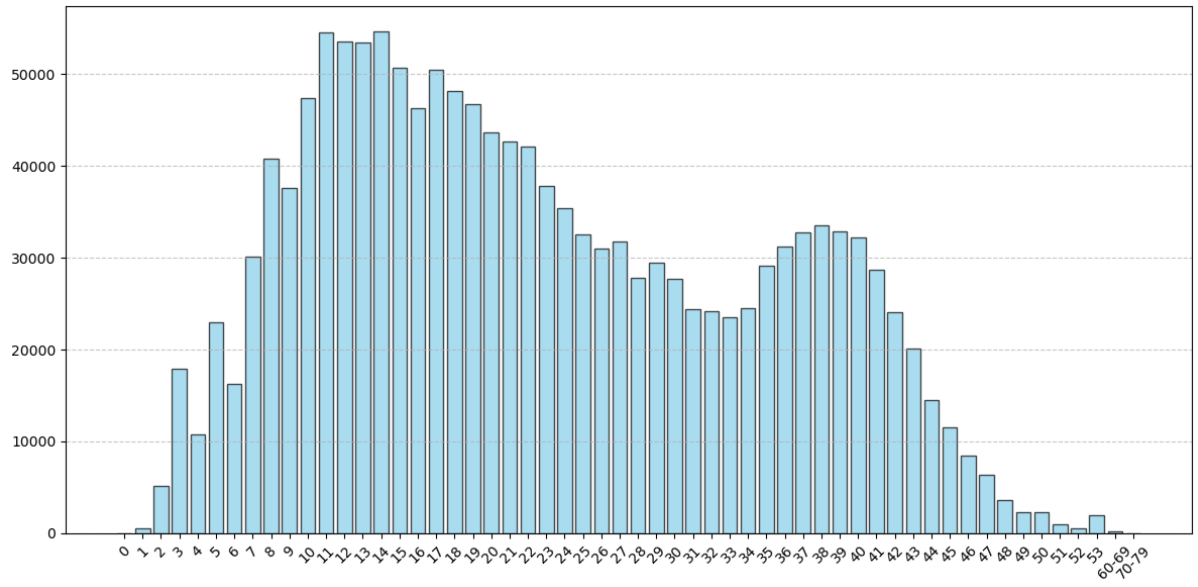


Figura 4.6: Distribuição de frequências do comprimento das sentenças no campo `Historico`, ilustrando a variação no tamanho das sentenças (número de palavras no eixo x e no eixo y a frequência em que aparecem na base de dados dos Itens de Empenho.

a cada 2, representados em verde. Para frequências superiores a 30, a agregação ocorre em intervalos a cada 5, em amarelo. Quando a frequência ultrapassa 55, os intervalos passam a ser a cada 25, representados em laranja. Finalmente, para frequências a partir de 230, os valores são agrupados em intervalos a cada 8.000, destacados em vermelho. Esse último agrupamento tem o propósito de concentrar credores com um número excepcionalmente alto de ocorrências, os quais podem ser considerados *outliers*. Observa-se que os credores estão majoritariamente concentrados nos intervalos destacados em verde e amarelo, cujas ocorrências estão, em média, acima de 10^3.

A partir da Figura 4.7, observa-se que os intervalos com menor repetição de credores para diferentes Itens de Empenho apresentam as maiores frequências. A Tabela 4.7 evidencia numericamente que os primeiros intervalos são os mais representativos. Ao somar as ocorrências de credores que aparecem apenas uma vez ou até duas vezes, chega-se a 60,85% do total, indicando que mais da metade das ocorrências correspondem a fornecedores que realizam transações pontuais, em vez de contratos recorrentes.

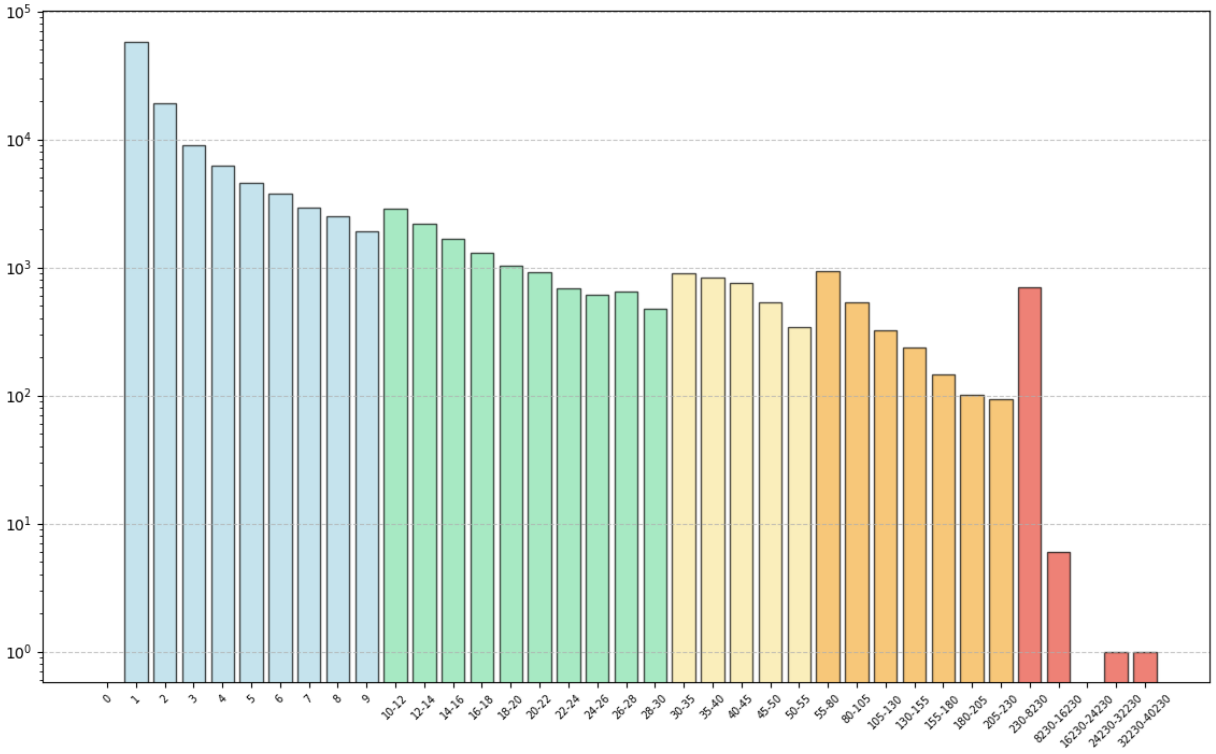


Figura 4.7: Distribuição da quantidade de ocorrências dos credores na base de dados, destacando a variabilidade na frequência com que os fornecedores aparecem nos registros.

Intervalo	Frequência	Porcentagem
1	58.236	45,77%
2	19.183	15,08%
3	9.025	7,09%
4	6.254	4,92%
5	4.563	3,59%
6	3.797	2,98%
7	2.932	2,30%
8	2.527	1,99%
9	1.900	1,49%
10	1.604	1,26%

Tabela 4.7: Tabela apresentando a distribuição das repetições dos credores na base de dados, com intervalos, frequência absoluta e porcentagem relativa.

Além disso, a Tabela 4.8 apresenta os 10 credores mais recorrentes na base de dados. O fornecedor mais frequente é o Instituto Nacional do Seguro Social, com 35.605 registros, correspondendo a 2,40% do total da base. Observa-se que a tabela também exibe a porcentagem de ocorrência de cada `Credor` em relação ao volume total de registros, evidenciando a concentração dos principais fornecedores.

Credor	Frequência	Porcentagem
Instituto Nacional do Seguro Social	35.605	2,40%
Tribunal de Justiça do Estado do Rio de Janeiro	25.786	1,74%
Fundo Municipal de Saúde	14.270	0,96%
Telemar Norte Leste S.A. em Recuperação	13.072	0,88%
Caixa Econômica Federal	11.771	0,79%
Ampla Energia e Serviços S.A.	11.288	0,76%
Prefeitura Municipal de Rio das Ostras	9.711	0,65%
Light Serviços de Eletricidade S.A.	8.340	0,56%
Instituto de Previdência Comendador Levy Gasparian	6.716	0,45%
Despesa com Pessoal - FMS	5.955	0,40%

Tabela 4.8: Lista dos 10 credores mais frequentes na base de dados, acompanhados de suas respectivas frequências absolutas e percentuais.

4.2.6 Valor Empenhado

Embora o campo `Vlr_Empenhado` não seja utilizado como variável de entrada do algoritmo usado neste presente estudo, é importante colocá-lo na análise exploratória da base de dados. Ela representa o valor reservado no orçamento para cobrir uma despesa pública prevista, ao contrário de `Vlr_Liquidado`, que se refere ao valor da despesa efetivamente realizada e reconhecida como devida pela Administração Pública.

Conforme ilustrado na Figura 4.8 e na Figura 4.9, observa-se que os intervalos de menor valor empenhado concentram tanto uma maior frequência de registros quanto um maior valor total empenhado por intervalo definido.

Nas figuras mencionadas, a intensidade da cor — com tons mais próximos do vermelho — indica faixas de intervalos de valor empenhado mais elevados. Ainda assim, mesmo com a ampliação dos intervalos superiores, percebe-se que a concentração de empenhos (em quantidade e em montante financeiro) permanece mais expressiva nos intervalos de até R$1 milhão, representando 1.466.934 Itens de Empenho ou 98,80% de todos os registros, o que indica uma grande importância aos empenhos de menor valor.

Além da análise da distribuição do Valor Empenhado, observa-se também a Figura 4.10, que aprofunda a investigação sobre a composição das variáveis associadas a esses intervalos. A figura apresenta a Entropia de Shannon dos campos `Unidade`, `ElemDespesaTCE` e `Credor` por intervalo de `Vlr_Empenhado`, permitindo avaliar o grau de diversidade dessas categorias em diferentes faixas de valor. Quanto mais próximo de zero estiver a entropia, menor será a diversidade, isto é, menos distribuídas estão as unidades ou elementos da despesa dentro de um

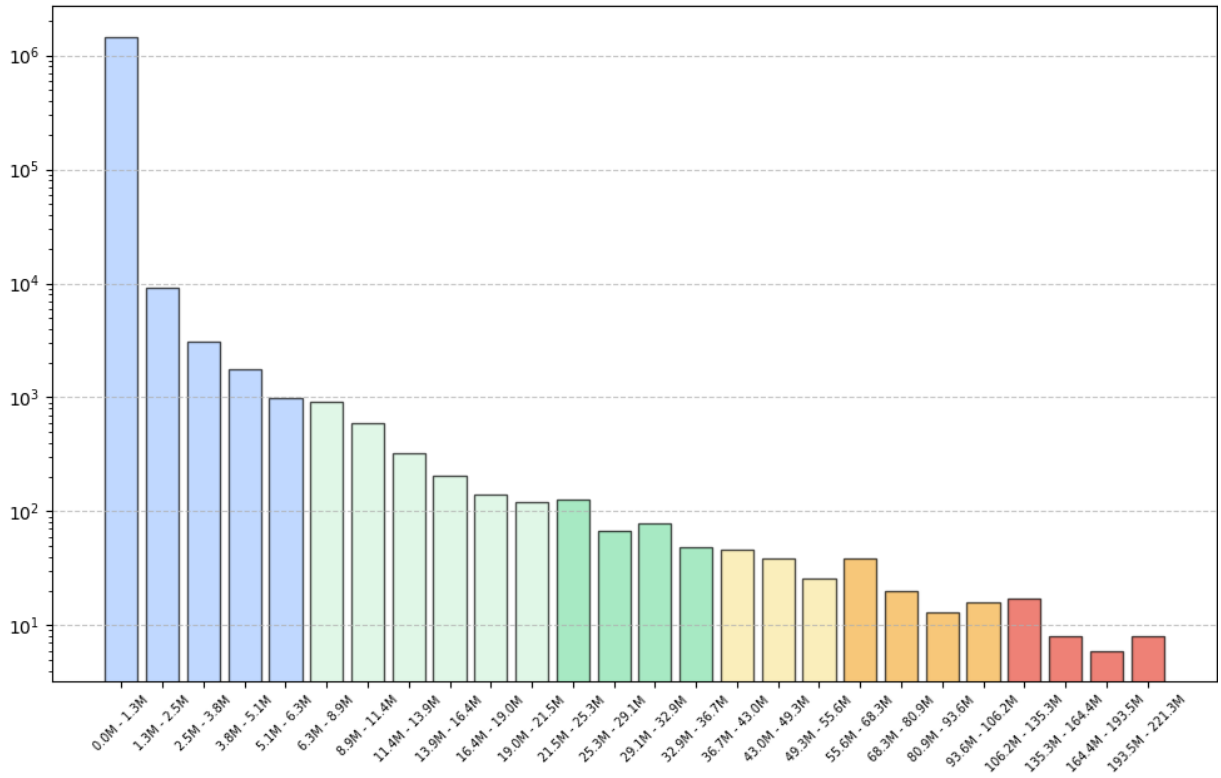


Figura 4.8: Distribuição da frequência de registros conforme os intervalos de valor empenhado. Observa-se maior concentração de registros nos intervalos de menor valor, especialmente até R$1,3 milhão. A intensidade da cor indica a magnitude do intervalo de valor empenhado.

dado intervalo de valor empenhado.

A pontuação de entropia está limitada ao intervalo $0 \leq H(p) \leq \log_2(N)$, em que N representa o número total de categorias distintas no campo analisado. É importante destacar que, para todos os intervalos de `Vlr_Empenhado`, o número total de categorias permanece constante: 771 unidades, 129 elementos de despesa e 127.243 credores, resultando em limites máximos teóricos de entropia de $log_2(771) = 9,59$, $log_2(129) = 7,01$ e $log_2(127.243) = 16,95$, respectivamente.

Observa-se que as pontuações de entropia para os campos `Unidade` e `Credor` iniciam em valores relativamente elevados, com 5,76 e 9,19, respectivamente, e diminuem à medida que os intervalos de valor empenhado aumentam, estabilizando-se próximos a 3,80 para as unidades e 4,30 para os credores. O fato de a entropia ser significativamente menor que o valor máximo teórico sugere que poucas categorias predominam nas faixas de valores mais altos.

Já para o campo `ElemDespesaTCE`, as pontuações de entropia permanecem consistentemente baixas em relação ao seu máximo teórico, independentemente do intervalo analisado. Esse comportamento indica uma baixa diversidade na classificação dos elementos da despesa,

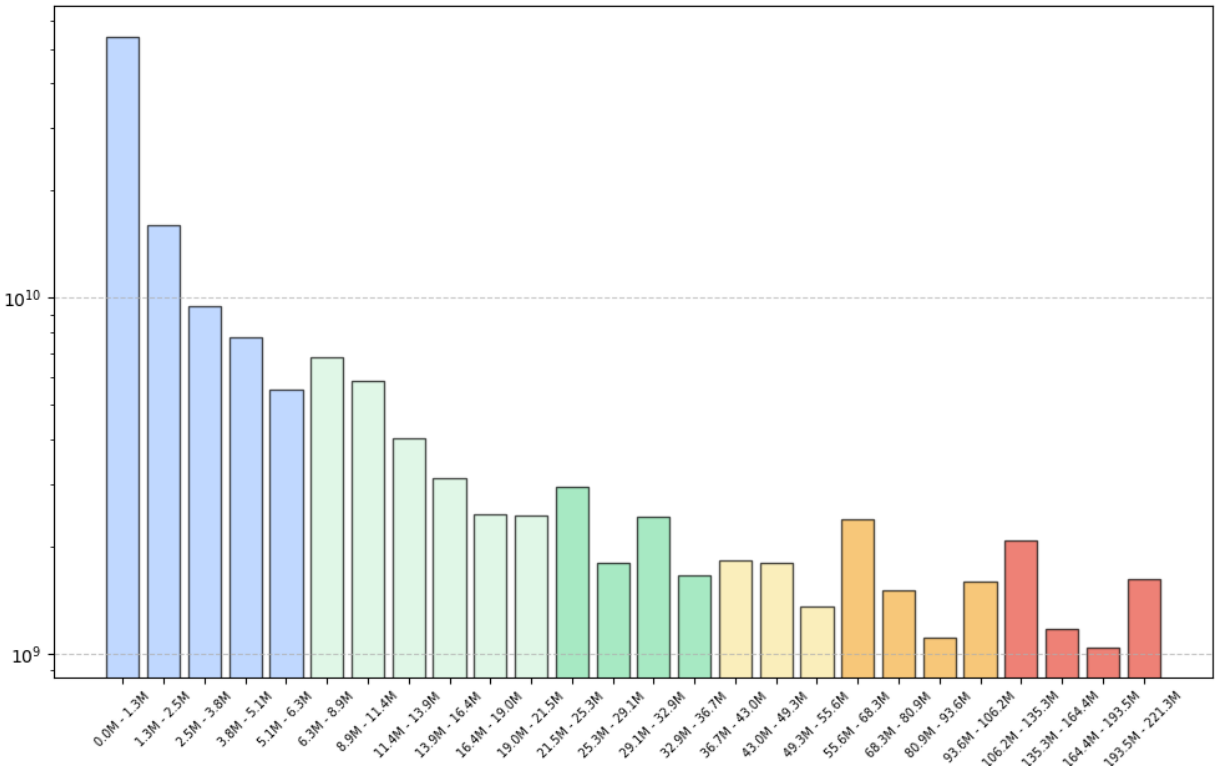


Figura 4.9: Distribuição do valor total empenhado por intervalo. Apesar da existência de intervalos superiores, a maioria do montante empenhado concentra-se nos intervalos de menor valor, evidenciando a predominância de empenhos de até R$1,3 milhão. Os intervalos em azul foram definidos conforme o desvio padrão de `Vlr_Empenhado`. Já os demais intervalos sofreram uma agregação conforme a intensidade da cor vai se aproximando ao vermelho.

mesmo entre diferentes faixas de valor.

No entanto, é importante destacar que, conforme observado na Figura 4.8, o primeiro intervalo, correspondente a valores empenhados de até R$1,3 milhão, concentra 98,80% das observações, tornando-se o mais representativo do conjunto de dados.

4.3 Pré-processamento dos dados

Como preparação para esta etapa, a base de dados foi convertida para o formato *Parquet*, que oferece melhor desempenho devido à sua eficiência na compactação e na leitura de grandes volumes de dados. Para isso, utilizamos a biblioteca *Pandas* para ler o arquivo e convertê-lo para o formato desejado. Esse processo foi realizado com programação paralela, devido ao grande volume de dados, garantindo maior eficiência no tempo de execução.

Para viabilizar o processamento desses dados na próxima etapa, onde eles são utilizados como entrada para o modelo, é necessário convertê-los para um formato que o modelo possa interpretar. Nesse processo, os dados do campo `Histórico` são transformados em *embeddings*,

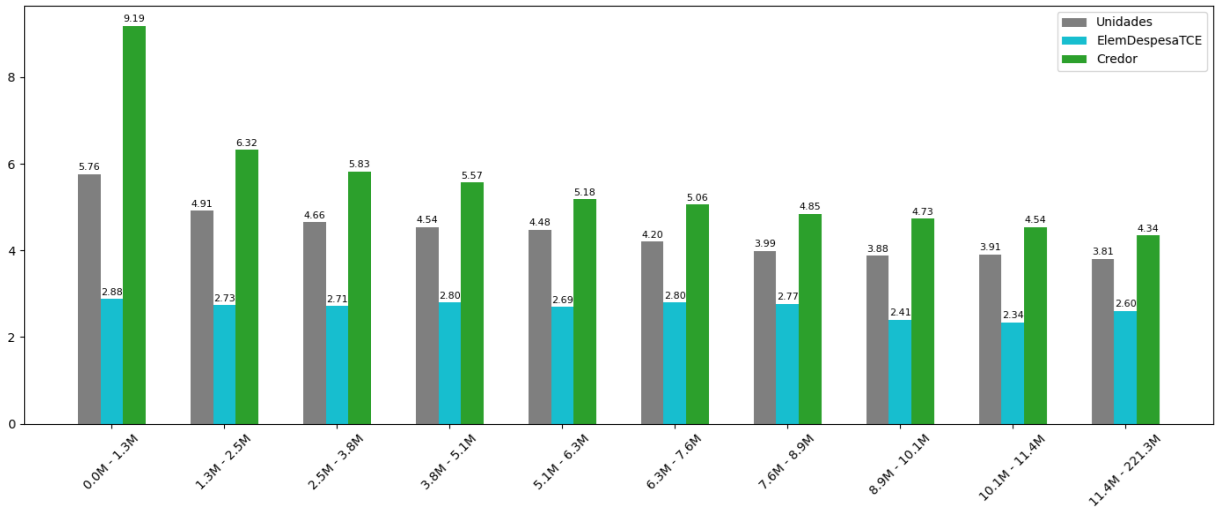


Figura 4.10: Entropia de Shannon dos campos `Unidade`, `ElemDespesaTCE` e `Credor` calculada para cada intervalo de valor empenhado. A figura evidencia como a diversidade de unidades responsáveis e elementos da despesa varia de acordo com a faixa de valor empenhado. O décimo e último intervalo apresentado possui a faixa mais ampla, para fins de melhor visualização.

o que permite representar as informações textuais de maneira densa e estruturada, adequada para o treinamento do modelo.

Para a transformação em *embeddings*, utilizamos um modelo pré-treinado do framework SBERT, descrito por Reimers and Gurevych [2019]. Este framework é considerado um dos mais avançados (state-of-the-art) para a transformação de sentenças, textos e imagens. Os *embeddings* de texto são gerados por meio de programação paralela, processando os dados em lotes (*batches*), devido ao grande volume de dados. Esses *embeddings* são então armazenados em arquivos no formato *NumPy* (.npy), facilitando o carregamento eficiente na etapa subsequente do processo.

4.3.1 Classe Empenhos

A próxima etapa de pré-processamento é executada pela classe *EMPENHOS*, que herda da classe *Dataset*. Essa herança permite estruturar e organizar os dados de forma compatível com o framework utilizado neste trabalho. Com isso, tornam-se possíveis operações essenciais como carregamento em lotes, embaralhamento e iteração eficiente dos dados durante o treinamento dos modelos. Essa classe está localizada no arquivo *df_empenhos.py*, disponível no repositório descrito por Paraizo [2024]. O algoritmo responsável pelo processamento dos dados invoca essa classe, na qual os procedimentos de pré-processamento são realizados. Inicialmente, a classe carrega em memória os arquivos *NumPy* contendo os *embeddings*, com

dimensão $(1.484.918, 384)$.

Logo em seguida, os campos `Unidade`, `ElemDespesaTCE` e `Credor` são transformados por meio de codificação por frequência, utilizando o método *value_counts* da biblioteca *Pandas*. Esse processo converte as categorias nominais em suas respectivas proporções de ocorrência na base de dados, viabilizando a representação desses atributos em formato numérico.

Após a codificação por frequência, todas as variáveis são normalizadas com o `StandardScaler`, da biblioteca *sklearn.preprocessing*, que transforma os dados de forma que apresentem média zero e desvio padrão igual a um. Essa normalização é necessária, pois algumas categorias possuem baixas ocorrências, o que gera escalas de frequência significativamente distintas.

Após esse processo, as três variáveis categóricas são empilhadas verticalmente com os *embeddings*, resultando em uma base de dados com dimensão $(1.484.918, 387)$.

Por fim, a classe pode, opcionalmente, realizar a separação do conjunto em dados de treinamento e validação para a etapa do *Autoencoder*, utilizando a proporção de 90% para treinamento e 10% para validação. Essa divisão visa reservar uma quantidade substancial de dados para o treinamento, o que é viável e desejável dada a expressiva dimensão da base.

É possível, também, realizar a separação do conjunto total de dados por meio do método `X_by_elem`, que recebe como parâmetro um índice, podendo variar de 0 a 128, e retorna um subconjunto contendo apenas um elemento específico da despesa (dentre os 129 existentes).

4.4 Etapa de Treinamento

Para este estudo, a aplicação de um algoritmo de agrupamento permite superar as limitações de abordagens tradicionais, como consultas estruturadas em *SQL*, que não são capazes de capturar nuances presentes nos registros textuais e contextuais da base de dados. Além disso, a escolha de um método baseado em técnicas de AP, como o DEC, é fundamental para este estudo, dado a alta dimensionalidade dos dados sendo analisados. Essa abordagem possibilita uma extração mais precisa de padrões e relações semânticas, permitindo uma organização mais estruturada e interpretável das informações.

4.4.1 Aplicação do DEC

Conforme descrito na Seção 2.4.2, o modelo DEC é composto por duas fases principais: a primeira envolve o treinamento do *Autoencoder*, responsável por codificar os dados em um

espaço latente; na segunda, os dados codificados são utilizados na etapa de agrupamento propriamente dito.

O algoritmo utilizado neste estudo foi adaptado a partir do repositório *PT-DEC*, disponível no GitHub Lukiyanov [2019]. A implementação original, desenvolvida com a biblioteca *PyTorch*, foi projetada para realizar o agrupamento do conjunto de dados MNIST, que possui etiquetas reais associadas a seus exemplos. Portanto, para garantir a compatibilidade com os objetivos deste trabalho, foi necessário substituir o MNIST pelo nosso próprio conjunto de dados.

Nesta seção, descrevemos as principais modificações realizadas na implementação original do DEC, visando adaptá-la ao contexto deste trabalho. Além dessas adaptações, também apresentamos extensões e aprimoramentos adicionais desenvolvidos ao longo da implementação.

- **Agrupamento por tipo de elemento da despesa:** a estrutura do algoritmo foi adaptada para possibilitar o treinamento de modelos independentes para cada um dos 129 elementos distintos da despesa, utilizando a divisão de dados realizada pelo método `X_by_elem`.

- **Ajuste dos hiperparâmetros:** valores como taxa de aprendizado, número de épocas, batch size, tamanho das camadas dos SAE, incluindo a camada oculta, foram ajustados empiricamente para melhor adequação ao nosso conjunto de dados, que possui características distintas do MNIST.

- **Métricas de avaliação:** as métricas baseadas em *Clustering Accuracy*, que dependem de rótulos verdadeiros (true labels), foram removidas, já que não são aplicáveis ao nosso problema. Em seu lugar, foi incorporado o cálculo do *Silhouette Score*, uma métrica não supervisionada mais apropriada para avaliar a coesão e separação dos agrupamentos formados.

A execução do código permite a definição de diversas flags importantes por meio de argumentos de linha de comando. Entre elas, destacam-se: a bandeira para uso de `CUDA`, que habilita a execução do código em GPU; a bandeira `train-autoencoder`, que permite o treinamento do *Autoencoder* do zero; e a bandeira `sort-by-elem`, que ativa o treinamento do DEC em subconjuntos de dados correspondentes a um elemento específico da despesa.

Além disso, uma diferença importante entre o uso do MNIST e o conjunto de dados de Itens de Empenho é que, no caso do MNIST, o número de agrupamentos k já é conhecido, pois há rótulos definidos (de 0 a 9). Já no conjunto de Itens de Empenho, essa informação não

está disponível, uma vez que não possuímos etiquetas. Para contornar essa limitação, foram realizados diversos agrupamentos com o algoritmo DEC, utilizando diferentes valores de k. Em seguida, para cada número de agrupamentos, avaliou-se o desempenho por meio da métrica do *Silhouette Score*, visando estimar o número ótimo de agrupamentos.

4.4.2 Agrupamento por Elemento da Despesa

Como mencionado na Seção 4.4.1, é possível dividir o conjunto de dados em até 129 subconjuntos, cada um contendo exclusivamente itens de um único `ElemDespesaTCE`. Essa segmentação permite aplicar o agrupamento de forma individualizada, analisando separadamente os itens de empenho associados a cada categoria de elemento de despesa.

Essa abordagem permite a especialização dos modelos de agrupamento com base nas características específicas de cada tipo de elemento da despesa, tornando o processo mais sensível às particularidades de cada categoria. Ao isolar os dados por elemento da despesa, possibilita-se identificar padrões intrínsecos a cada grupo, permitindo uma análise mais refinada e segmentada do comportamento dos empenhos.

Além disso, o próprio elemento da despesa já atua como um critério de diferenciação relevante entre os registros, e aplicar técnicas de agrupamento dentro de cada subconjunto reforça essa estrutura hierárquica implícita nos dados.

Da mesma forma que no agrupamento realizado com o conjunto completo, também foi conduzida, neste caso, uma análise detalhada para estimar o valor ótimo do parâmetro k, agora aplicada individualmente a cada subconjunto de dados.

No entanto, realizar o treinamento com o algoritmo DEC seria computacionalmente custoso, já que exigiria testar diversos valores de k para cada subconjunto gerado. Por essa razão, optou-se por utilizar o algoritmo *MiniBatchKMeans*, devido à sua natureza mais eficiente em processamento por lotes (mini-batches).

O intervalo de valores testados para k variou conforme o tamanho de cada subconjunto, respeitando sua escala e complexidade. Para subconjuntos menores, o intervalo foi significativamente reduzido, para evitar sobreajuste e agrupamentos artificiais em conjuntos com baixa variabilidade.

Foi estabelecido um limiar mínimo para o *Silhouette Score*; caso o escore máximo obtido para um determinado subconjunto seja inferior a esse limiar, considera-se que não há uma estrutura de agrupamento relevante, e define-se $k_{\text{optimal}} = 1$, ou seja, nenhum agrupamento é

realizado nesse caso. O mesmo critério é aplicado quando o subconjunto contém apenas um Item de Empenho.

Para os casos em que o número ótimo de agrupamentos (k) foi maior que 1, foram gerados gráficos dos escores de *silhouette*, representando a variação do *Silhouette Score* em função dos diferentes valores de k. A análise visual desses gráficos permitiu identificar o valor de k a partir do qual aumentos sucessivos resultam em ganhos marginais no escore. Esse ponto foi considerado o número ideal de agrupamentos para o respectivo subconjunto. A Figura 4.11 apresenta o gráfico que relaciona os diferentes valores de k testados com os respectivos escores de Silhouette, permitindo uma análise mais detalhada sobre o possível valor k ótimo associado a este subconjunto em específico.

Neste exemplo, pelo gráfico, observa-se que o valor $k = 2$ apresenta um escore de Silhouette inicialmente alto, mas há uma queda acentuada ao passar para $k = 3$, com crescimento gradual nos valores seguintes. Esse comportamento, no entanto, pode indicar um possível sobreajuste. O aumento do número de agrupamentos tende a aproximar os pontos de seus respectivos centroides, o que nem sempre reflete uma melhor qualidade no agrupamento.

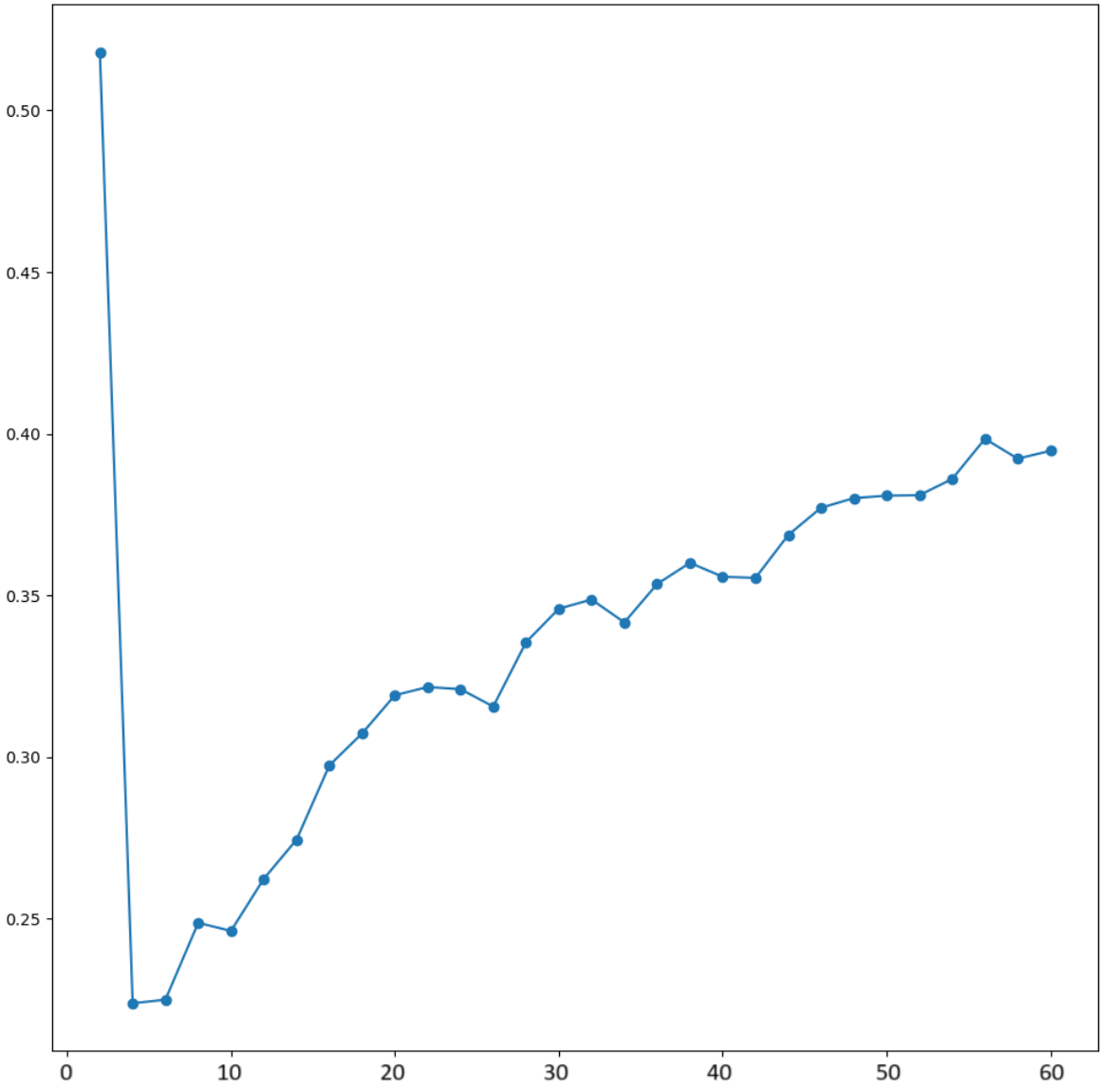


Figura 4.11: Gráfico representando o escore de Silhouette por valor *k*. Este gráfico foi gerado utilizando o primeiro subconjunto do conjunto de dados de Itens de Empenho.

Capítulo 5

Avaliação experimental

Neste capítulo, apresentamos os experimentos realizados e os resultados obtidos. Na Seção 5.1, descrevemos as configurações de hardware e software utilizadas. Em seguida, na Seção 5.2, detalhamos os experimentos conduzidos, destacando os hiperparâmetros adotados e suas influências nos resultados. O objetivo desta avaliação é validar a eficácia da abordagem proposta, analisando a capacidade do modelo de agrupamento dos dados, com ênfase na métrica *Silhouette Score*.

5.1 Configuração de Software e Hardware

Para a realização dos experimentos, foi utilizada a linguagem Python, versão 3.11. Para garantir a compatibilidade e evitar erros durante a execução do código, foi criado um ambiente virtual com *Conda*. Para baixar todas as bibliotecas necessárias para a execução do código, foi criado um arquivo *requirements.txt*, que fornece o nome e a versão das bibliotecas.

Os experimentos foram conduzidos em duas máquinas distintas. Uma delas, denominada *Librae*, possui um processador Intel® Core™ i9-9900K, CPU @ 3.60GHz, com 8 núcleos físicos e 16 threads, arquitetura *x86_64*, suporte aos modos de operação de 32 e 64 bits, e tamanho de endereçamento de 39 bits físicos e 48 bits virtuais. Essa máquina opera com ordem de bytes *Little Endian* e conta com 2 GPUs compatíveis com CUDA, utilizando o driver NVIDIA versão 555.42.02 e CUDA versão 12.5. Para lidar com o grande volume de dados, especialmente durante o treinamento do *Autoencoder* empregou-se a programação em GPU, possibilitando a paralelização das operações de multiplicação de matrizes. A outra máquina é um laptop pessoal com o sistema operacional Windows 11, equipada com um processador Intel® Core™ i7-8565U, de 1,80 GHz, 8 GB de memória RAM, 4 núcleos e 8 processadores lógicos.

5.2 Resultados

Nesta seção, apresentamos e discutimos os resultados obtidos nos experimentos realizados com o *Autoencoder* e as duas abordagens de agrupamento de dados propostas. A primeira abordagem considera a base de dados completa como um único conjunto; a segunda realiza a agrupamento de forma segmentada, ou seja, separadamente para cada tipo de elemento da despesa.

Cabe destacar que todos os hiperparâmetros utilizados nos experimentos estão definidos no arquivo `config.yaml`, o que permite a reprodução dos resultados e facilita eventuais reajustes parametrizados em todo o código-fonte.

5.2.1 Experimentação com Autoencoder

O *Autoencoder* recebe como entrada os *embeddings* do campo `Historico`, concatenados com as representações codificadas por frequência (*frequency encoding*) dos demais campos selecionados.

A extração das representações vetoriais do campo `Historico` foi realizada utilizando o modelo `sentence-transformers/all-MiniLM-L12-v1`, pertencente à família de modelos *Sentence-Transformers* (SBERT). A escolha deste modelo justifica-se por sua eficiência na captura de relações semânticas com baixo custo computacional, sendo especialmente adequado para tarefas envolvendo grandes volumes de dados textuais. Podemos definir os hiperparâmetros para a geração dos *embeddings*:

- **Modelo de *embedding*:** `sentence-transformers/all-MiniLM-L12-v1`
- **Tamanho do batch para *embeddings*:** 128
- **Dimensão dos *embeddings*:** 384

Para a definição da arquitetura do *Autoencoder*, foi necessário um extenso processo de experimentação empírica a fim de determinar o número ideal de camadas e a quantidade adequada de neurônios por camada. Essa etapa foi essencial, considerando que o repositório original estava ajustado para o conjunto MNIST, cuja dimensionalidade é 784, ao passo que, neste trabalho, a entrada possui dimensão 387.

Durante o treinamento, utilizou-se um *data_iterator* com a biblioteca `tqdm`, permitindo o monitoramento, a cada época, da função de custo (*loss*) e da *validation loss*. Essa visualização contínua foi particularmente útil durante a fase de pré-treinamento do *Autoencoder*, na qual as camadas são treinadas individualmente. Com isso, foi possível ajustar de maneira mais criteriosa os hiperparâmetros da rede, incluindo a taxa de aprendizado (*learning rate*).

Podemos definir os hiperparâmetros para o *Autoencoder*:

- **Tamanho do lote** (*batch size*): 256
- **Épocas de pré-treinamento**: 50
- **Épocas de ajuste fino** (*finetuning*): 100
- **Dimensão da entrada** (`input_dim`): 387
- **Camadas intermediárias**: [300, 300, 1000]
- **Dimensão da camada latente** (*hidden layer*): 10
- **Otimizador**: SGD (*Stochastic Gradient Descent*)
- **Taxa de aprendizado** (*learning rate*): 0.1
- **Momentum**: 0.9
- **Taxa de corrupção** (*corruption rate*): 0.2

Ao final das etapas de pré-treinamento e treinamento, o valor da métrica MSE atingiu 0.0083 para o modelo final, o que indica que a reconstrução dos dados de entrada a partir da representação latente foi realizada com baixo erro médio quadrático. Esse resultado evidencia a capacidade do *Autoencoder* em capturar as características principais dos dados, comprimindo informações relevantes de maneira eficiente.

Além do modelo principal, foi realizado também o treinamento de um *Autoencoder* com dimensão da camada latente igual a 3, visando unicamente possibilitar a representação gráfica dos *embeddings* de entrada. Para isso, foi utilizado o *TensorBoard Embedding Projector*, ferramenta disponibilizada pelo Keras que permite a visualização interativa de representações vetoriais de alta dimensionalidade.

Essa visualização é especialmente útil para investigar a organização semântica dos dados no espaço latente, contribuindo para uma compreensão mais intuitiva dos agrupamentos formados nas camadas internas do modelo. O gráfico gerado pelo plugin é interativo, permitindo explorar os pontos em diferentes ângulos e perspectivas. Essa interface pode ser acessada via `localhost:6006` durante a execução do *TensorBoard*. Um recorte da visualização pode ser observado na Figura 5.1.

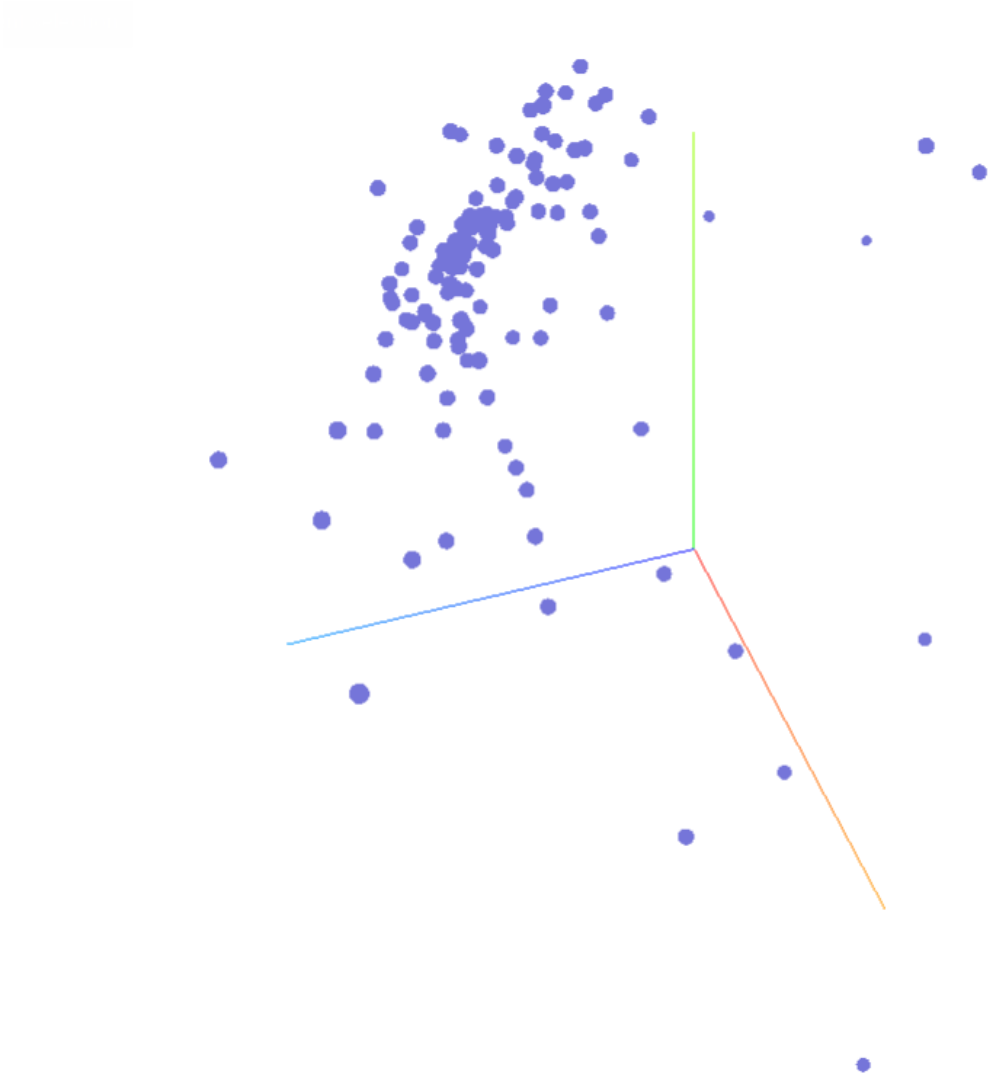


Figura 5.1: Visualização tridimensional dos *embeddings* projetados por meio da camada latente de dimensão 3 do *Autoencoder*, utilizando o plugin *Embedding Projector* do TensorBoard. A interface interativa permite explorar a organização espacial dos dados em diferentes ângulos, contribuindo para a interpretação qualitativa dos agrupamentos.

5.2.2 Agrupamento Geral

Para a realização de um agrupamento eficaz considerando todo o conjunto de dados, foi necessário, primeiramente, definir o número ótimo de grupos, como mencionado na Seção 4.4.1. Como referência inicial, considerou-se o número total de categorias distintas do campo `Elemento da Despesa`, que totaliza 129. No entanto, observou-se que uma parte significativa desses elementos pertence a categorias raras, com poucos registros. Por isso, adotar $k = 129$ não se mostrou adequado, pois resultaria em agrupamentos excessivamente fragmentados.

Portanto, o valor $k = 129$ foi considerado um limite superior plausível, baseado na hipótese idealizada de que cada categoria do `Elemento da Despesa` poderia formar um agrupamento distinto.

Para investigar o número ótimo de grupos, inicialmente foram testados diferentes valores de k, utilizando o algoritmo *MiniBatchKMeans*. No entanto, os resultados obtidos apresentaram baixa qualidade de agrupamento e revelaram um padrão em que o *Silhouette Score* aumentava continuamente à medida que k crescia, sem indicar um ponto claro de estabilização. Esse comportamento pode ser observado na Figura 5.2, e sugere um possível sobreajuste aos dados por parte do algoritmo.

Devido aos resultados insatisfatórios obtidos com o *MiniBatchKMeans*, diferentes valores de k foram testados utilizando o próprio DEC. Os resultados estão ilustrados na Figura 5.3, na qual se observa a variação do *Silhouette Score* em função de diferentes valores de k, permitindo uma análise visual do comportamento dessa métrica.

Com base nos escores obtidos, observa-se que o valor ótimo de k é 56, apresentando um escore elevado (91.72).

Ainda que valores ligeiramente superiores possam resultar em agrupamentos mais granulares, como observamos o aumento do escore para $k = 100$, optou-se por uma quantidade menor de agrupamentos por razões de interoperabilidade e simplicidade na integração com sistemas posteriores.

Uma vez o k_{optimal} definido, o algoritmo do DEC pode ser executado. Podem ser definidos os seguintes hiperparâmetros:

- **optimal_k**: número ótimo de agrupamentos, conforme determinado por meio da análise do *Silhouette Score*;
- **epochs_dec**: número de épocas de treinamento do DEC, definido como 50;

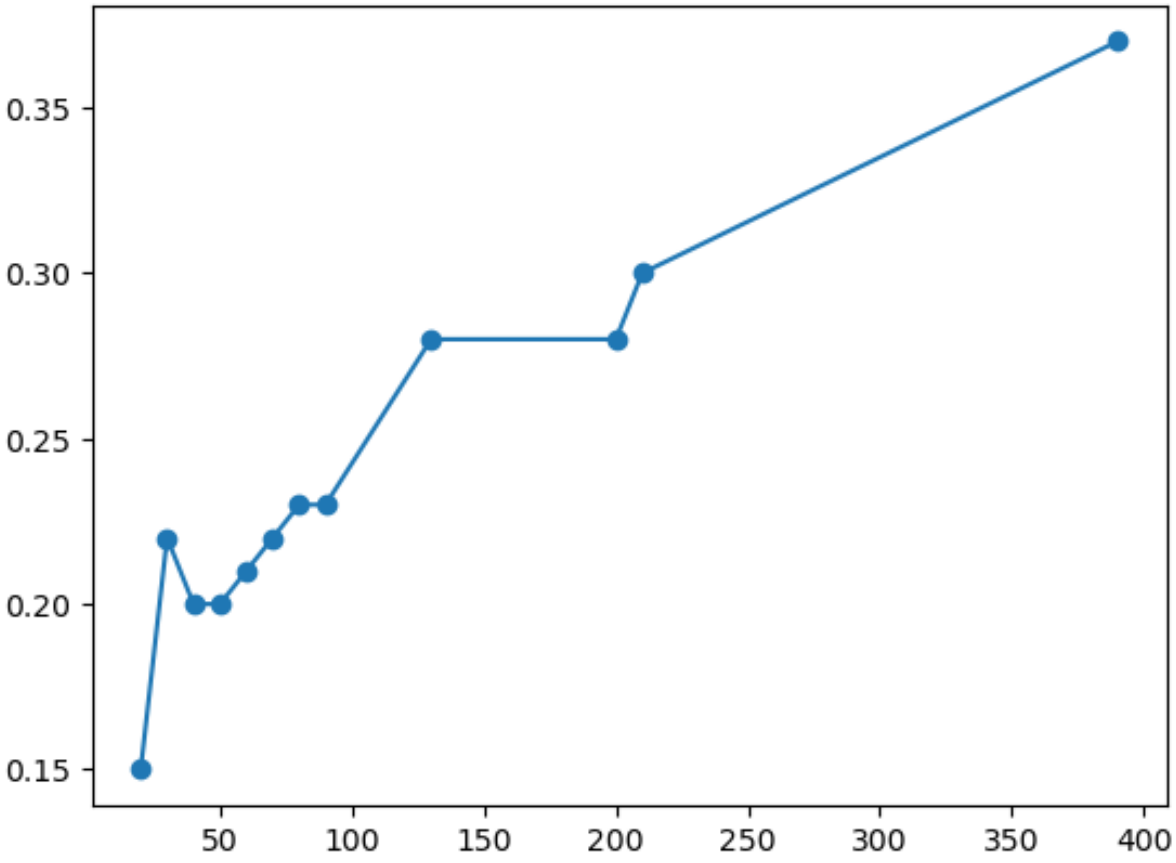


Figura 5.2: Variação do *Silhouette Score* em função dos valores de k para o conjunto completo de dados, utilizando o algoritmo *MiniBatchKMeans*.

- **otimizador**: o algoritmo de otimização utilizado é o SGD (*Stochastic Gradient Descent*);
- **taxa de aprendizado (lr)**: 0.01;
- **momentum**: 0.8;
- **weight decay**: 1×10^{-4}.

A partir da Tabela 5.1, observa-se os dez agrupamentos mais representativos no conjunto de dados, juntamente com suas respectivas frequências. Nota-se que o grupo 8 é o mais frequente, abrangendo aproximadamente 7.20% do total de itens de empenho. Ainda assim, esse percentual indica que não há um único agrupamento dominante no conjunto, o que sugere uma distribuição relativamente equilibrada entre os grupos.

Também é possível realizar uma análise do campo `Vlr_Empenhado` por agrupamento, conforme ilustrado na Figura 5.4. Ao considerar os quatro agrupamentos mais frequentes da base, observa-se que o grupo 7 apresenta uma média de valor empenhado significativamente baixa.

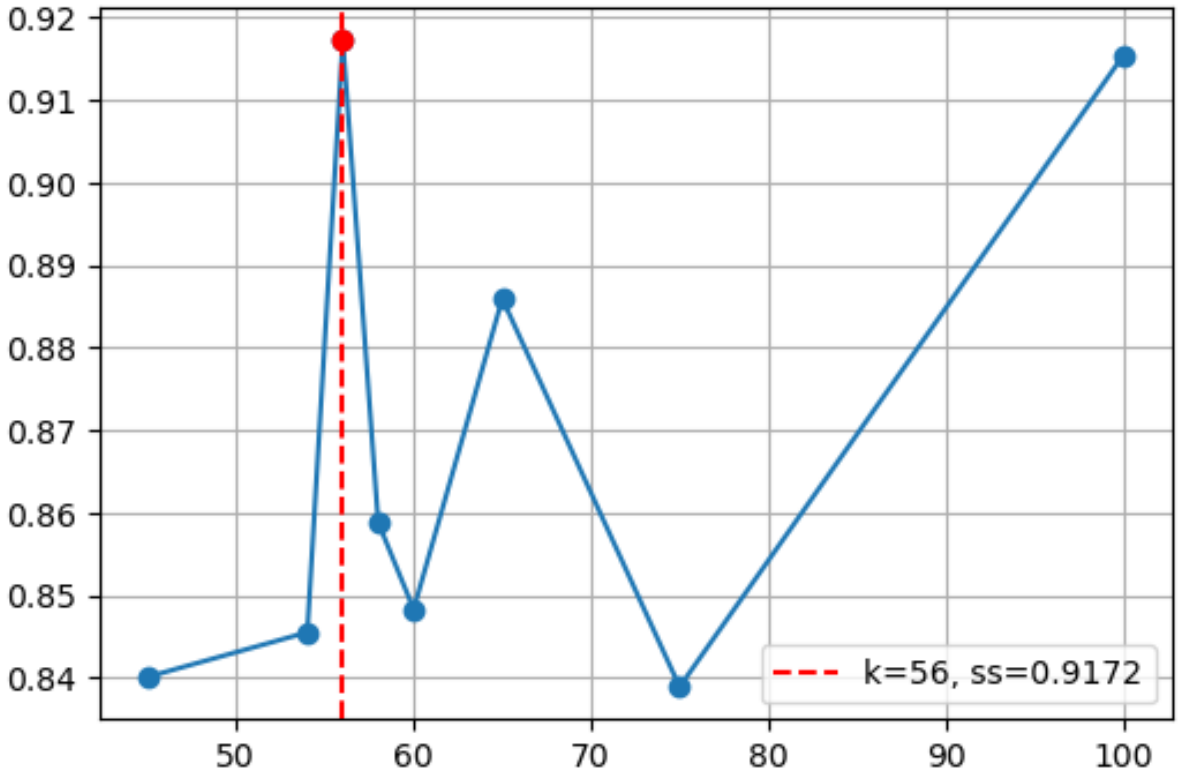


Figura 5.3: Silhouette Scores por valor k para o conjunto completo de dados.

No entanto, sua alta frequência no conjunto pode indicar que, apesar dos valores individuais reduzidos, o total empenhado por esse grupo é elevado.

5.2.3 Agrupamento Segmentado por Elemento da Despesa

Como mencionado na Seção 4.4.2, utilizou-se o algoritmo *MiniBatchKMeans* para realizar os agrupamentos e o *Silhouette Score* como métrica de avaliação. Neste experimento, antes da aplicação do algoritmo de agrupamento de dados, o conjunto de dados também foi transformado por meio do codificador (*encoder*) do *Autoencoder*, visando reduzir a dimensionalidade e fornecer representações mais compactas e informativas ao algoritmo de agrupamento de dados.

O intervalo de valores de k foi determinado segundo as seguintes regras:

- Se `len(dataset)` $= 1$, então $k = 1$.
- Se `len(dataset)` $= 2$, testar apenas $k = 2$.
- Se $2 <$ `len(dataset)` < 11, então $k \in \{2, 3\}$.
- Se $11 \leq$ `len(dataset)` < 40, então $k \in \{2, \ldots, 10\}$.

Agrupamento	Frequência	Percentual
8	106.982	7.20%
22	79.011	5.32%
7	72.445	4.87%
43	72.352	4.87%
50	69.253	4.66%
2	62.379	4.20%
10	61.965	4.17%
24	59.592	4.01%
0	58.278	3.92%
30	57.773	3.89%

Tabela 5.1: Tabela contendo os 10 agrupamentos mais frequentes no conjunto de dados, com suas respectivas frequências absolutas e percentuais.

- Se $40 \leq \texttt{len(dataset)} < 1000$, então $k \in \{2, \ldots, 16\}$.
- Se $1000 \leq \texttt{len(dataset)} < 2000$, então $k \in \{2, \ldots, 31\}$.
- Se $2000 \leq \texttt{len(dataset)} < 10000$, então $k \in \{2, \ldots, 51\}$.
- Se $\texttt{len(dataset)} \geq 10000$, então $k \in \{2, \ldots, 61\}$.

Além disso, conforme mencionado na Seção 4.4.2, foi estabelecido um limiar mínimo de 0.25 para o *Silhouette Score*. Quando o escore calculado para um agrupamento é inferior a esse valor, define-se automaticamente $k = 1$. Assim, nos casos em que o algoritmo não identifique mais de um grupo distinto ou em que a partição em dois grupos resulte em baixa qualidade, opta-se por considerar apenas um único agrupamento. Observa-se, na Figura 5.5 a distribuição dos valores k para os subconjuntos.

Uma vez os k_{optimal} definidos, o algoritmo do DEC foi executado com a bandeira –sort-by-elem como *True* para habilitar o treinamento de até 129 modelos diferentes, com seus respectivos valores k. Os seus hiperparâmetros, configurados no arquivo config.yaml, são os mesmos utilizados pelo agrupamento feita de maneira geral, exceto pelo optimal_k, onde aqui utilizamos o optimal_ks_by_elem.

Uma vez concluído o treinamento, os resultados do *Silhouette Score* para os subconjuntos em que $k > 1$ estão apresentados na Figura 5.6.

É possível observar que a maior concentração de modelos encontra-se na faixa de 0,3–0,4, com 15 ocorrências. Faixas superiores, como 0,6–0,7, 0,7–0,8 e até 0,9–1,0, apresentam menor frequência, o que indica que poucos modelos atingiram agrupamentos de alta qualidade. Esse resultado sugere que ainda há espaço para melhorias nos parâmetros do modelo ou na escolha

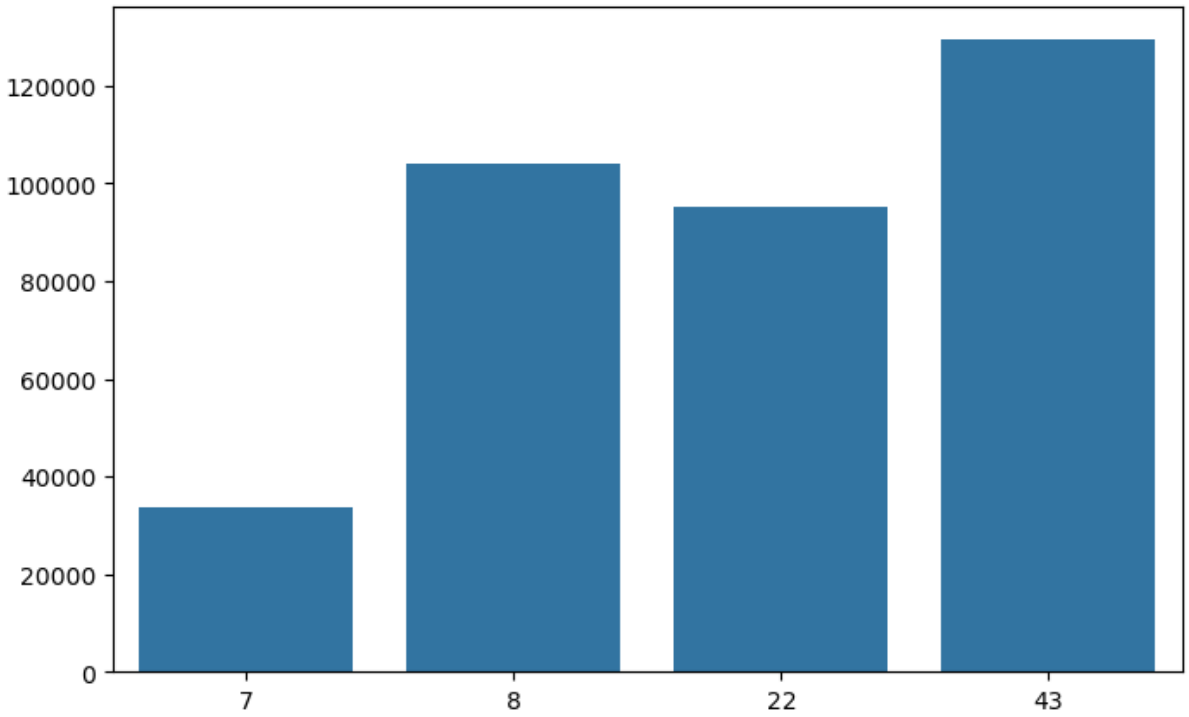


Figura 5.4: Gráfico com o valor empenhado médio dos itens de empenho pertencentes aos quatro agrupamentos mais frequentes no conjunto de dados.

do número de grupos k, considerando especialmente a dificuldade em definir o valor ótimo de k. Isso ocorre porque, ao aumentar k com o *MiniBatchKMeans*, o *Silhouette Score* tende a crescer indefinidamente, o que pode levar a uma superestimação da quantidade ideal de agrupamentos.

Com o agrupamento realizado por subconjunto, foi possível identificar um número maior de grupos em comparação com o agrupamento aplicado ao conjunto completo dos dados, totalizando 276 grupos distintos. A estrutura de identificação dos agrupamentos segue o padrão: índice do subconjunto seguido do identificador do agrupamento dentro desse subconjunto, separados por um ponto. Por exemplo, o agrupamento 1 do subconjunto 0 é representado como 0,1. É possível observar os 20 agrupamentos mais frequentes pela Figura 5.7

Pode-se, ainda, realizar uma análise exploratória com base no campo `Vlr_Empenhado`, observando as diferenças entre os agrupamentos dentro de um mesmo subconjunto. A Figura 5.8 ilustra essa variação ao apresentar a média do valor empenhado por agrupamento no subconjunto 40.

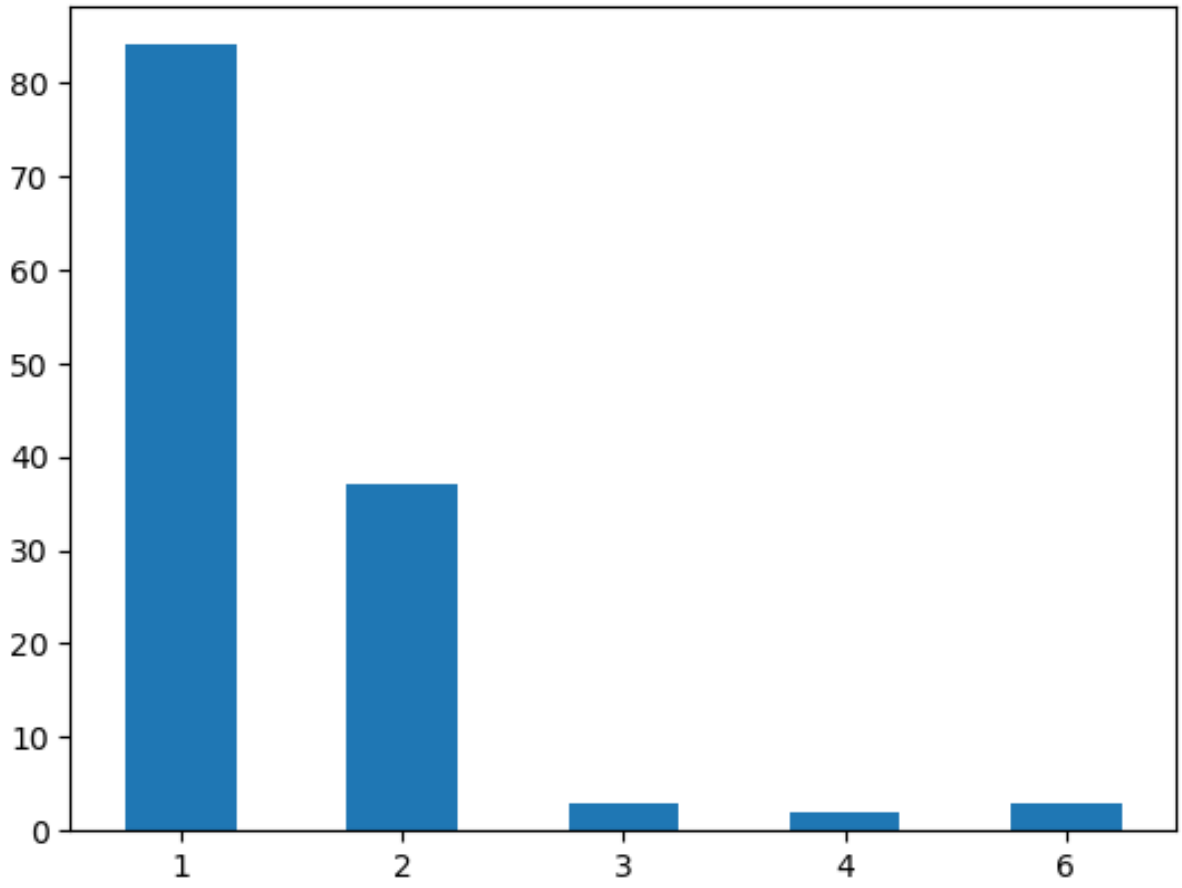


Figura 5.5: Gráfico representando as ocorrências do hiperparâmetro *k* escolhido para a análise de agrupamento de dados por subconjunto.

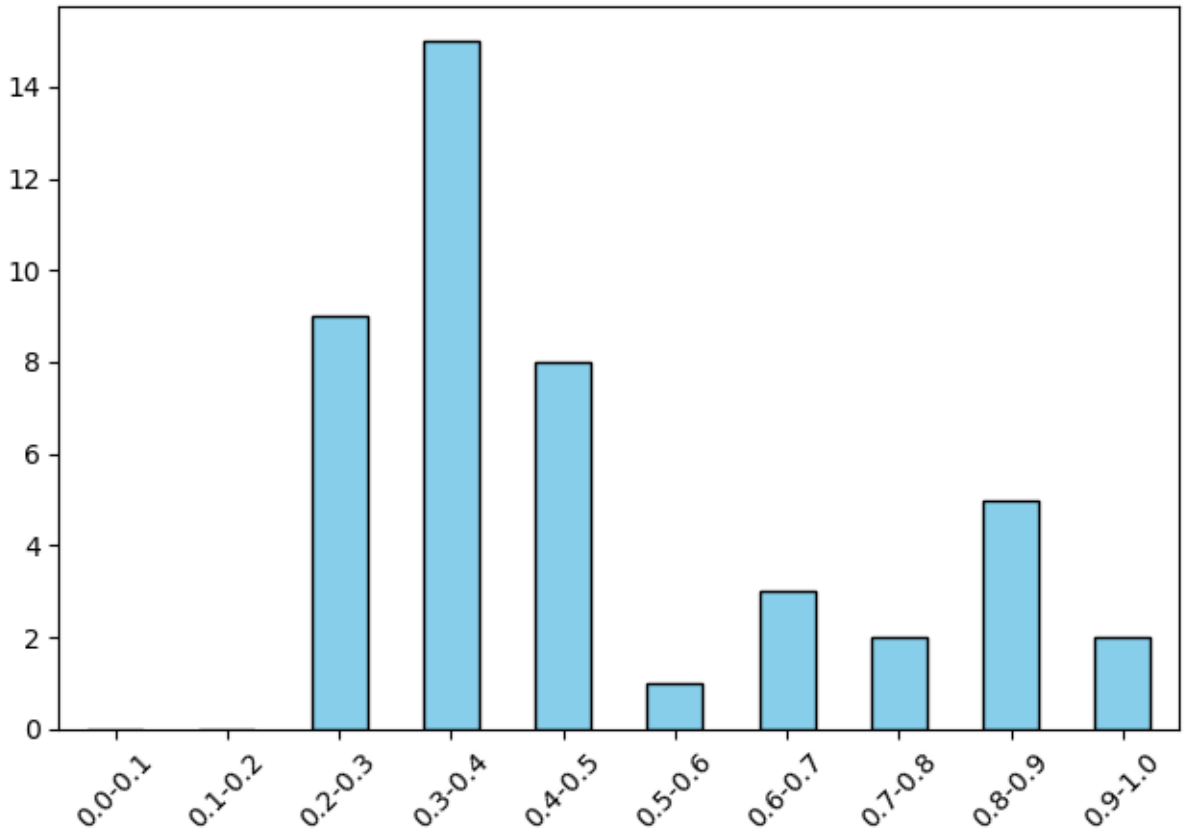


Figura 5.6: Gráfico representando a distribuição dos *Silhouette Scores* obtidos para cada subconjunto, considerando apenas os casos em que $k > 1$.

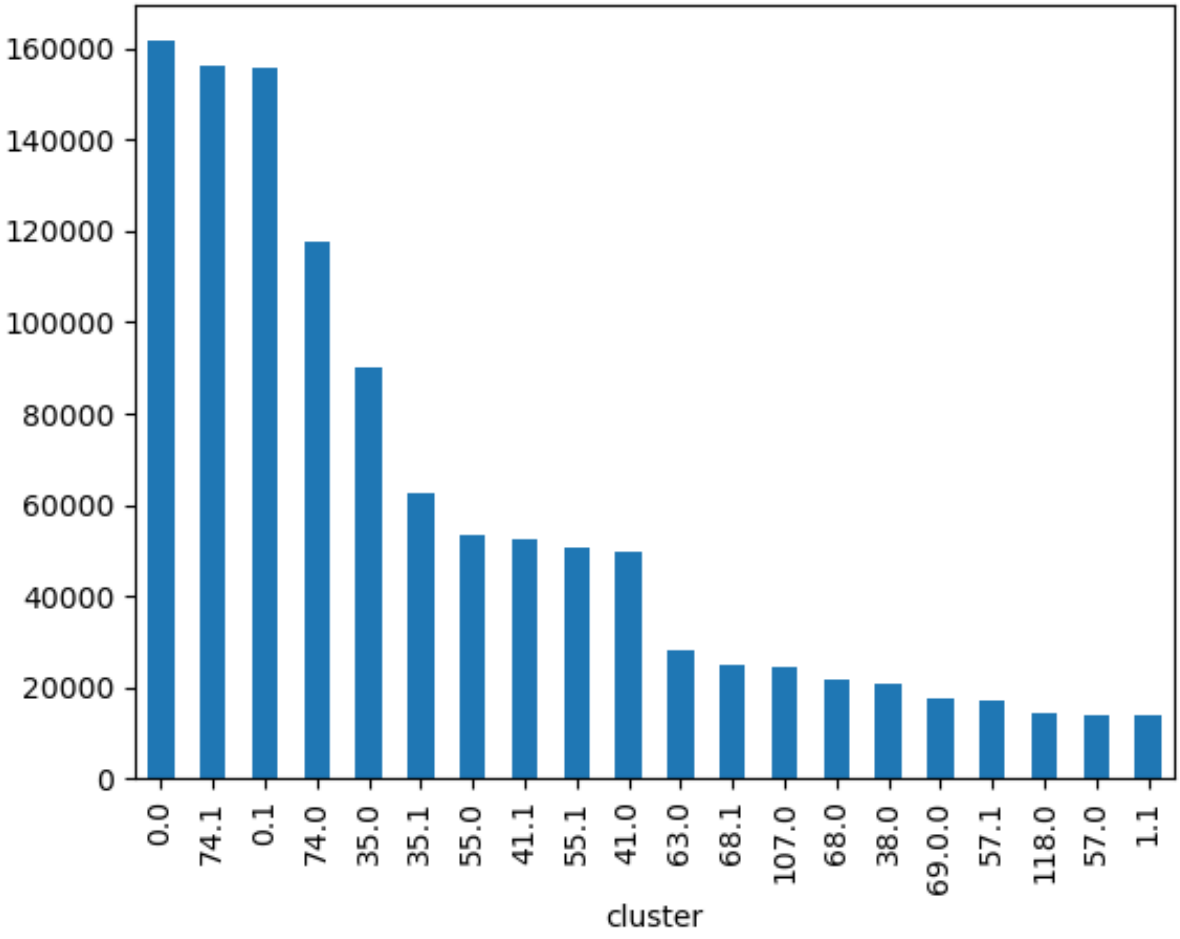


Figura 5.7: Gráfico representando a distribuição agrupamentos mais frequentes, apresentando somente o top 20.

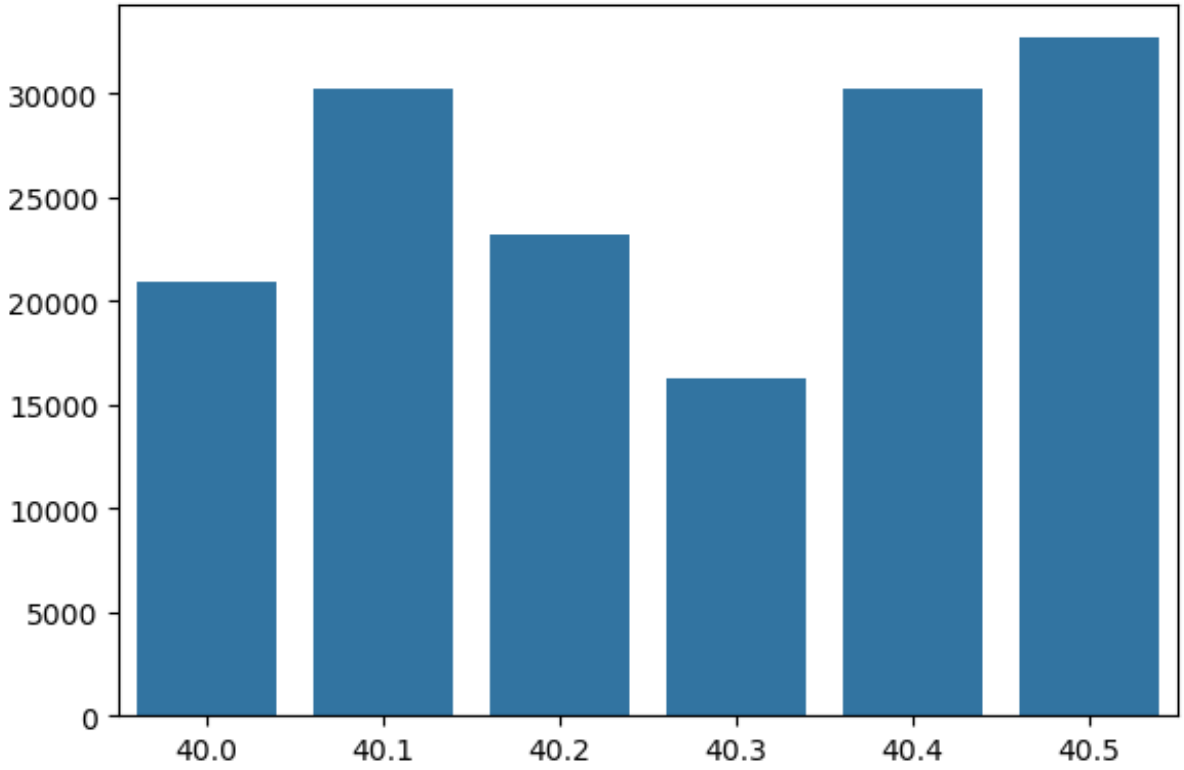


Figura 5.8: Gráfico representando a média do valor empenhado (`Vlr_Empenhado`) para cada grupo identificado no subconjunto de dados com índice 40. A figura evidencia como os diferentes grupos apresentam comportamentos distintos em relação aos valores financeiros empenhados, permitindo identificar padrões ou anomalias dentro do subconjunto analisado.

Capítulo 6

Considerações Finais

Neste capítulo, são apresentadas as considerações finais deste trabalho, com destaque para os principais resultados obtidos e as contribuições alcançadas ao longo da pesquisa.

O objetivo principal deste estudo foi desenvolver uma abordagem para o agrupamento do conjunto de dados de Itens de Empenho por meio da aplicação de técnicas de agrupamento de dados combinadas com *Autoencoders*, dentro do contexto da gestão financeira do TCE/RJ.

Para atingir esse objetivo, foram realizadas etapas de pré-processamento dos dados, definição de subconjuntos, treinamento de modelos de codificação e aplicação do modelo DEC. A métrica do *Silhouette Score* foi utilizada para avaliar a qualidade dos agrupamentos.

Os resultados indicaram que a abordagem de agrupamento foi capaz de gerar grupos com uma boa relação entre as distâncias intra e interclusters. Isso pôde ser verificado a partir do valor do *Silhouette Score*, que atingiu 0,9172 para 56 grupos. Como comparação, o agrupamento baseada em subconjuntos também resultou em uma segmentação refinada, composta por 276 grupos distintos. Ademais, a análise por faixas de *Silhouette Score* revelou que a maioria dos agrupamentos apresenta coerência moderada, indicando um bom ponto de partida, ainda que com espaço para melhorias.

Apesar dos resultados promissores, algumas limitações foram identificadas, como a presença de ruído nos dados, que impactou negativamente a qualidade do agrupamento. Outro desafio relevante foi a definição do número ótimo de grupos (k), especialmente em subconjuntos mais ruidosos, nos quais a segmentação adequada se mostrou mais difícil.

Este trabalho contribui ao demonstrar que a segmentação por subconjuntos, aliada ao uso de *Autoencoders* e técnicas de agrupamento de dados, constitui uma estratégia eficaz para lidar com conjuntos de dados de alta dimensionalidade.

6.0.1 Trabalhos Futuros

Como continuidade deste estudo, sugere-se investigar combinações de técnicas de agrupamentos integradas ao DEC, como o agrupamento espectral. Este algoritmo pode capturar

padrões que o *KMeans*, utilizado na inicialização dos centroides no DEC, pode não identificar de forma eficiente.

Além disso, recomenda-se a exploração de uma faixa mais ampla de hiperparâmetros, incluindo a taxa de aprendizado, o *momentum* e o número de épocas, visando otimizar o desempenho do modelo.

Por fim, outra direção promissora consiste na experimentação de abordagens para a determinação do número ótimo de grupos (k), visando aumentar a adaptabilidade da segmentação. Neste trabalho, foi utilizado o algoritmo *MiniBatchKMeans* em conjunto com a métrica do escore de Silhouette. No entanto, embora o *MiniBatchKMeans* tenha se mostrado eficiente em termos de tempo de execução ao lidar com grandes volumes de dados, os resultados obtidos não foram ideais em termos de qualidade de agrupamento. Observou-se que, à medida que o valor de k aumentava, o *Silhouette Score* também crescia, sugerindo um possível sobreajuste.

Referências Bibliográficas

Asadi, R. and Regan, A. (2019). Spatio-temporal clustering of traffic data with deep embedded clustering. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility*, pages 45–52.

Google Scholar (2024). Google Scholar. Acessado em: 10 fev. 2024.

IEEE Xplore (2024). IEEE Xplore Digital Library. Acessado em: 10 fev. 2024.

Joksimovic, J., Zoran, L., and Ženko, B. (2023). Detecting corruption in slovenian public spending from temporal data. *Proceedings of the ITIS 2023 Conference (Faculty of Information Studies, Novo mesto / Jožef Stefan Institute)*.

Lee, Y., Park, C., and Kang, S. (2022). Deep embedded clustering framework for mixed data. *IEEE Access*, 11:33–40.

Lukiyanov, V. (2019). pt-dec: Deep embedded clustering in pytorch. https://github.com/vlukiyanov/pt-dec.

Min, W., Liang, W., Yin, H., Wang, Z., Li, M., and Lal, A. (2021). Explainable deep behavioral sequence clustering for transaction fraud detection. *arXiv preprint arXiv:2101.04285*.

Paraizo, V. (2024). Dec: Adaptação aplicado à notas de empenho.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Xie, J., Girshick, R., and Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.

Apêndice A

Descrição da Aplicação Web

Neste apêndice, apresentamos a aplicação web desenvolvida concomitantemente a esta pesquisa. O objetivo principal da plataforma é fornecer um ambiente interativo que facilite a consulta e análise de itens de empenho presentes na base de dados, possibilitando ao usuário o acesso a registros similares com base em critérios semânticos e estruturais.

A aplicação foi construída utilizando a biblioteca *Streamlit*, que oferece uma abordagem simplificada para o desenvolvimento de interfaces web em Python, com foco em projetos de ciência de dados. Além disso, a aplicação integra outras ferramentas utilizadas nesta pesquisa, como a biblioteca *Langchain*, permitindo a execução de consultas semânticas e a recuperação de informações relevantes de forma eficiente e intuitiva.

Os itens de empenho retornados em resposta à consulta exibem os campos mais relevantes para esta pesquisa: `Historico`, `Credor`, `Unidade`, `Elemento da Despesa`, além de metadados adicionais, como o *Valor Empenhado* e o *cluster* ao qual o item pertence, uma vez realizada a etapa de clusterização com o modelo *DEC*.

A interface da aplicação permite a visualização direta dessas informações, conforme ilustrado na Figura A.1.

Após a realização da consulta, o sistema retorna, logo abaixo dos campos de pesquisa, os itens de empenho que apresentam alta similaridade com a *string* informada. São exibidos 10 itens por página, sem limite máximo de páginas. A quantidade de páginas depende apenas do total de itens de empenho retornados.

A Figura A.2 ilustra a visualização desses resultados na interface da aplicação.

A.0.1 Vector Store

Para viabilizar a busca semântica na aplicação, foi criada uma *vector store* utilizando a ferramenta `Chroma`, integrante do ecossistema do *framework* Langchain. Essa estrutura é responsável por armazenar, de forma otimizada, os itens de empenho da base de dados juntamente com seus respectivos *embeddings*, previamente extraídos a partir do modelo *sentence-transformers*.

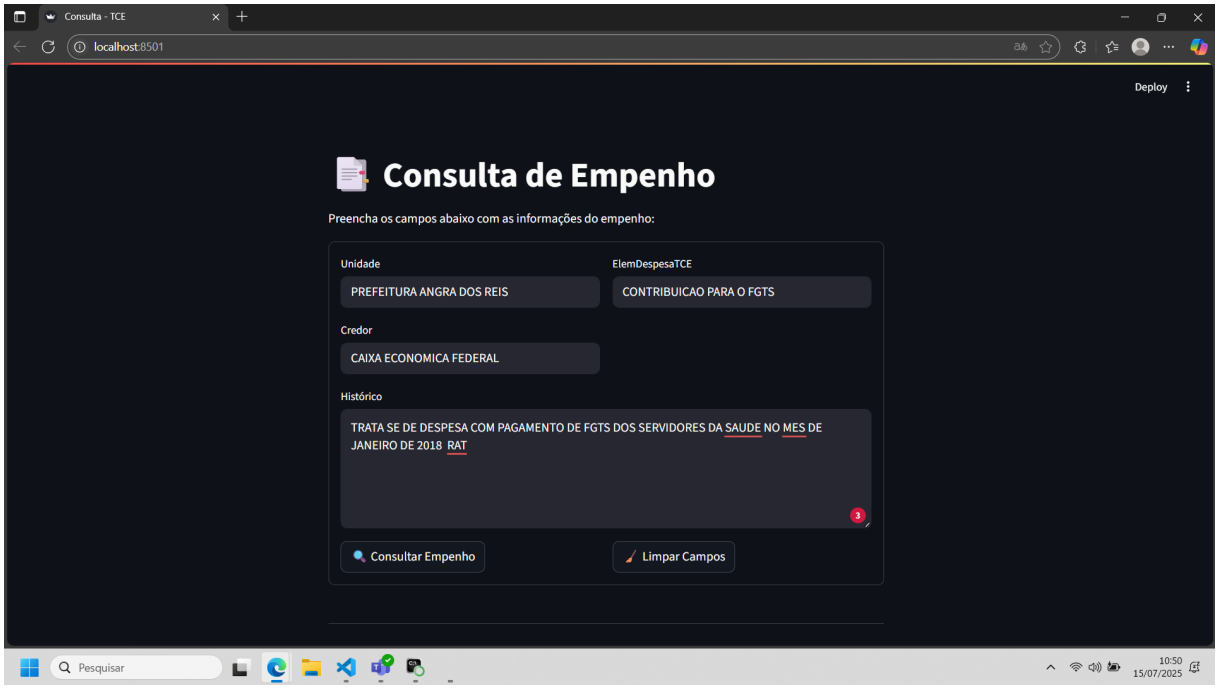


Figura A.1: Interface do sistema Web de consultas à base de itens de empenho por meio de um vector store, permitindo a realização de buscas na base de dados.

A utilização da *vector store* permite a realização de consultas vetoriais com alta eficiência, o que é fundamental para o funcionamento do mecanismo de similaridade textual implementado. Assim, ao submeter uma consulta textual à aplicação, o sistema é capaz de recuperar, em tempo real, os itens de empenho mais semanticamente próximos à entrada fornecida pelo usuário.

Durante a execução da aplicação, ao receber uma consulta textual por meio da interface, o texto inserido é transformado em um vetor de *embedding*, utilizando o mesmo modelo previamente aplicado à base original. Em seguida, calcula-se a similaridade entre esse vetor de entrada e todos os vetores presentes na *vector store*, por meio de uma métrica de distância vetorial.

Os itens de empenho que apresentarem distância inferior a um limiar pré-definido, fixado como hiperparâmetro com valor de 1.5, são retornados como resultados relevantes à consulta. Esse limiar foi escolhido de forma empírica, com o intuito de balancear precisão e abrangência na recuperação dos itens semelhantes.

A.0.2 Reprodutibilidade do Aplicativo

Para a execução da aplicação web, é necessário que a *vector store* já tenha sido gerada previamente, bem como que a clusterização utilizando o modelo *DEC* tenha sido realizada. A gera-

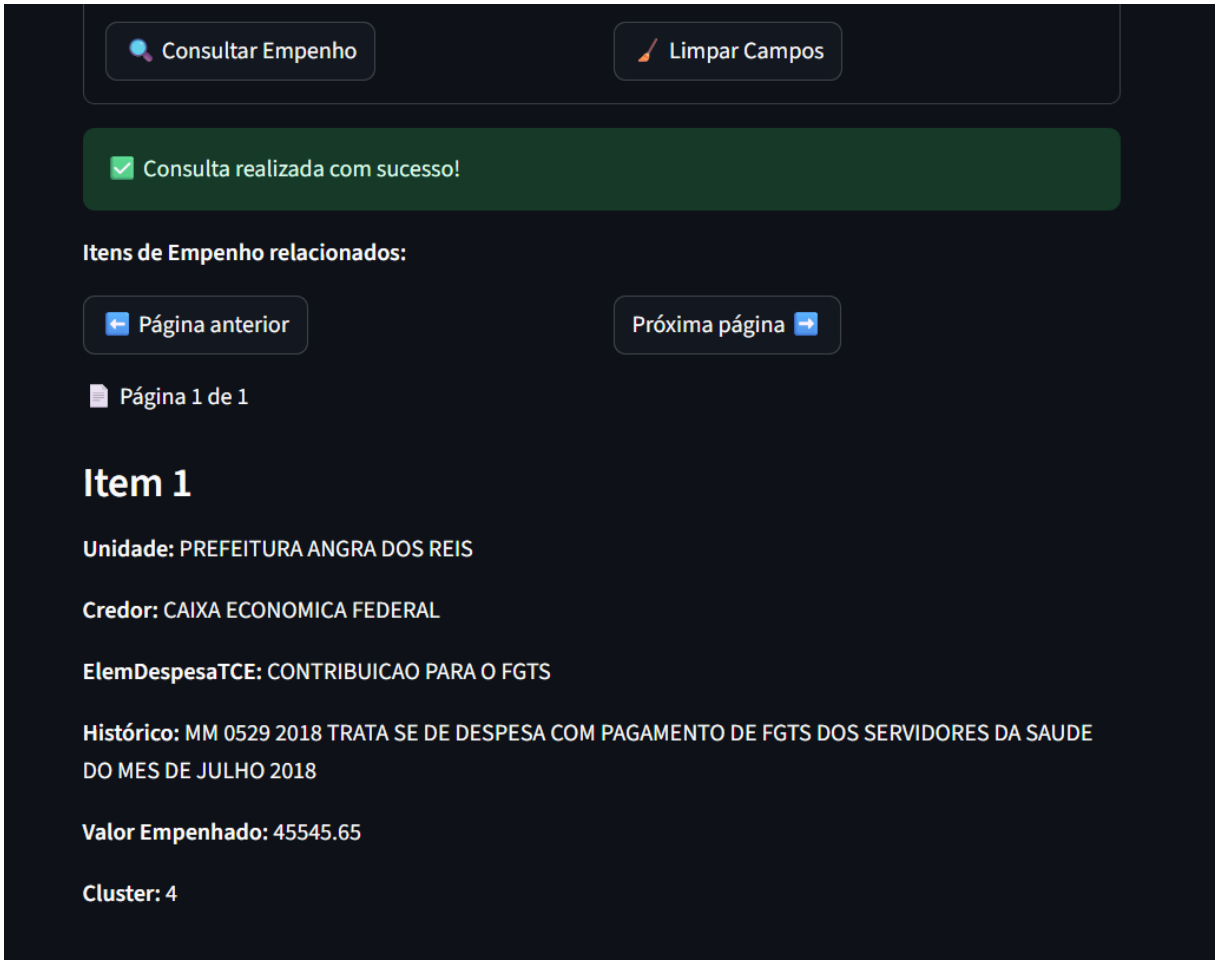


Figura A.2: Resultados da consulta exibidos abaixo dos campos de entrada, com itens de empenho que possuem alto escore de similaridade. A informação de 'Cluster' também é apresentada.

ção da *vector store* pode ser feita por meio da execução do script `chroma_vector_store.py`, calculando também os *embeddings* da base completa, caso estes ainda não existam.

Os *embeddings* são construídos a partir da concatenação dos campos `Historico`, `Credor`, `Elemento da Despesa` e `Unidade`, de modo a capturar as representações semânticas de cada item de empenho de forma mais abrangente. Essa estratégia permite que, durante a busca, a similaridade seja avaliada com base em múltiplos atributos relevantes, e não apenas no texto do campo `Historico`.