Universidade de Brasília
Departamento de Estatística

Agrupador de processos de controle concentrado

Bruno Gondim Toledo

Projeto apresentado para o Departamento de Estatística da Universidade de Brasília como parte dos requisitos necessários para obtenção do grau de Bacharel em Estatística.

Brasília
2024

Bruno Gondim Toledo

Agrupador de processos de controle concentrado

Orientador(a): Profª Drª Thais Carvalho Valadares Rodrigues.

Projeto apresentado para o Departamento de Estatística da Universidade de Brasília como parte dos requisitos necessários para obtenção do grau de Bacharel em Estatística.

Brasília
2024

Sumário

1 Introdução

Modelagem estatística tornou-se essencial em diversos setores, inclusive no poder Judiciário, visto que tem sido do interesse dos próprios tribunais saber indicadores e estatísticas relacionadas ao próprio tribunal, além de agilizar processos e direcionar os recursos humanos para tarefas menos mecânicas. A modelagem surge neste contexto para agregar valor às decisões estratégicas deste poder. Com o advento de novas técnicas computacionais, é possível agrupar e classificar textos do contexto jurídico, conforme descrito por Freitas et al. (2024).

O propósito deste trabalho é formular um agregador de processos de controle concentrado do Supremo Tribunal Federal (STF). Compõem o controle concentrado as seguintes classes processuais: Ações Diretas de Inconstitucionalidade (ADIs), as Ações Declaratórias de Constitucionalidade (ADCs), as Ações Diretas de Inconstitucionalidade por Omissão (ADOs) e as Arguições de Descumprimento de Preceito Fundamental (ADPFs).

Para a construção deste agregador, pretende-se vetorizar as petições dos processos de controle concentrado com a técnica de vetorização (do ingês, *embedding*) utilizada em *"Text clustering applied to data augmentation in legal contexts"* (Freitas et al, 2024), com posterior aplicação de métodos de agrupamento destes vetores. Uma etapa anterior à vetorização é o pré-processamento do texto obtido, como a remoção de *stopwords*, pontuações, termos legais, números, espaços em branco, conforme discutido em Freitas et al. (2023).

Por conta da natureza cíclica dos processos que compõem o acervo do STF, não é esperado a formulação de classes rígidas para este agrupamento, sendo somente necessário encontrar similaridades e dissimilaridades entre os processos em atual tramitação à fim de agrupá-los de forma a facilitar o fluxo de trabalho do tribunal. Por exemplo, ao agrupar processos semelhantes, estes podem ser analisados conjuntamente, facilitando o fluxo de trabalho do STF. Além disso, o agrupador pode facilitar uma priorização de pauta de determinados processos. Portanto, a proposta deste trabalho é estudar técnicas de agrupar estes processos de forma funcional e contínua, fornecendo assim um produto para o tribunal, a fim de ser utilizado pelos analistas que hoje buscam manualmente esses agrupamentos.

2 Objetivos

2.1 Objetivo Geral

Elaborar um agrupador de processos de controle concentrado para os processos em tramitação no Supremo Tribunal Federal. O agrupador fornecerá subsídios aos responsáveis pelo encaminhamento dos processos que chegam ao STF, visando reduzir o trabalho mecânico dos servidores do tribunal que atualmente necessitam analisar e encaminhar processos semelhantes aos setores responsáveis.

2.2 Objetivos Específicos

- Processar os dados utilizando técnicas de Processamento de Linguagem Natural (PLN), transformando as petições iniciais dos processos em PDF (dados de entrada) para vetores numéricos;
- Comparar técnicas;
- Aplicar e comparar tecnologias de agrupamento dos dados;
- Avaliar os agrupamentos fornecidos pela aplicação com as reais demandas do tribunal.
- Estudar técnicas de processamento de linguagem natural, análise multivariada e visualização de dados.

O processamento inicial dos dados (transformação dos textos em vetores numéricos) será realizado utilizando o fluxo do tribunal, ou seja, os modelos já existentes neste para ocerização, processamento de texto e vetorização, a fim de tornar a aplicação comparável com as demais existentes no tribunal.

3 Metodologia

A metodologia aplicada para processamento e conversão de dados textuais em vetores numéricos (vetorização) para posterior agrupamento será a mesma aplicada por Freitas et al. (2024), consistindo nas etapas de limpeza de dados, vetorização e organização dos dados. Após o processamento inicial, os dados serão agrupados segundo alguma medida de similaridade. Métodos de agrupamento hierárquico e não hierárquico serão explorados.

3.1 Pré-processamento

Rotinas de pré-processamento devem ser realizadas para viabilizar qualquer aplicação de PLN. No contexto deste trabalho, as petições iniciais disponibilizada pelo Portal do STF encontram-se em formato PDF. Dentre estas, encontram-se petições oriundas de documentos impressos e posteriormente digitalizados e salvos neste formato, bem como documentos nativamente digitados em alguma plataforma e posteriormente exportado em formato PDF. Em ambos os casos, os documentos contém suas respectivas imagens, figuras, timbres e demais elementos visuais que não compõem o paradigma desta aplicação, necessitando portanto serem filtrados. Funções implementadas em linguagens de programação (como *pdf_text* do pacote *pdftools* da linguagem de programação **R**) em geral já realizam a leitura do conteúdo textual do documento sem levar em consideração os elementos gráficos deste, para o caso de documentos nativamente digitados.

No caso dos documentos digitalizados, é necessária uma etapa posterior de ocerização (do acrônimo OCR - *Optical Character Recognition*). Esta técnica, também chamada de Reconhecimento ótico de caracteres, é utilizada para transcrever o conteúdo textual do documento — que na prática é uma imagem neste caso — para caracteres numéricos e alfanuméricos. Diversas implementações de ocerizadores estão disponíveis na maior parte das linguagens de programação mais populares, muitas vezes na forma de pacotes. Para o escopo deste trabalho, em que o recorte amostral é do ano de 2016 até 2024, grande parte das petições iniciais encontravam-se disponíveis em PDF oriundos de documentos nativamente digitais, portanto não necessitando de uma etapa de ocerização. No caso das petições que necessitavam de ocerização, estas foram fornecidas diretamente pela STI do STF já ocerizadas pela implementação própria do tribunal desta tecnologia.

Tendo sido os textos importados para o ambiente de uma linguagem de programação, por vezes são necessários ajustes ortográficos das palavras importadas. Como nas tipografias mais usuais existem caracteres similares entre si (como a letra i maiúscula e a letra l minúscula, ou a letra o e o número 0), ou ainda palavras que sofrem translineação

(isto é, quando a palavra é dividida por um hífen quando não cabe em apenas uma linha), por vezes os algoritmos utilizados nas etapas anteriores de pré processamento retornam palavras incompletas ou sem sentido por conta destes fenômenos. Diversas técnicas podem ser utilizadas para corrigir estas distorções, como o algoritmo de Metropolis-Hastings aplicado à descriptografia, como utilizado por Veroutis e Fajardo (2021). Para os dados desta aplicação, foi realizada uma rotina para correção destas distorções nos dados pela STI do STF. Detalhes do funcionamento desta tecnologia não podem ser descritas por motivos de segurança de informação.

Sob a intenção de aplicar técnicas de vetorização, o estado da arte de PLN sugere que sejam removidas palavras que não contribuem informativamente com a semântica de um texto, conforme descrito por Sarica e Luo (2021). Em geral, constituem as ditas palavras de parada (ou ainda palavras vazias, no inglês *stopwords*), conectivos e conjunções. Não existe consenso de quais palavras podem ser consideradas palavras vazias, variando também em função do contexto em que se pretende manipular estes textos. Portanto, esta lista pode variar de centenas de palavras ou até nenhuma palavra, a depender da aplicação. Para esta aplicação, utilizou-se do modelo pré-treinado do tribunal para vetorização deste trabalho, e não se sabe ao certo quais as palavras foram consideradas palavras de parada e portanto filtradas nos seus dados de treinamento, pois detalhes dos parâmetros deste modelo não podem ser revelados por questões de segurança de informação. Até o momento, para a vetorização dos dados que serão utilizados nesta aplicação, não foi aplicada qualquer filtragem de palavras vazias, vetorizando assim a totalidade do corpo textual, ainda que possivelmente possa ter sido aplicada uma rotina de remoção de palavras de parada aos dados de treino do modelo, possibilitando assim que esta aplicação seja utilizada como paradigma para um posterior avanço no estado da arte.

3.2 Vetorização

Vetorização, ou *Embedding*, é uma técnica geralmente aplicada na área de processamento de linguagem natural. Em poucas palavras, é um procedimento de transformar texto em um vetor numérico.

A ideia por trás da vetorização é transformar os dados originais, texto, em um vetor numérico que mapeie e preserve a carga semântica do texto original. Sem embargos, é possível rapidamente identificar a utilidade desta técnica — visto que, para utilização de técnicas e algoritmos estatísticos, em geral buscamos dados numéricos, e não textuais — quando intencionamos aplicar alguma técnica para modelagem destes dados. Diversas técnicas para este procedimento vem sendo estudadas, implementadas e utilizadas, como por exemplo o método *Term Frequency — Inverse Document Frequency* (TF-IDF) (Salton e Buckley, 1988, apud Freitas et al, 2024), *Word2vec* (Mikolov et al, 2013, apud Freitas

et al, 2024) e *doc2vec* (Le e Mikolov, 2014, apud Freitas et al, 2024). Este último busca a obtenção do vetor numérico representando o texto original via treinamento de uma rede neural. Nessa rede, os parâmetros do modelo (ou pesos) são atualizados buscando prever a próxima palavra dado o seu contexto (palavras ao redor) e dado um vetor de palavras que formam coletivamente o texto, conforme descrito em Freitas et al. (2024). Mais detalhes do funcionamento destes métodos podem ser encontrados em Freitas et al. (2024).

A utilização do contexto para a construção da vetorização pode aumentar a precisão desta. "O texto [...] é considerado parte integrante do contexto, e deve ser observado em relação às demais partes consideradas relevantes na declaração do contexto."(Firth, 1962, p.7, tradução própria[1]). Podemos argumentar que, ao considerar a vizinhança das palavras, conseguimos extrair significados que tragam sentido a utilização destas, inclusive aplicado ao contexto jurídico, evitando assim a homonímia. Por exemplo, um suposto processo com origem no estado do Rio de Janeiro que cite o nome do estado em sua petição inicial — suposição esta não descolada da realidade dos fatos — poderia confundir alguma técnica de vetorização que não considerasse vizinhanças para entendimento semântico do texto, visto que neste caso a palavra Rio não se refere ao corpo de água, tampouco a palavra Janeiro se refere a um mês do calendário.

Para esta aplicação, utilizaremos o método *doc2vec*, em conformidade com a rotina de vetorização de textos do Supremo Tribunal Federal. Com isso, além de garantir conformidade com demais tecnologias do tribunal que também utilizam alguma rotina de PLN, certificamos que o contexto semântico gerado pela vetorização esteja em conformidade com o vocabulário jurídico que se espera nos dados, visto que utilizaremos um modelo *doc2vec* pré-treinado da STI do tribunal. A utilização de um modelo pré treinado com dados do tribunal assegura o significado léxico dos vetores formados, visto que um modelo treinado com dados do escopo jurídico assegura uma menor quantidade de ruído quanto às expressões encontradas nos dados. Por exemplo, este modelo quase certamente não irá confundir a palavra Juiz com um contexto futebolístico de um árbitro de futebol, entendendo portanto que se refere a um magistrado no contexto da frase, evitando assim a necessidade de treinamento do modelo com quantidades massivas de dados como acaba sendo necessário em modelos de linguagem de grande escala para usos gerais, desta forma tornando viável a utilização desta aplicação em computadores pessoais com capacidades computacionais limitadas em um tempo significativamente baixo.

O modelo de vetorização foi implementado pela STI utilizando o pacote *gensim* na linguagem *python*. Para a rotina de vetorização desta aplicação, foi testado utilizar como dimensão dos vetores de características produzidos pelo modelo sob os dados inseridos o

[1]No original: "The text in the focus of attention on renewal of connection with an instance, is regarded as an integral part of the context, and is observed in relation to the other parts regarded as relevant in the statement of the context".

tamanho 300, padrão sugerido pela própria função de vetorização do pacote *gensim* ao inserir os dados, visto que este é um parâmetro opcional da função. Os resultados para esta dimensionalidade foram satisfatórios, e por isso decidiu-se manter o padrão para esta aplicação, possibilitando assim melhorias no estado da arte numa futura aplicação.

Preocupações éticas ao aplicar técnicas de vetorização devem ser sempre observadas. Estereótipos e fatores de gênero nos dados podem ser criados, mantidos ou até amplificados como observado por Bolukbasi et al. (2016). Aplicações ingênuas desta técnica podem servir para perpetuação de vieses sociais deletérios sob a justificativa de serem produzidos pela máquina, como o Racismo algorítmico descrito por Silva (2022). Esta aplicação propõe-se tão somente a auxiliar o agrupamento de processos em sua fase inicial de tramitação, sem qualquer interesse de substituir uma posterior análise humana rigorosa de seu escopo — e muito menos de substituir um magistrado quanto ao mérito do conteúdo do processo — evitando assim que esta aplicação seja uma responsável oculta na perpetuação destes vieses.

3.3 Medidas de similaridade e dissimilaridade

Tendo sido aplicadas as rotinas anteriores para obtenção de vetores numéricos criados a partir dos textos originais das petições iniciais dos processos, estudaremos a utilização de medidas de similaridade e dissimilaridade para comparação desses processos por meio das distâncias entre seus vetores, que serão em seguida definidas. Acredita-se que estas distâncias servirão para medir com fidedignidade as semelhanças e dissemelhanças do conteúdo textual das petições iniciais dos processos de controle concentrado, tornando assim possível o reconhecimento de determinados padrões no acervo.

Para agrupar os processos, será necessário estudar medidas de similaridade e dissimilaridade. Artes e Barroso (2023) argumentam que, para medidas de similaridade, quanto maior o valor, maior será a semelhança entre objetos, enquanto que para medidas de dissimilaridade, quanto maior o valor, mais diferente serão os objetos. Dado que estarei trabalhando com vetores numéricos fornecidos pelo processamento inicial dos dados (3.2), um dos caminhos possíveis a seguir é de pensar em distâncias estatísticas.

Uma medida de distância estatística pode ser definida simplesmente como a distância entre um ponto arbitrário P e um segundo ponto arbitrário Q, em qualquer espaço n-dimensional. Conforme definido por Johnson e Wichern (2007), qualquer medida de distância $d(P, Q)$ entre os pontos P e Q é válida se atender as seguintes propriedades, sendo R um terceiro ponto intermediário qualquer:

- $d(P,Q) = d(Q,P)$;
- $d(P,Q) > 0 \; se \; P \neq Q$;
- $d(P,Q) = 0 \; se \; P = Q$;
- $d(P,Q) \leq d(P,R) + d(R,Q)$ (Desigualdade triangular).

A última propriedade será evocada quando definirmos a distância do cosseno (seção 3.3.4). A seguir serão apresentadas algumas medidas de distância que poderão ser exploradas, sendo elas as distâncias euclidiana, de quarteirão, de Mahalanobis e de cossenos.

3.3.1 Distância Euclidiana

A distância Euclidiana é uma consequência direta do Teorema de Pitágoras. Segundo Johnson e Wichern (2007), considerando os pontos arbitrários P com coordenadas $P = (x_1, x_2, ..., x_p)$, e Q com coordenadas $Q = (y_1, y_2, ..., y_p)$, de uma hiperesfera p-dimensional, a distância reta entre eles, ou distância Euclidiana, é dada por:

$$d(P,Q) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_p - y_p)^2}. \tag{3.3.1}$$

A Figura 1 ilustra geometricamente a distância entre dois pontos **P** e **Q** no $\mathbb{R}^2$.

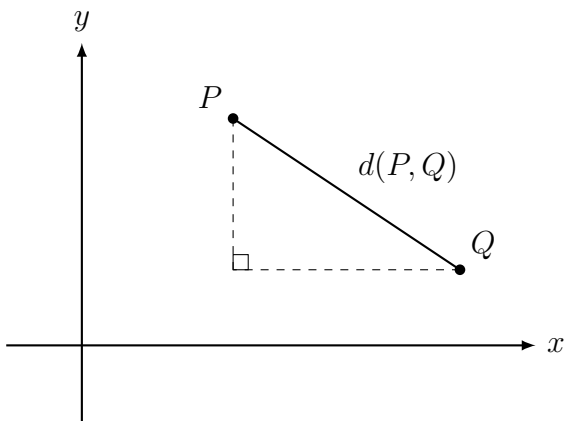

Figura 1: Distância Euclidiana entre os pontos **P** e **Q**

Note que esta definição para P, Q pode ser expandida para a situação em que estes são vetores X_i, X_k, i, k amostras com p dimensões. Utilizando a notação de Artes e Barroso (2023):

$$d_{ik} = \sqrt{(x_i - x_k)^T (x_i - x_k)} = \sqrt{\sum_{j=1}^{p} (X_{ij} - X_{kj})^2}. \tag{3.3.2}$$

3.3.2 Distância de quarteirão

Seja $x_i = (X_{i1}, ..., X_{ip})^T$ vetor de observações da i-ésima amostra, $i = 1, ..., n$, no qual X_{ij} é o valor assumido pela variável X_j da amostra i. Artes e Barroso (2023) definem a distância de quarteirão (ou *Manhattan*) entre a i-ésima amostra e a k-ésima amostra como

$$d_{ik} = \sum_{j=1}^{p} |X_i - X_k|. \tag{3.3.3}$$

A Figura 2 ilustra geometricamente a distância de quarteirão entre dois pontos $\mathbf{X_i}$ e $\mathbf{X_k}$ no $\mathbb{R}^2$.

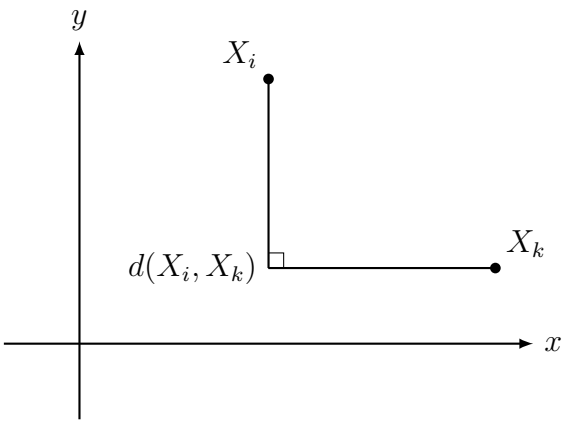

Figura 2: Distância de quarteirão entre os pontos $\mathbf{X}_i$ e $\mathbf{X}_k$.

3.3.3 Distância de Mahalanobis

Seja $\mathbf{x} = (X_1, ..., X_p)^T$ vetor aleatório com $\mathbb{E}(\mathbf{x}) = \mu = (\mu_1, ..., \mu_p)^T$, e $Cov(\mathbf{x}) = \mathbf{\Sigma}$. Artes e Barroso (2023) definem a distância de Mahalanobis ao quadrado entre $\mathbf{x}$ e μ como:

$$D^2_M(\mathbf{x}, \mu) = (\mathbf{x} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu). \tag{3.3.4}$$

Seja $i = i, ..., n$ unidades amostrais de dimensão p. Logo, o vetor $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ define a amostra. Seja $\bar{\mathbf{x}}$ vetor de média desta amostra, e $\mathbf{S}$ matriz de covariância amostral. Logo, a distância de um ponto i em relação ao centróide da amostra será

$$D^2_M(\mathbf{x_i}, \bar{\mathbf{x}}) = (\mathbf{x_i} - \bar{\mathbf{x}})^T \mathbf{S}^{-1} (\mathbf{x_i} - \bar{\mathbf{x}}). \tag{3.3.5}$$

Pela definição trazida na Equação (3.3.4), necessitaríamos saber os valores μ e $\mathbf{\Sigma}$. Artes e Barroso (2023) argumentam que, em termos práticos, isto não é possível. Portanto, sugerem utilizar os estimadores usuais $\bar{\mathbf{x}}$ e $\mathbf{S}$, respectivamente, dando suporte para a definição 3.3.5. Os resultados passam a ser aproximados, com melhor desempenho para grandes amostras.

3.3.4 Distância de cossenos

Seja $\mathbf{A}$ e $\mathbf{B}$ vetores não nulos. Então, pela Lei dos cossenos:

$$\mathbf{A} \cdot \mathbf{B} = |\mathbf{A}||\mathbf{B}| \cos\theta. \tag{3.3.6}$$

Ricardo e Berthier (2011) argumentam que podemos utilizar, para avaliar o grau de similaridade entre os vetores $\mathbf{A}$ e $\mathbf{B}$, a correlação entre eles. E para quantificar esta correlação, podemos utilizar o cosseno do ângulo entre estes vetores. Da Equação 3.3.6, temos que esta similaridade será:

$$\begin{aligned} \text{similaridade}(\mathbf{A}, \mathbf{B}) = \cos\theta &= \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}||\mathbf{B}|} \\ &= \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} \mathbf{A}_i^2} \cdot \sqrt{\sum_{i=1}^{n} \mathbf{B}_i^2}}. \end{aligned}$$

A Figura 3, ilustra o ângulo θ formado entre os vetores $\mathbf{A}$ e $\mathbf{B}$.

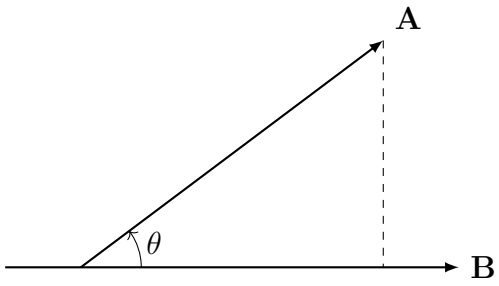

Figura 3: O cosseno como medida de similaridade entre $\mathbf{A}$ e $\mathbf{B}$

Se considerarmos $\mathbf{A}$ e $\mathbf{B}$ os vetores numéricos (vetorização) das petições de dois processos, conseguimos assim calcular uma medida de similaridade entre eles.

Note que a distância do cosseno não respeita a desigualdade triangular definida em 3.3, portanto não pode ser considerada uma medida de distância estatística naquelas definições.

3.4 Métodos de agrupamento hierárquicos

Segundo von Borries e Wang (2009), métodos hierárquicos buscam dividir ou agrupar um conjunto de dados em uma sequência de partições aninhadas, segundo alguma medida de similaridade (ou dissimilaridade). Estes dividem-se em dois grupos: métodos aglomerativos e divisivos.

3.4.1 Aglomerativo

Este método se inicia com uma observação em cada grupo (passo 0), e a cada iteração as observações são agrupadas ao seu par mais similar de acordo com alguma medida de similaridade ou dissimilaridade, até que todas as observações estejam em um único grupo. Alguns exemplos de algoritmos que executam este tipo de agrupamento são:

1. Ligação Simples: Une os elementos mais próximos;
2. Ligação Completa: Une os elementos mais distantes;
3. Ligação média: Une os elementos de acordo com suas respectivas medóides;
4. Ward: Faz novos grupos baseado na soma de quadrados das distâncias de todas as possíveis combinações de pares.

A Figura 4 ilustra o fluxograma do algoritmo aglomerativo de forma geral.

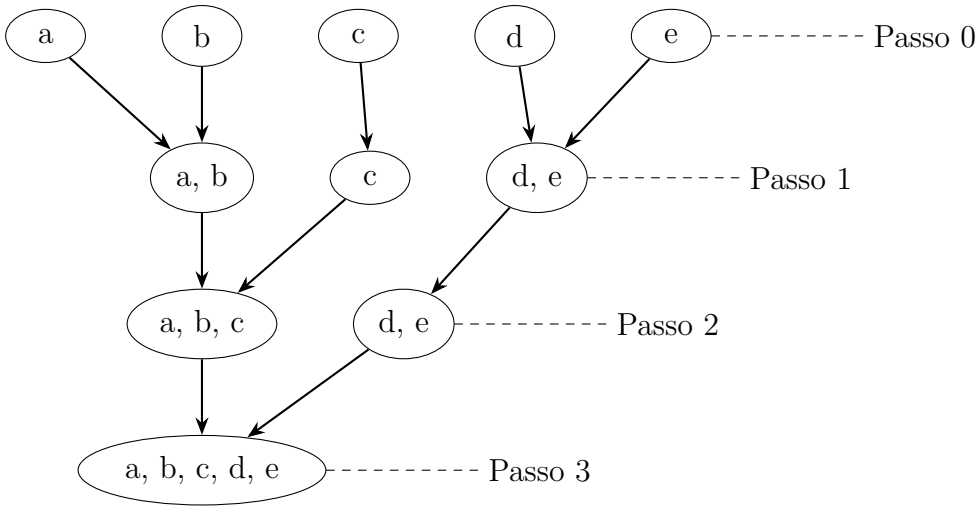

Figura 4: Fluxograma do algoritmo aglomerativo

Para mais informações, consultar Kaufman e Rousseeuw (1990)

3.4.2 Divisivo

Faz o caminho inverso do aglomerativo, ou seja, inicia o procedimento com todas as observações em um único grupo, e a cada iteração as observações são separadas utilizando alguma medida de similaridade ou dissimilaridade, até que cada observação forme um grupo contendo apenas ela. Este método foi primeiramente apresentado por Kaufman e Rousseeuw (1990) com o algoritmo DIANA (*Divisive Analysis* - Análise divisiva).

A Figura 5 ilustra o fluxograma do algoritmo divisivo de forma geral.

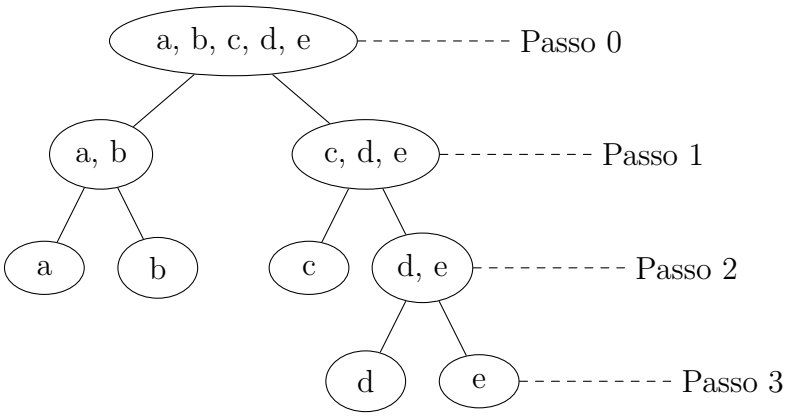

Figura 5: Fluxograma do algoritmo divisivo

3.5 Métodos de agrupamento não hierárquicos

Os métodos não hierárquicos buscam encontrar uma partição dos n elementos em g grupos. Estes grupos são representados por suas medóides e possuem forma convexa (elipsoide), conforme mostrado em MacQueen et al. (1967)

3.5.1 k-médias

k-médias, geralmente evocado como *k-means*, ou ainda k-medóides (Artes e Barroso, 2023) é um algoritmo que, segundo Morettin e Singer (2021), busca agregar pontos em K grupos de forma a minimizar a soma dos quadrados das distâncias euclidianas entre o centro (meio) de cada grupo e os pontos que o compõem.

Seja k o número de grupos em que se deseja agrupar os dados. Seja $\mathbf{x} = \mathbf{x}_1, ..., \mathbf{x}_n$ o vetor de amostra. O procedimento realizado pelo algoritmo pode ser descrito como:

- Fixar um número de k grupos (O usuário precisa especificar este número);
- Alocar os elementos aleatoriamente aos k grupos, e calcular a medóide $\bar{\mathbf{x}}_k$ de cada grupo;

- Calcular a distância Euclidiana de cada item ao medóide do grupo em que está alocado:

$$d_{i,\bar{x}_k} = \sum_{k=1}^{k} \sum_{g_{(i)}=k} (\mathbf{x}_i - \bar{x}_k)^T (\mathbf{x}_i - \bar{x}_k). \tag{3.5.1}$$

  em que $\bar{x}_k$ é a medóide do grupo k, e $g_{(i)}$ é o grupo que contém $\mathbf{x}_i$

- Realocar cada item $\mathbf{x}_i$ ao grupo com medóide mais próximo tal que reduza $d_{i,\bar{x}_k}$. Então, calcular os novos valores de $\bar{x}_k$.

- Repetir os dois últimos passos de forma iterativa, até estabilizar o valor de $d_{i,\bar{x}_k}$.

Este algoritmo é especialmente útil quando temos alguma ideia de quantos k grupos desejamos separar os dados.

No paradigma deste trabalho, não seria possível definir de forma rígida a quantidade de k grupos ideal para o conjunto de dados. O interesse é em realizar agrupamento — não classificação — dos dados, pois não existem categorias rígidas em que os processos se encaixariam, se tratando de categorias dinâmicas de acordo com os assuntos em voga no acervo. Ainda assim, o algoritmo k-médias pode ser útil para analisar de forma exploratória os dados, sendo possível testar diversos valores de k e analisar como os dados se agrupam em cada teste. Além disso, por se tratar de um algoritmo extremamente popular, torna possível a identificação e comparação com trabalhos análogos, e possibilitando o avanço do estado da arte do projeto com algoritmos mais sofisticados em uma possível futura aplicação.

3.6 Conjunto de dados

O conjunto de dados deste trabalho é dado pela petição inicial (texto em formato PDF) dos processos de controle concentrados tramitados ou em tramitação entre 1º de Janeiro de 2016 e 10 de Abril de 2024. Os dados são públicos e encontram-se disponíveis em Corte aberta. Optou-se pelo recorte à partir de 2016 a fim de parear o recorte deste estudo com o recorte estabelecido por outras aplicações do Supremo Tribunal Federal, como a RAFA2030.

A aplicação da ocerização e pré-processamento do texto, bem como algoritmo de vetorização, foram fornecidos pela Secretaria de Tecnologia e Inovação (STI) do tribunal para a manufatura deste trabalho.

Como os agrupamentos são construídos a partir dos dados que compõem o acervo atual do tribunal em uma determinada data, será testada a técnica em diversos recortes temporais dos dados obtidos, considerando em cada teste apenas os dados que compõem o acervo na data fixada de cada teste.

Não existe uma forma analítica simples de avaliar quais processos compunham o acervo em uma determinada data. De forma simplificada, um processo compõe o acervo no início de sua autuação, e deixa de compor quando tem baixa definitiva. Entretanto, retiradas de pauta, suspensões, trocas de relatoria e outras situações eventuais podem mover o processo como pertencente ou não ao acervo. Apesar disso, essas exceções não serão determinantes para a determinação da composição das amostras, visto que simplesmente ao considerar a diferença da data da baixa pela data de autuação do processo, é possível determinar com grande fidedignidade o acervo na data selecionada (especialistas do tribunal afirmam que é possível obter mais de 95% da composição do acervo desta forma). Como a intenção da aplicação é ser utilizada no acervo atual — que aí sim, sabemos com precisão o acervo da data presente em tempo real — sendo as datas passadas apenas para avaliação do modelo, esta métrica será utilizada sem embargos para definição da composição de cada amostra.

Visto que o painel de transparência do tribunal traz um quantitativo de acervo de controle concentrado por ano, optou-se por realizar um teste para cada um dos anos do recorte temporal estabelecido (2016 à 2024), fixando a data em primeiro de janeiro de cada ano. Portanto, serão nove conjuntos de amostra em que a aplicação será testada para avaliação de sua eficácia.

3.7 Visualização de dados

Uma parte crucial deste trabalho será mostrar de forma simplificada as indicações de agrupamentos fornecidos pela modelagem. Para isso, iremos explorar algumas técnicas de visualização de dados para estes resultados, afim de auxiliar no processo de tomada de decisões. Duas ferramentas particularmente úteis para este tipo de modelagem são dendogramas e *t-SNE*, que serão definidos a seguir.

3.7.1 Dendograma

Pela definição contida em Everitt e Skrondal (2010), dendograma é um diagrama geralmente utilizado para ilustrar a série de etapas executada por um método hierárquico, representando os passos de agrupamento do algoritmo. A altura do eixo y representa alguma medida de distância entre os agrupamentos. A Figura 6 ilustra um dendrograma genérico.

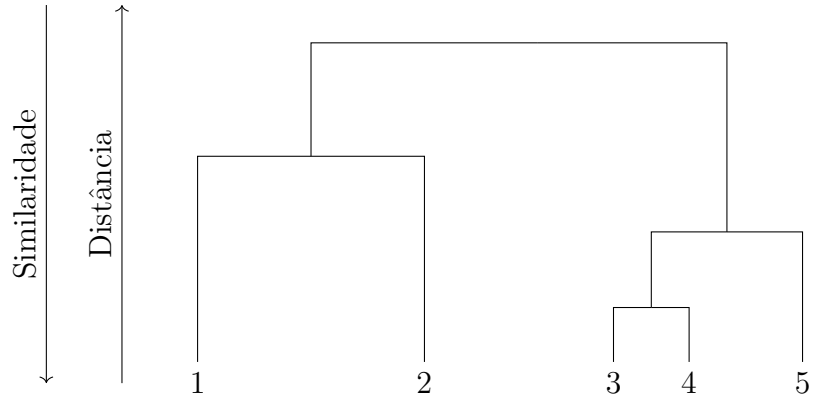

Figura 6: Exemplo simples de um dendograma

3.7.2 t-SNE

Agrupamento estocástico de vizinhança com distribuição t-student — *t-distributed stochastic neighbor embedding (t-SNE)*, tradução livre — é uma técnica não linear de visualização de dados multidimensionais em duas (ou três) dimensões. Esta técnica busca apresentar a estrutura de todo um conjunto de dados multidimensional com escalas diversas em um só gráfico.

A Figura 7 traz o exemplo da técnica *t-SNE* para visualizar o conjunto de dados MNIST, de LeCun et al. (1998)

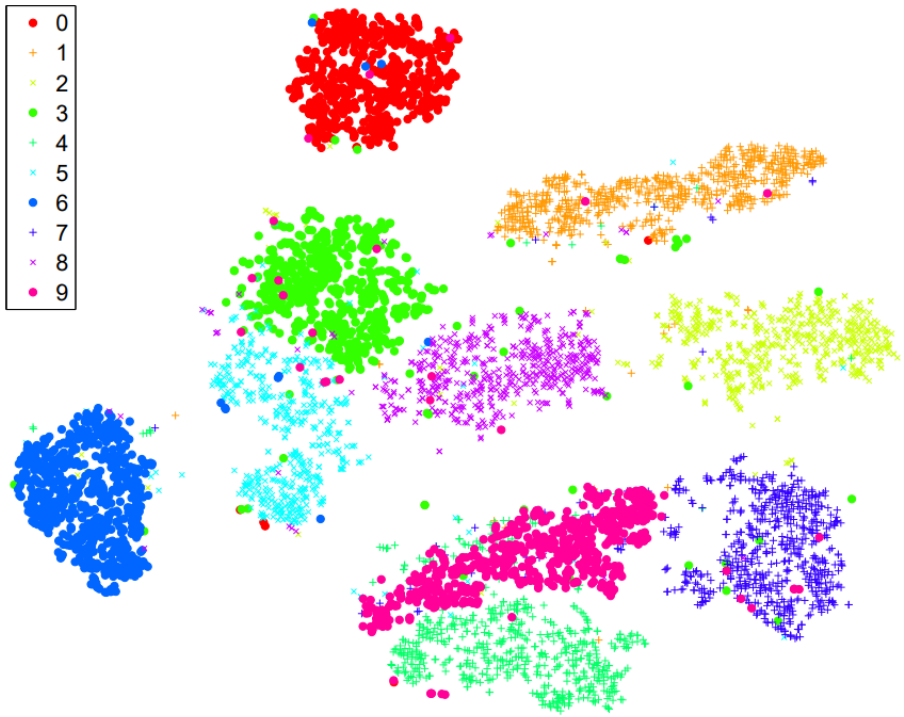

Figura 7: O algoritmo t-SNE aplicado aos dados MNIST.
Fonte: Maaten e Hinton (2008)

O conjunto de dados MNIST (LeCun et al, 1998) se tornou "padrão ouro" para teste de técnicas nas diversas áreas de reconhecimento de padrões, como análise multivariada clássica, processamento de linguagem natural, etc.

A técnica *t-SNE* busca apresentar de forma visual o resultado produzido pelo algoritmo utilizado. Por vezes, técnicas clássicas de visualização multivariada produzem visualizações de difícil interpretação, com interpolações dos agrupamentos formados na projeção dos resultados em duas dimensões. Para evitar estes problemas, utilizaremos a técnica *t-SNE*, em que Maaten e Hinton (2008) formularam o procedimento nos 4 passos a seguir:

- Passo 1: Converter distâncias Euclidianas de dados em probabilidades que representem similaridades

Na notação de Maaten e Hinton (2008), seja pontos x_j e x_i em um espaço N-dimensional, N grande. Iremos modelar a similaridade entre estes pontos como a probabilidade condicional $p(j|i)$ a probabilidade condicional de x_j escolher x_i como seu vizinho, tal que estas probabilidades condicionais sejam proporcionais a similaridade entre estes pontos. Para calcular esta probabilidade condicional de $p(j|i)$, Hinton e Roweis (2002) definem que esta probabilidade condicional $p(j|i)$ é proporcional a uma densidade de probabilidade gaussiana centrada em x_i e com variância σ_i $\forall i \neq j$, ou seja:

$$p(j|i) = \begin{cases} \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)}, \forall i \neq j; \\ 0, i = j. \end{cases} \tag{3.7.1}$$

Forçamos esta probabilidade $p(i|i) = 0$ pois estamos somente interessados em modelar similaridades pareadas.

- Passo 2: Mapear os pontos $x_1, ..., x_k$ em alta dimensionalidade para pontos $y_1, ..., y_k$ em dimensionalidade reduzida.

Seja $y_1, ..., y_k$ os pontos de dimensionalidade reduzida referentes aos pontos $x_1, ..., x_k$. Seja $q(j|i)$ a probabilidade condicional de y_j escolher o ponto y_i como seu vizinho, de acordo com a distância Euclidiana entre eles, e q_{ij} a probabilidade conjuta destes pontos. Segundo Maaten e Hinton (2008), a probabilidade conjunta q_{ij} é proporcional a uma densidade de probabilidade $t - student$ com 1 grau de liberdade (ou seja, proporcional à uma Cauchy), definindo assim:

$$q_{ij} = \begin{cases} \frac{(1+||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1+||y_k - y_l||^2)^{-1}}, \forall i \neq j; \\ 0, i = j. \end{cases} \tag{3.7.2}$$

Em contraposição assim ao algoritmo SNE proposto por Hinton e Roweis (2002), que utilizaria uma gaussiana ao invés da *t-student*. Esta troca por uma distribuição de cauda pesada é feita para evitar sobreposições de pontos, garantindo assim o padrão típico da visualização $t - SNE$, que faz uma separação mais robusta entre os agrupamentos, tornando a técnica extremamente eficaz em separar pontos dissemelhantes para visualização e interpretação gráfica. A justificativa teórica para utilizar a *t-student* é que "esta distribuição está diretamente relacionada à distribuição Gaussiana, dado que a distribuição *t-student* é uma mistura de infinitas Gaussianas"(Maaten e Hinton, 2008, p. 2586, tradução própria[2]), com médias iguais e precisões como variável aleatória gama.

- Passo 3: Minimizar a soma da divergência de Kullback-Leibler para as probabilidades conjuntas de todos os pontos, utilizando gradiente descentente.

A justificativa para o funcionamento da técnica, segundo Maaten e Hinton (2008), é que se o mapeamento dos pontos y_i, y_j — em dimensionalidade reduzida — modelar com fidedignidade a similaridade entre os pontos x_i e x_j, então as probabilidades condicionais $p(j|i)$ e $q(j|i)$ serão iguais. Portanto, o algoritmo busca justamente minimizar a

[2]No original: "the Student t-distribution is that it is closely related to the Gaussian distribution, as the Student t-distribution is an infinite mixture of Gaussians".

diferença entre $p(j|i)$ e $q(j|i)$, e para isso, o algoritmo minimiza a soma da divergência de Kullback-Leibler (que neste caso é proporcional a entropia cruzada) para as probabilidades condicionais em todos os pontos utilizando o método de gradiente descendente. Por motivos de otimização computacional, o algoritmo busca minimizar a soma da divergência de Kullback-Leibler para as probabilidades conjuntas p_{ij}, definindo estas como simétricas tal que $p_{ij} = \frac{p(j|i)+p(i|j)}{2n}$, justificando assim a utilização da probabilidade conjunta em (3.7.2).

- Passo 4: Plotar os pontos y_k.

Note que, a termos práticos, $\mathbf{x_1}, ..., \mathbf{x_n}, \mathbf{y_1}, ..., \mathbf{y_n}$ serão vetores, de um conjunto $\mathbf{N}$ dimensional para os vetores $\mathbf{x_k}$, e $\mathbf{n} < \mathbf{N}$ dimensional para os vetores $\mathbf{y_k}$, sem perda de generalidade.

Portanto, o *t-SNE* busca separar no gráfico os agrupamentos formados, esclarecendo a estrutura dos resultados do procedimento de forma visualmente compreensível. Para mais detalhes do funcionamento da técnica, consultar Maaten e Hinton (2008).

4 Cronograma

As atividades a serem desenvolvidas durante o Trabalho de Conclusão de Curso são:

Tabela 1: Cronograma TCC 1

Atividades	1/2024				
	Mar	Abr	Mai	Jun	Jul
Escolha do tema a ser abordado.					
Levantamento de bibliografia relacionada ao tema.					
Definição do recorte temporal com a AAJ do STF.					
Solicitação dos dados para a STI do STF.					
Solicitação dos algoritmos à STI do STF.					
Revisão de literatura.					
Desenvolvimento da proposta de projeto.					
Análise preliminar do banco de dados.					
Entrega da proposta de projeto.					
Elaboração da apresentação da proposta.					
Manipulação do banco de dados.					
Análise do banco de dados.					
Elaboração do relatório parcial.					
Entrega do relatório parcial a Professora Orientadora.					
Correção do relatório parcial.					
Entrega do relatório parcial a banca.					

Tabela 2: Cronograma TCC 2

Atividades	2/2024				
	Ago	Set	Out	Nov	Dez
Desenvolvimento do modelo e da aplicação.					
Elaboração do relatório final.					
Entrega do relatório final ao Professor Orientador(a).					
Correção do do relatório final.					
Entrega do relatório final para a banca.					

Referências

ARTES, R.; BARROSO, L. P. *Métodos multivariados de análise estatística*. [S.l.]: São Paulo: Blucher, 2023.

BOLUKBASI, T. et al. *Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings*. 2016.

EVERITT, B.; SKRONDAL, A. *The cambridge dictionary of statistics*. [S.l.]: Cambridge University Press, 2010. v. 4.

FIRTH, J. R. *Studies in linguistic analysis*. 8ª. ed. [S.l.]: Blackwell, Oxford, 1962.

FREITAS, L. J. G. et al. Catboost algorithm application in legal texts and un 2030 agenda. *Revista de Informatica Teórica e Aplicada - RITA - ISSN 2175-2745. Vol. 30, Num. 02 (2023) 51-58*, 2023.

FREITAS, L. J. G. et al. Text clustering applied to data augmentation in legal contexts. *arXiv preprint arXiv:2404.08683*, 2024.

HINTON, G. E.; ROWEIS, S. Stochastic neighbor embedding. MIT Press, v. 15, 2002. Disponível em: ⟨https://proceedings.neurips.cc/paper_files/paper/2002/file/6150ccc6069bea6b5716254057a194ef-Paper.pdf⟩.

JOHNSON, R. A.; WICHERN, D. W. *Applied Multivariate Statistical Analysis*. [S.l.]: 6. ed.[S.l.]:Prentice Hall, 2007.

KAUFMAN, L.; ROUSSEEUW, P. J. *Finding groups in data: an introduction to cluster analysis*. [S.l.]: John Wiley & Sons, 1990.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Ieee, v. 86, n. 11, p. 2278–2324, 1998.

MAATEN, L. Van der; HINTON, G. Visualizing data using t-sne. *Journal of machine learning research*, v. 9, n. 11, 2008.

MACQUEEN, J. et al. *Some methods for classification and analysis of multivariate observations*. [S.l.], 1967. v. 1. 281–297 p.

MORETTIN, P. A.; SINGER, J. M. *Estatística e Ciência de Dados*. [S.l.]: LTC, 2021.

RICARDO, B.-Y.; BERTHIER, R.-N. *Modern information retrieval: the concepts and technology behind search*. [S.l.]: New Jersey, USA: Addi-son-Wesley Professional, 2011.

SARICA, S.; LUO, J. Stopwords in technical language processing. *Plos one*, Public Library of Science San Francisco, CA USA, v. 16, n. 8, p. e0254937, 2021.

SILVA, T. *Racismo algorítmico: inteligência artificial e discriminação nas redes digitais*. [S.l.: s.n.], 2022. ISBN 978-65-86111-70-5.

VEROUTIS, P.; FAJARDO, E. Markov chain monte carlo methods in cryptography. 2021.

von Borries, G.; WANG, H. Partition clustering of high dimensional low sample size data based on p-values. *Computational statistics & data analysis, v. 53, n. 12, p. 3987-3998*, 2009.