UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE QUIXADÁ

BACHARELADO EM ENGENHARIA DE SOFTWARE

PAULO HENRIQUE NOBRE

RECOMENDAÇÃO DE PRODUTOS BASEADA EM PERFIS DE CONTATOS DE CLIENTES E POPULARIDADE DE PRODUTOS

QUIXADÁ

2021

PAULO HENRIQUE NOBRE

RECOMENDAÇÃO DE PRODUTOS BASEADA EM PERFIS DE CONTATOS DE CLIENTES E POPULARIDADE DE PRODUTOS

Trabalho de Conclusão de Curso apresentado ao Curso de graduação em Engenharia de Software do campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software. Área de concentração: Computação.

Orientador: Prof. Dr. Marcos Antônio de Oliveira.

QUIXADÁ

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

N673r Nobre, Paulo Henrique.
Recomendação de produtos baseada em perfis de contatos de clientes e popularidade de produtos / Paulo Henrique Nobre. – 2021.
49 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2021.
Orientação: Prof. Dr. Marcos Antônio de Oliveira.

1. Sistemas de recomendação (filtragem de informações) 2. Aprendizado do computador. I. Título.
CDD 005.1

PAULO HENRIQUE NOBRE

RECOMENDAÇÃO DE PRODUTOS BASEADA EM PERFIS DE CONTATOS DE CLIENTES E POPULARIDADE DE PRODUTOS

Trabalho de Conclusão de Curso apresentado ao Curso de graduação em Engenharia de Software do campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software. Área de concentração: Computação.

Aprovada em: ___/___/_____.

BANCA EXAMINADORA

_____
Prof. Dr. Marcos Antônio de Oliveira (Orientador)
Universidade Federal do Ceará (UFC)

_____
Profa. Ma. Lívia Almada Cruz
Universidade Federal do Ceará (UFC)

_____
Prof. Dr. Regis Pires Magalhães
Universidade Federal do Ceará (UFC)

À minha família, em especial minha mãe e meu padrasto. À minha esposa pelo incentivo e todo o apoio. À minha filha por ter mudado minha vida. A todos meus amigos, professores, orientador.

AGRADECIMENTOS

Ao Prof. Dr. Marcos Antônio de Oliveira, pela orientação.

A Sabrina Alves da Silva por todo apoio prestado.

“A inteligência é o que você usa quando não sabe o que fazer.”

(Jean Piaget)

RESUMO

Os sistemas de *Customer Relationship Management* (CRM) possuem uma grande quantidade de informações sobre contatos. Saber quais produtos devem ser oferecidos a essas pessoas não é uma tarefa simples, devido a grande quantidade e diversidade entre os produtos, então realizar recomendações personalizadas para cada perfil de contato pode ajudar significativamente no processo de vendas. Com base nisso, este trabalho apresenta um sistema recomendador de produtos para contatos, fazendo uso de recomendação baseada em popularidade de produtos e recomendação por filtragem colaborativa. Na recomendação baseada em popularidade é criado um *ranking* com os produtos mais vendidos, e os produtos no topo desse *ranking* são recomendados. Na recomendação por filtragem colaborativa são utilizadas duas métricas: a distância de *Levenshtein* e a similaridade de cosseno. Com cada métrica são criadas listas de contatos com perfis semelhantes para cada contato, e a partir dessas listas os produtos são recomendados. Também é aplicada clusterização nos dados, utilizando as métricas de cosseno e *Levenshtein.* Ao final, os resultados dos métodos de recomendações são comparados, onde os métodos que usam clusterização obtiveram melhores resultados.

Palavras-chave: Sistemas de Recomendação. Filtragem colaborativa. Similaridade Cosseno. Aprendizagem de Máquina. K-Medoids.

ABSTRACT

Customer Relationship Management (CRM) systems have a large amount of contact information. Knowing which products should be offered to these people is not a simple task, due to the large quantity and diversity among the products, so making personalized recommendations for each contact profile can help in the sales process. Based on this, this work presents a product recommendation system for contacts, using recommendation based on product popularity and recommendation using collaborative filtering. In the popularity-based recommendation, a ranking is created with the best-selling products, and the products at the top of that ranking are recommended. In the recommendation for collaborative filtering, two metrics are used: the Levenshtein distance and the cosine similarity. With each metric contact lists with similar profiles are created for each contact, and from these lists the products are recommended. Clustering is also applied to the data, using the cosine and Levenshtein metrics. At the end, the results of the recommendation methods are compared, where the methods that use clustering obtained better results.

Keywords: Recommendation Systems. Collaborative Filtering. Cosine Similarity. Machine Learning. K-Medoids.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SR	Sistema de Recomendação
CRM	*Customer Relationship Management*
AM	Aprendizado de Máquina
IA	Inteligência Artificial
PAM	*Partitioning Around Medoids*
RDS	*Amazon Relational Database Service*
CSV	*Comma-separated Values*
PCA	*Principal Component Analysis*

SUMÁRIO

1 INTRODUÇÃO

Os sistemas de recomendação (SRs) a anos vêm sendo utilizados em sistemas computacionais, com objetivo de ajudar os usuários na tomada de decisões, frente a grande sobrecarga de informações que existe atualmente. Os SRs podem recomendar a um usuário quais itens comprar, quais músicas ouvir, quais filmes assistir, quais notícias ler, e muitas outras coisas, tudo isso com base em seu perfil de consumo.

Os SRs estão presentes na maioria dos serviços de comércio eletrônicos, serviços de *streaming*, redes sociais, entre muitas outras. Redes sociais como Facebook[1] e Instagram[2] fazem uso de SRs para sugerir novas conexões, plataformas de *streaming* como Netflix[3] e Amazon Prime Video[4] recomendam filmes e séries com base nas preferências do usuário, e a Amazon[5] faz recomendação de produtos em seu *e-commerce*.

Em 2000 a Netflix introduziu um SR chamado *Cinematch*, que foi aprimorado ao longo dos anos. Esse sistema fez com que a Netflix obtivesse 60% de suas vendas realizadas através de recomendações (THOMPSON, 2008). A Amazon usa diversas técnicas para recomendar itens, como recomendados para você, frequentemente comprados juntos, ou mais vendidos na categoria. O sistema de recomendação é responsável por gerar 35% das receitas através de suas recomendações (MACKENZIE et al., 2013).

Nos sistemas de *Customer Relationship Management* (CRM) existe uma grande quantidade de *leads* e contatos. Saber quais de seus produtos podem ser oferecidos para um determinado perfil é algo que pode auxiliar significativamente no processo de vendas.

1.1 Objetivos

Nesta seção serão apresentados os objetivos deste trabalho.

1.1.1 Objetivo Geral

Este trabalho tem como objetivo criar um sistema de recomendação capaz de realizar recomendações de programas a contatos, com base nas informações do perfil do contato.

1.1.2 Objetivos Específicos

[1] https://www.facebook.com/
[2] https://www.instagram.com/
[3] https://www.netflix.com/
[4] https://www.primevideo.com/
[5] https://www.amazon.com.br/

- Criar sistema de recomendação por popularidade de produtos;
- Criar sistema de recomendação por filtragem colaborativa usando similaridade de cosseno;
- Criar sistema de recomendação por filtragem colaborativa usando distância de *Levenshtein*;
- Criar sistema de recomendação por clusterização usando similaridade de cosseno;
- Criar sistema de recomendação por clusterização usando distância de *Levenshtein*;
- Comparação e interpretação dos resultados obtidos.

1.2 Organização

O restante deste trabalho está organizado da seguinte forma: no Capítulo 2 é abordada a fundamentação teórica com os principais conceitos necessários para a realização deste trabalho. No Capítulo 3 apresentam-se os trabalhos relacionados. No Capítulo 4 são apresentados os procedimentos metodológicos para a execução deste trabalho. No Capítulo 5 são detalhados os resultados obtidos e, por fim, no Capítulo 6 são mostradas as conclusões e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos para o entendimento e execução deste trabalho. A seção 2.1, apresenta os conceitos de Sistema de Recomendação; na seção 2.2, são apresentados os conceitos sobre Aprendizagem de Máquina; na seção 2.3, são apresentados os conceitos sobre clusterização.

2.1 Sistema de recomendação

Um sistema de recomendação (SR) tem o objetivo de sugerir itens de forma a satisfazer sob algum aspecto as necessidades de um usuário. Geralmente, esses sistemas atuam em contextos onde a tomada de decisão sobre a escolha de itens se dá em um conjunto grande de opções, no qual uma busca por meio de mecanismos clássicos como palavras-chaves ou termos de interesse tem a chance de retornar resultados insatisfatórios (BRUNIALTI et al., 2015).

Os Sistemas de Recomendação são ferramentas de software e técnicas que fornecem sugestões de itens que podem ser usados pelo usuário. Estas sugestões estão relacionadas a vários processos de tomada de decisão, como quais itens comprar, que músicas ouvir ou quais notícias ler (RICCI et al., 2011).

Item é o termo geral usado para denotar o que o sistema recomenda aos usuários. Um SR normalmente tem foco em um tipo específico de item (i.e., *CDs* ou notícias) e, consequentemente, seu *design*, sua interface gráfica e a técnica de recomendação são todos personalizados para fornecer sugestões úteis e eficazes para aquele específico tipo de item (RICCI et al., 2011).

2.1.1 Recomendação baseada em conteúdo

Os sistemas de recomendação baseados em conteúdo recomendam itens a um usuário usando a semelhança dos itens. Este sistema de recomendação recomenda produtos ou itens com base em sua descrição ou características. Ele identifica a semelhança entre os produtos com base em suas descrições. Também considera o histórico anterior do usuário para recomendar um produto semelhante (SUBRAMANIAN, 2020).

2.1.2 Recomendação baseada em colaboração

O princípio do algoritmo da filtragem colaborativa considera que o usuário ativo possui maior probabilidade de se interessar por itens que usuários semelhantes preferem ou preferiram. Para isto, calcula-se um grau de similaridade entre o usuário ativo (alvo) e os outros usuários. Os itens com maior grau de similaridade são recomendados ao usuário alvo (RESNICK; VARIAN, 1997).

O sistema gera recomendações usando apenas informações sobre perfis de classificação para diferentes usuários. Os sistemas colaborativos localizam usuários pares com um histórico de classificação semelhante ao do usuário atual e geram recomendações usando esta vizinhança (BURKE, 2007).

2.2 Aprendizagem de máquina

Aprendizado de Máquina (AM) é uma área de Inteligência Artificial (IA) cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática (MONARD; BARANAUSKAS, 2003).

AM é uma subárea de pesquisa muito importante em IA, pois, a capacidade de aprender é essencial para um comportamento inteligente. A AM estuda métodos computacionais para adquirir novos conhecimentos, novas habilidades e novos meios de organizar o conhecimento já existente (MITCHELL, 1997).

2.2.1 Aprendizagem supervisionada

A aprendizagem supervisionada é a tarefa de aprendizado de máquina de aprender uma função que mapeia uma entrada para uma saída com base em pares de entrada-saída de exemplo (RUSSELL; NORVIG, 2010).

Os algoritmos geram uma função que mapeia as entradas para as saídas desejadas. Uma formulação padrão da tarefa de aprendizagem supervisionada é o problema de classificação: o aluno deve aprender (para aproximar o comportamento de) uma função que mapeia um vetor em uma das várias classes, observando vários exemplos de entrada-saída da função (NASTESKI, 2017).

2.2.2 Aprendizagem não supervisionada

É uma forma de aprendizado em que não é fornecido nenhum rótulo ou classificação ao algoritmo, ou mapeamento de entrada-saída, de forma que a aprendizagem ocorra por meio de observação e descoberta (REZENDE, 2003).

2.2.3 Aprendizagem por reforço

Os algoritmos aprendem uma política de como agir dada uma observação do mundo. Cada ação tem algum impacto no ambiente, e o ambiente fornece *feedback* que orienta o algoritmo de aprendizagem (NASTESKI, 2017).

A aprendizagem por reforço trata de aprender por meio de interação e *feedback*, ou em outras palavras, aprender a resolver uma tarefa por tentativa e erro, agindo em um ambiente e recebendo recompensas por isso (MONI, 2019).

2.3 Clusterização

A análise de *cluster* é a arte de encontrar grupos de dados. Basicamente, deseja-se formar grupos de forma que os objetos no mesmo grupo sejam semelhantes entre si, enquanto os objetos em grupos diferentes sejam tão diferentes quanto possível (KAUFMAN; ROUSSEEUW, 1990).

Os métodos de agrupamento são aplicados em muitos domínios, incluindo inteligência artificial e reconhecimento de padrões, quimiometria, ecologia, economia, geociências, marketing, pesquisa médica, ciência política, psicometria e muito mais (KAUFMAN; ROUSSEEUW, 1990).

2.3.1 Dissimilaridade

As dissimilaridades são números não negativos d(i,j) que são pequenos (próximos de zero) quando i e j estão próximos um do outro, e que se tornam maiores quando i e j são muito diferentes. Devemos geralmente supor que as dissimilaridades são simétricas e que a

dissimilaridade de um objeto em relação a si mesmo é zero (KAUFMAN; ROUSSEEUW, 1990).

2.3.1.1 Distância de Levenshtein

A distância de *Levenshtein* é uma medida entre duas *strings*, a *string* origem *s*, e a *string* destino *t*. A distância é o número de deleções, inserções ou substituições necessárias para transformar *s* em *t*. Quanto maior a distância de *Levenshtein*, mais dissimilares são as *strings* (HALDAR; MUKHOPADHYAY, 2011).

Na Figura 1 vemos que a quantidade de operações necessárias para transformar a palavra gato em rato é de 1operação, logo a distância de Levenshtein entre essas palavras é 1.

Figura 1- Distância de *Levenshtein*

		g	a	t	o
r		**1**	2	3	4
a		1	**1**	2	3
t		1	2	**1**	2
o		1	2	3	**1**

Fonte: Tomalok E.[6]

2.3.2 Similaridade

Em vez de usar um coeficiente de dissimilaridade d(*i,j*) para indicar quão remotos dois objetos *i* e *j* estão, é possível trabalhar com um coeficiente de similaridade s(*i,j*). Normalmente a similaridade assume valores entre 0 e 1, onde, 0 significa que os objetos não são semelhantes, e 1 reflete a similaridade máxima entre eles (KAUFMAN; ROUSSEEUW, 1990).

Existem diversas métricas de similaridade que são usadas por sistemas de recomendação na área de AM. É possível usar diferentes métricas para computar a similaridade entre objetos, de forma a obter resultados diferentes, já que as métricas possuem fórmulas diferentes.

2.3.2.1 Similaridade de Cosseno

[6] disponível em: https://medium.com/@everton.tomalok/calculando-similaridades-entre-strings-ebbea21d5b7a

A similaridade de Cosseno mede a similaridade de dois vetores de um espaço interno do produto. É medida pelo cosseno do ângulo entre os vetores, e determina se dois vetores estão apontando na mesma direção, e é geralmente usada para medir semelhanças em análise de texto (HAN et al., 2012).

Dados dois vetores x e y, usando a medida cosseno como função de similaridade, nós temos:

$$sim(x, y) = \frac{xy}{||x||||y||}$$

onde $||x||$ é a norma euclidiana do vetor $x = (x_1, x_2, \dots, x_p)$, definida por $\sqrt{x_1^2 + x_2^2 + \cdots + x_p^2}$.

2.3.3 K-Medoids

O algoritmo *Partitioning Around Medoids* (PAM), conhecido simplesmente como *k-medoids*, é um dos algoritmos mais populares para clusterização de dados não euclidianos. PAM usa medóides o que permite que seja utilizado com qualquer medida de dissimilaridade (SCHUBERT; ROUSSEEUW, 2019).

O algoritmo *K-medoids* é baseado na busca por k objetos representativos chamados medóides, que devem representar os diversos aspectos da estrutura de dados. O medóide de um *cluster* deve ser o objeto para o qual a dissimilaridade média de todos os objetos do *cluster* é mínima (KAUFMAN; ROUSSEEUW, 1987).

2.4 Método do cotovelo

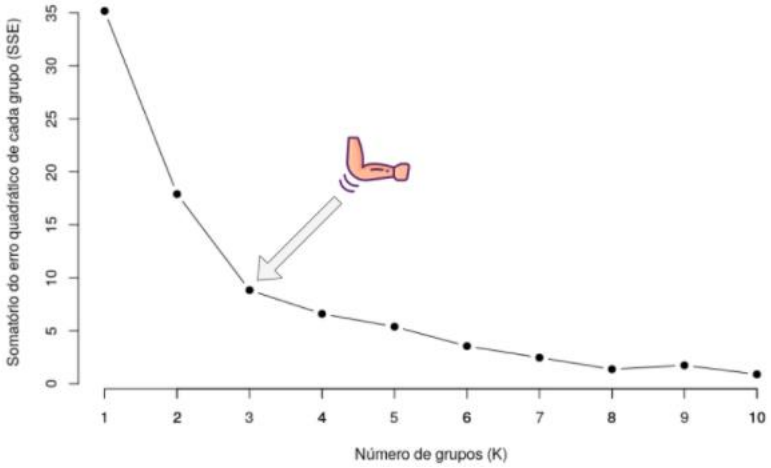
Chamado de *elbow method*, ou método do cotovelo, é uma metodologia para auxiliar o encontro do número apropriado de *clusters* em um conjunto de dados (THORNDIKE, 1953). O método tem esse nome por seu formato parecer com o de um braço, e formar um ângulo semelhante ao de um “cotovelo” ao definir os números aceitáveis de *clusters* a serem gerados.

A metodologia analisa a porcentagem da variação como uma função pelo número de *clusters*, ou seja, ao se traçar a porcentagem de variação pelos *clusters* em relação ao seu

número, o resultado trará uma variação muito alta em primeiro momento, mas após alguns pontos, esse número cai formando assim o ângulo do "cotovelo" (THORNDIKE, 1953).

A Figura 1 mostra que o número ideal de *clusters* é escolhido quando a porcentagem variação tende a não variar muito.

Figura 2 - Método do cotovelo

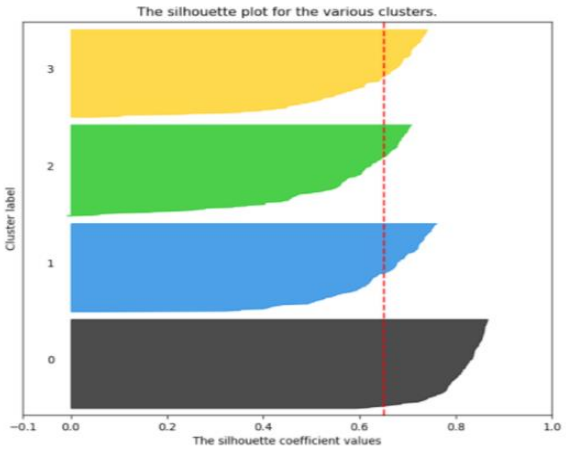


Fonte: Grupo DataAt[7]

2.5 Método da silhueta

O método da silhueta é um método que auxilia no encontro no número de clusters apropriados para um conjunto de dados (ROUSSEEUW, 1987). Ele permite visualizar graficamente através das silhuetas quais objetos estão bem dentro do *cluster* e quais objetos estão em apenas em posição intermediária, como mostra a Figura 2.

Figura 3 - Método de Análise da silhueta



[7] disponível em: https://dataat.github.io/introducao-ao-machine-learning/index.html#grupo-dataat
[8] disponível em: https://medium.com/cwi-software/entendendo-clusters-e-k-means-56b79352b452

Fonte: cwi Software[8]

A silhueta é um gráfico do *cluster* C formado por um valor de silhueta *s(i), i = 1, ..., n,* onde cada objeto do *cluster* é representado por *i*, e o valor *s(i)* é calculado por:

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

Onde *a(i)* é dissimilaridade média de *i* em relação a todos os objetos do *cluster* C, e *b(i)* é a dissimilaridade média entre *i* em relação a todos os objetos do *cluster* vizinho mais próximo a ele.

Quando *s(i)* está próximo de 1, implica que a dissimilaridade em *a(i)* é muito menor que a dissimilaridade em *b(i)*, o que quer dizer que *i* está bem classificado. Quando *s(i)* está próximo de 0, então *a(i) e b(i)* estão próximos, portanto, não está claro se *i* foi bem atribuído, e no pior caso quando *s(i)* está próximo de -1, então *a(i)* é muito maior que *b(i)*, logo *i* está muito mais próximo de B do que de A (KAUFMAN; ROUSSEEUW, 1990).

KAUFMAN e ROUSSEEUW (1990) propuseram uma interpretação subjetiva chamada coeficiente de silhueta *CS(i)*, o coeficiente de silhueta é definido como a largura máxima média da silhueta para todo o conjunto de dados (Tabela 1).

Tabela 1 - Agrupamentos segundo coeficiente de silhueta (*CS(i)*)

CS(i)	**Interpretação Sugerida**
0.71 – 1.00	Grupos descobertos possuem uma estrutura muito robusta.
0.51 − 0.70	Grupos possuem uma estrutura razoável.
0.26 − 0.50	Os grupos encontrados possuem uma estrutura fraca e pode ser artificial. É aconselhável tentar outros métodos sobre o conjunto de dados.
≤ 0.25	Nenhuma estrutura foi descoberta.

Fonte: Adaptada de (KAUFMAN; ROUSSEEUW, 1990)

3 TRABALHOS RELACIONADOS

Neste capítulo é apresentado o trabalho de Teixeira (2018) que utiliza sistemas de recomendação para preenchimento de *playlists* de músicas. Também é apresentado o trabalho de Leite (2019), que usa sistema de recomendação semi-supervisionado, utilizando filtragem baseada em conteúdo e filtragem colaborativa, para indicação de restaurantes com base nos perfis de clientes. Por fim é apresentado o trabalho de Henrique (2019) que faz recomendações de rotas seguras para ciclistas com uso do algoritmo *Dijkstra*.

Na seção 3.4 é apresentado um quadro com os comparativos entre este trabalho e os trabalhos relacionados.

3.1 Preenchimento de playlists utilizando técnicas de sistemas de recomendação baseadas em filtragem colaborativa

Em Teixeira (2018) é apresentada uma solução de um sistema recomendador de músicas para preenchimento de *playlist* utilizando filtragem colaborativa. O trabalho utiliza *playlists* feitas por usuários do Spotify, e tem como objetivo recomendar músicas, comparar os resultados e precisões dos métodos utilizados.

No pré-processamento dos dados, os mesmos foram separados em dados de treino e dados de teste, onde treino consiste em 70% dos dados, e teste nos 30% restantes. Para recomendação foram utilizados recomendação baseada em popularidade e recomendação baseada em similaridade.

Na recomendação baseada em popularidade foi criado um *ranking* com as músicas mais populares dos dados de treino, ordenadas pela sua popularidade. As *playlists* incompletas dos dados de testes foram preenchidas com as músicas mais populares, excluindo as músicas já presentes na *playlist*.

Na recomendação baseada em similaridade foi utilizado a similaridade cosseno e coeficiente de Jaccard, onde cada *playlist* dos dados de treino foi comparada uma a uma com as demais *playlists*, de forma que no final cada *playlist* possuía um ranking das 5 *playlists* mais similares. As *playlists* incompletas dos dados de testes foram preenchidas com as músicas das *playlists* que atingiram os maiores valores em casa similaridade.

Na etapa de avaliação foi realizado com cálculo em R para cada abordagem, com intuito de fazer um comparativo entre os algoritmos de recomendação. A recomendação

utilizando o coeficiente de Jaccard foi a que obteve o melhor resultado, seguido pela similaridade cosseno.

3.2 Sistema de recomendação aplicado em plataformas de reserva *online* de restaurante

Em Leite (2019) é construído um modelo de sistema de recomendação a partir do histórico de uso de clientes e de seus restaurantes favoritos. Foram extraídos dados utilizando filtragem colaborativa e filtragem baseada em conteúdo, e a partir dos dados extraídos foi realizado um agrupamento dos dados por meio de clusterização usando o método *K-means*.

Os dados do trabalho foram obtidos a partir de um aplicativo de *food service*, depois de obtidos os dados passaram por um processo de normalização para remover informações incorretas, ou fora dos padrões da base. Em seguida os dados passaram por um analisador onde os dados foram analisados e recomendados com base no método *K-means*.

Após a análise, os dados recomendados foram organizados em modelos de objetos, e depois apresentados aos usuários através de um aplicativo *mobile*. Foram realizados testes com usuários a fim de validar os resultados do trabalho, e por meio desses testes, obteve-se que 70% das recomendações foram bem positivas para os usuários.

3.3 Desenvolvimento de um sistema capaz de recomendar rotas seguras para ciclistas

No trabalho de Henrique (2019) é feito um sistema recomendador de rotas seguras para ciclistas, que além de considerar a distância entre os pontos, também considera os locais perigosos, evitando rotas que passem por esses lugares. Os dados das ruas utilizados foram referentes às ruas da cidade de Fortaleza, assim como os dados sobre violência.

Com os dados coletados as ocorrências sobre roubo de veículos, roubo a pessoas e morte com arma de fogo foram projetadas no mapa através da geolocalização de cada ocorrência. A estratégia escolhida para traçar rotas foi a de penalizar as rotas em que existiam pontos de ocorrências, e o algoritmo de *Dijkstra* foi utilizado para calcular as rotas com os menores custos, o algoritmo *Dijkstra* faz o cálculo do menor custo utilizando distância x penalidade.

A Figura 3 ilustra um exemplo de rota sendo recomendada em azul, enquanto a rota mais curta em vermelho está penalizada.

Figura 4 - Rota recomendada

Fonte: Henrique (2019)

3.4 Comparativo entre os trabalhos

O Trabalho de Teixeira (2018) se assemelha a este trabalho por utilizar filtragem colaborativa para realizar as recomendações, e por fazer uso de métricas como popularidade e similaridade de cosseno. O trabalho de Leite (2019) se assemelha a este trabalho por realizar recomendações com uso de filtragem colaborativa, e por realizar clusterização nos dados. O Trabalho de Henrique (2019) se assemelha a este trabalho por realizar recomendações.

O Quadro 2 apresenta um comparativo entre as características deste trabalho com os trabalhos relacionados.

Quadro 1 - Comparativo de trabalhos relacionados

	Contexto	Técnicas de Filtragem	Métricas	Clusterização
(TEIXEIRA, 2018)	Preenchimento de *Playlists*	Colaborativa	Popularidade, Cosseno, Jaccard	-
(LEITE, 2019)	Recomendação de restaurantes	Colaborativa, Baseada em Conteúdo	Distância Euclidiana	*K-Means*
(HENRIQUE, 2019)	Recomendação de rotas	-	*Dijkstra*	-
Este Trabalho	Recomendação de produtos	Colaborativa	Popularidade, Cosseno, *Levenshtein*	*K-Medoids*

Fonte: Elaborado pelo autor

4 METODOLOGIA

Neste capítulo, é descrito todo o processo de implementação deste trabalho, desde os primeiros passos para a obtenção dos dados, até o desenvolvimento do sistema de recomendação, e as validações dos resultados obtidos.

4.1 Obtenção dos dados

Neste estudo são utilizados dados de um CRM pertencente a uma empresa que atua no Ceará e no Rio Grande do Norte. Essa empresa atua através da prestação de serviços de consultorias em gestão, governança, educação e soluções inovadoras e personalizadas.

4.2 Dados de treino e dados de teste

Os dados são divididos em dois conjuntos de dados, os dados de treino que são utilizados para realizar as recomendações, e os dados de teste, que são utilizados para validar o que for recomendado.

4.3 Limpeza dos dados e criação de nova coluna

Na etapa de normalização são filtrados os dados relativos apenas aos processos de vendas de interesse deste estudo. Também são removidos valores nulos, valores incorretos, artigos, e quaisquer outras informações que possam vir a atrapalhar os resultados pretendidos neste trabalho.

Após a normalização é criada uma nova coluna chamada perfil, onde são concatenadas todas as informações do perfil do contato.

4.4 *Ranking* dos produtos mais vendidos

Criação de um *ranking* com os todos os produtos, ordenados pelo número de processos de vendas ganhos de cada produto. Esse *ranking* é utilizado para realizar as recomendações por popularidade de produtos.

4.5 Matriz de similaridade com similaridade de cosseno

A matriz de similaridade é construída utilizando a similaridade de cosseno. O cálculo é feito utilizando a lista com os contatos onde é feita uma comparação entre pares de contatos, de modo que cada contato é comparado com todos os contatos da lista, inclusive com ele mesmo. Ao final temos uma matriz da ordem de NxN onde N é a quantidade de contatos na lista. O Quadro 3 mostra um exemplo de matriz de similaridade.

Quadro 2 - Exemplo de matriz de similaridade entre contatos

	CONTATO A	CONTATO B	CONTATO C
CONTATO A	1	0,1	0,5
CONTATO B	0,1	1	0,9
CONTATO C	0,5	0,9	1

Fonte: elaborado pelo autor.

4.6 Matriz de dissimilaridade com distância de Levenshtein

O cálculo de dissimilaridade é realizado utilizando como medida a distância de *Levenshtein*. Assim como no cálculo de similaridade do passo 4.4 todos os contatos são comparados entre si em pares, e ao final temos uma matriz da ordem de NxN onde N é a quantidade de contatos na lista. O Quadro 4 mostra um exemplo de matriz de dissimilaridade.

Quadro 3 - Exemplo de matriz de dissimilaridade entre contatos

	CONTATO A	CONTATO B	CONTATO C
CONTATO A	0	3	5
CONTATO B	3	0	7
CONTATO C	5	7	0

Fonte: elaborado pelo autor.

4.7 Clusterização por k-medoids com similaridade de cosseno

Nesta etapa é realizada uma clusterização a partir da matriz de similaridade gerada utilizando a similaridade de cosseno. Os contatos são divididos em x *clusters*, agrupados de acordo com a semelhança entre eles.

4.8 Clusterização por k-medoids com distância de *Levenshtein*

Nesta etapa é realizada uma clusterização a partir da matriz de dissimilaridade gerada utilizando a distância de *Levenshtein*. Os contatos são divididos em x *clusters*, agrupados de acordo com a semelhança entre eles.

4.9 Sistemas de recomendação

Nesta etapa são feitos 5 tipos de recomendações; a recomendação por popularidade que recomenda os produtos mais vendidos a partir do *ranking* de popularidade de produtos; recomendação por filtragem colaborativa a partir dos contatos mais próximos utilizando-se a matriz de similaridade de cosseno; recomendação por filtragem colaborativa a partir dos contatos mais próximos utilizando-se a matriz de dissimilaridade com distância de *Levenshtein*; recomendação por filtragem colaborativa utilizando-se a clusterização com similaridade de cosseno; recomendação por filtragem colaborativa utilizando-se a clusterização com distância de *Levenshtein*.

5 RESULTADOS

Neste capítulo, é descrito todo o processo de execução deste trabalho, desde a obtenção dos dados até o resultado das recomendações.

5.1 Obtenção dos dados

Os dados extraídos contêm informações sobre os contatos, produtos, e sobre as oportunidades de vendas, os quais possuem os atributos apresentados no Quadro 4.

Quadro 4 - Atributos das tabelas contatos, produtos e oportunidades de vendas

Contatos	**Produtos**	**Oportunidades de venda**
Id	Id	Id
Nome	Nome	Identificador do produto
Área de atuação	Categoria	Identificador do contato
Cargo	Tipo	Status
Nível de escolaridade	Quantidade de vendas realizadas	Data
Nível hierárquico		
Tipo de instituição		

Fonte: Elaborado pelo autor.

Foram extraídos dados de 7032 contatos, 18 produtos de capacitação e 2116 oportunidades de vendas ganhas.

Os produtos possuem três diferentes tipos, capacitação, intervenção e misto. Os produtos que são vendidos para os contatos são os produtos do tipo capacitação, os produtos de intervenção são vendidos para empresas, e os produtos mistos é uma junção dos outros dois.

As oportunidades de vendas têm vários status como ganha, aberta e cancelada. Ganha significa que todo o processo de venda foi executado e que a venda foi concretizada, aberto significa que o processo de venda ainda está em andamento, e por fim cancelado significa que a venda não foi realizada por qualquer motivo.

5.2 Dados de treino e dados de teste

Os dados coletados foram analisados em dois cenários diferentes, com base no período em que as vendas foram realizadas, com o objetivo de verificar se houve mudança no padrão de consumo durante a pandemia. No cenário 1 os dados de treino são compostos pelas

oportunidades de vendas realizadas entre janeiro de 2015 e dezembro de 2018, e os dados de testes são compostos pelas oportunidades de vendas realizadas entre janeiro de 2019 e janeiro de 2021.

No cenário 2 os dados de treino são compostos pelas oportunidades de vendas realizadas entre janeiro de 2015 e junho de 2019, e os dados de testes são compostos pelas oportunidades de vendas realizadas entre julho de 2019 e janeiro de 2021.

5.3 Limpeza dos dados e criação de nova coluna

As colunas área de atuação, cargo, escolaridade, nível hierárquico e tipo de instituição passaram por um processo de limpeza dos dados, sendo removidos os valores nulos, valores incorretos, *stopwords*, também foram removidos o espaçamento entre palavras, e as mesmas foram escritas em caixa baixa. Dessa forma um cargo como "Analista de Departamento Pessoal" após passar pela normalização ficou com valor "analistadepartamentopessoal".

Na última etapa após passarem pelo processo de normalização, os valores dessas colunas foram concatenados e transformados em uma nova coluna na tabela contatos, chamada de perfil.

5.4 *Ranking* dos produtos mais vendidos

Os produtos da base de treino foram ranqueados a partir da quantidade de oportunidades de vendas ganhas de cada produto, em ordem decrescente de forma que os produtos com mais oportunidades de vendas ficam posicionados no início do *ranking*, e os produtos com menos oportunidades de vendas ficam no final do *ranking*.

5.5 Matriz de similaridade com similaridade de cosseno

Utilizando o *CountVectorizer* do *scikit-learn* foi gerada uma matriz de ocorrências que será usada para o cálculo da matriz de similaridade. A matriz de ocorrências foi construída a partir das informações presentes na coluna perfil da tabela contatos, e ao final do processamento a mesma ficou com as dimensões de 7032 x 2620, onde 7032 representam cada um dos contatos, e onde 2620 representam as palavras únicas contidas nas informações da

coluna perfil. A matriz foi preenchida marcando a quantidade de ocorrências em que cada palavra ocorre no perfil de cada contato.

A matriz de similaridade foi gerada utilizando o *scikit-learn*, que calcula o produto escalar normalizado de todos os vetores, e que recebeu como entrada a matriz de ocorrências. A Figura 5 mostra a matriz gerada.

Figura 5 - Matriz de similaridade cosseno

	Contato 1	Contato 2	Contato 3	...	Contato 4	Contato 5	Contato 6
Contato 1	1	0.1690	0.3380	...	0.5070	0.1889	0.1889
Contato 2	0.1690	1	0.6	...	0	0	0
Contato 3	0.3380	0.6	1	...	0.2	0.2236	0.2236
...	...	...	...	...	...	...	...
Contato 4	0.5070	0	0.2	...	1	0.2236	0.2236
Contato 5	0.1889	0	0.2236	...	0.2236	1	0.25
Contato 6	0.1889	0	0.2236	...	0.2236	0.25	1

Fonte: Elaborado pelo autor.

5.6 Matriz de dissimilaridade com distância de Levenshtein

A matriz de dissimilaridade utilizando a distância de *Levenshtein* foi calculada usando-se a biblioteca *Levenshtein*. Cada contato presente na lista de contatos foi comparado com todos os contatos da lista através da coluna perfil. O resultado de cada comparação foi inserido na matriz de dissimilaridade de dimensão 7032 x 7032, onde 7032 representa a quantidade de contatos presentes na lista. A Figura 6 mostra a matriz gerada.

Figura 6 - Matriz de dissimilaridade Levenshtein

	Contato 1	Contato 2	Contato 3	...	Contato 4	Contato 5	Contato 6
Contato 1	0	45	38	...	34	37	46
Contato 2	45	0	19	...	60	56	58
Contato 3	38	19	0	...	52	46	51
...	...	...	...	...	...	...	...
Contato 4	34	60	52	...	0	49	55
Contato 5	37	56	46	...	49	0	46
Contato 6	46	58	51	...	55	46	0

Fonte: Elaborado pelo autor.

5.7 Clusterização por k-medoids com similaridade de cosseno

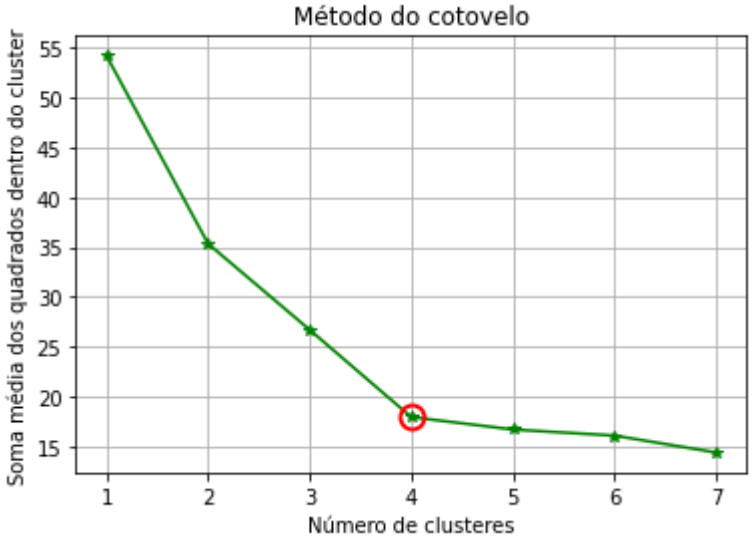
O primeiro passo para a clusterização foi definir a quantidade de clusters que irão agrupar os contatos, para isso foi utilizado o método do cotovelo e a análise da silhueta, onde foram plotados clusters com tamanho variando de 1 a 7.

5.7.1 Método do cotovelo e análise da silhueta

A análise do método do cotovelo Figura 7, mostra que o ponto é que a variação começa a achatar formando o ‘cotovelo’ no número 4, logo o número ideal segundo o método do cotovelo é de 4 *clusters*.

A análise da silhueta Figura 8, mostra que os resultados com 3 e 4 possuem valores acima do coeficiente de silhueta em todos os *clusters*, e a Tabela 2 mostra que o coeficiente de silhueta é maior para o número 4, logo pela análise da silhueta tem-se que o número ideal de *clusters* é 4.

Figura 7 - Método do cotovelo com similaridade do cosseno



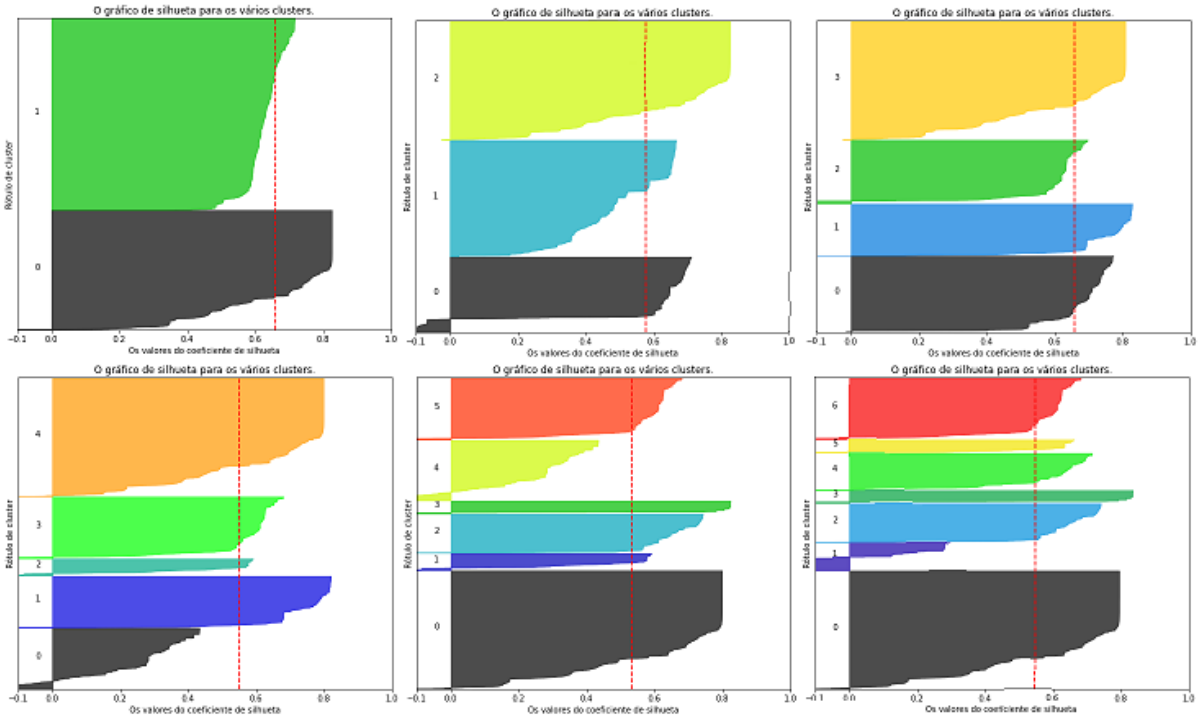
Fonte: Elaborado pelo autor.

Tabela 2 - Coeficiente de silhueta com similaridade de cosseno

Número de *Clusters*	Coeficiente de Silhueta
2	0.6573
3	0.5756
4	0.6588
5	0.5506
6	0.5330
7	0.5471

Fonte: Elaborado pelo autor.

Figura 8 - Análise de silhueta com similaridade cosseno

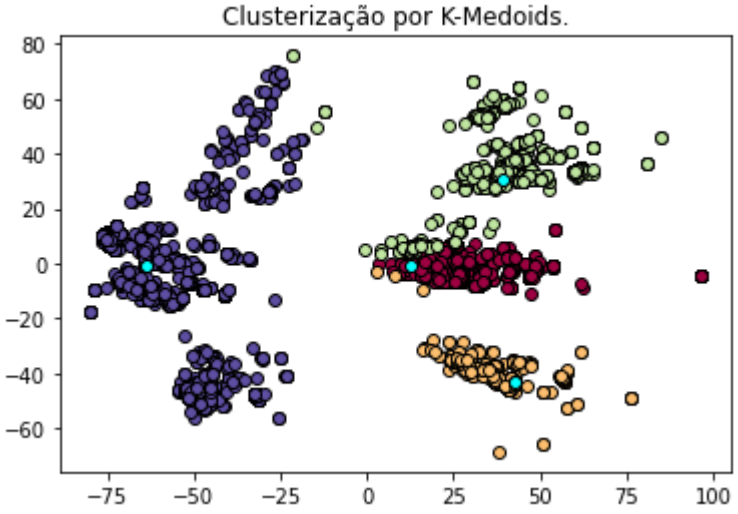


Fonte: Elaborado pelo autor.

5.7.2 Clusterização

A matriz de similaridade com similaridade de cosseno foi escalada e reduzida utilizando-se uma redução de dimensionalidade linear, *Principal Component Analysis* (PCA), ficando com as dimensões 7032 x 2. Então foi utilizando *KMedoids* do *scikit-learn*, passando como parâmetros a matriz reduzida e o número de clusters indicado pelo método do cotovelo e análise da silhueta. A Figura 9 mostra como ficaram organizados os *clusters*.

Figura 9 - Clusterização k-medoids com similaridade de cosseno



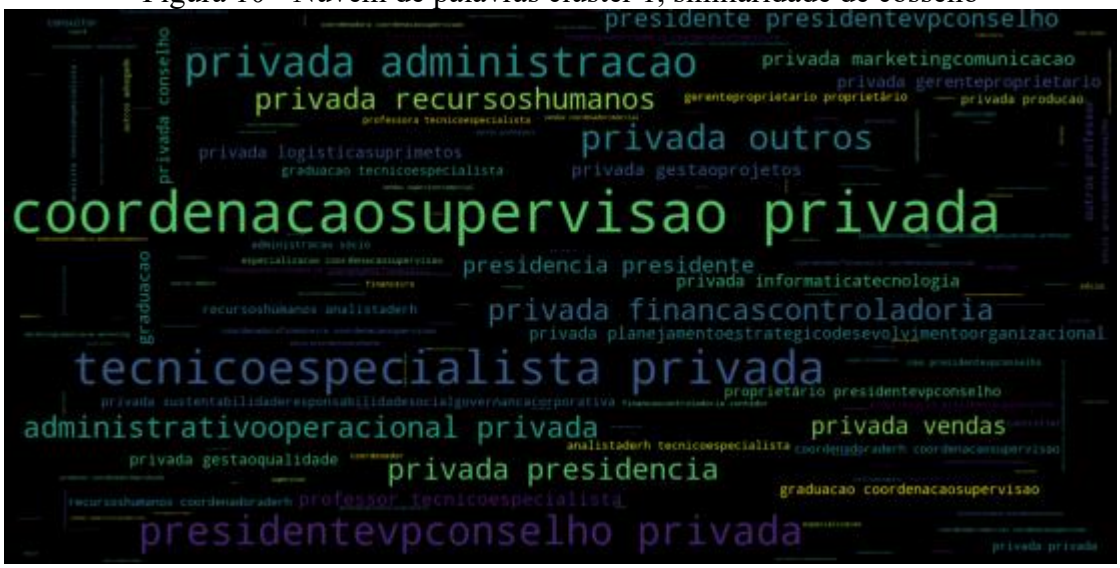
Fonte: Elaborado pelo autor.

5.7.2.1 Nuvem de palavras

Foi gerada uma visualização de nuvem de palavras para identificação dos perfis mais comuns presentes em cada um dos *clusters* gerados.

A partir das nuvens de palavras foi possível identificar que o *cluster* 1 é composto por contatos de instituições privadas, com perfis de presidência, administração e coordenação. O *cluster* 2 é composto por contatos de instituições privadas, com perfis de gestão, gerência e finanças. O *cluster* 3 é composto por contatos de instituições privadas, com perfil de diretoria. O *cluster 4* é composto por contatos de instituições públicas. As nuvens de palavras podem ser visualizadas nas Figuras 10 a 13.

Figura 10 - Nuvem de palavras cluster 1, similaridade de cosseno



Fonte: Elaborado pelo autor.

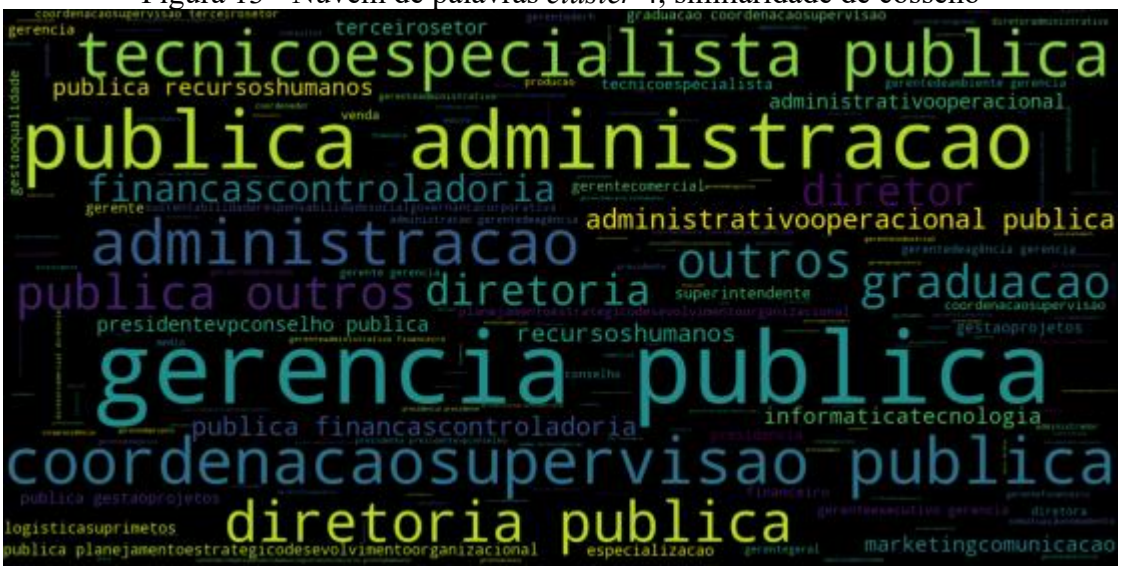
Figura 11 - Nuvem de palavras *cluster* 2, similaridade de cosseno

Fonte: Elaborado pelo autor.

Figura 12 - Nuvem de palavras *cluster* 3, similaridade de cosseno

Fonte: Elaborado pelo autor.

Figura 13 - Nuvem de palavras *cluster* 4, similaridade de cosseno



Fonte: Elaborado pelo autor.

5.8 Clusterização por k-medoids com distância de *Levenshtein*

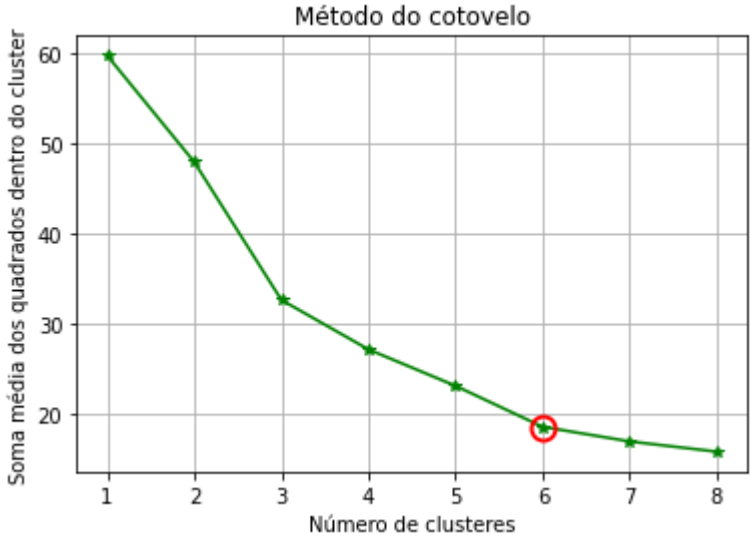
O primeiro passo para a clusterização foi definir a quantidade de clusters que irão agrupar os contatos, para isso foi utilizado o método do cotovelo e a análise da silhueta, onde foram plotados clusters com tamanho variando de 3 a 8.

5.8.1 Método do cotovelo e análise da silhueta

A análise do método do cotovelo Figura 14, mostra que o ponto é que a variação começa a achatar formando o ‘cotovelo’ no número 6, logo o número ideal segundo o método do cotovelo é de 6 *clusters*.

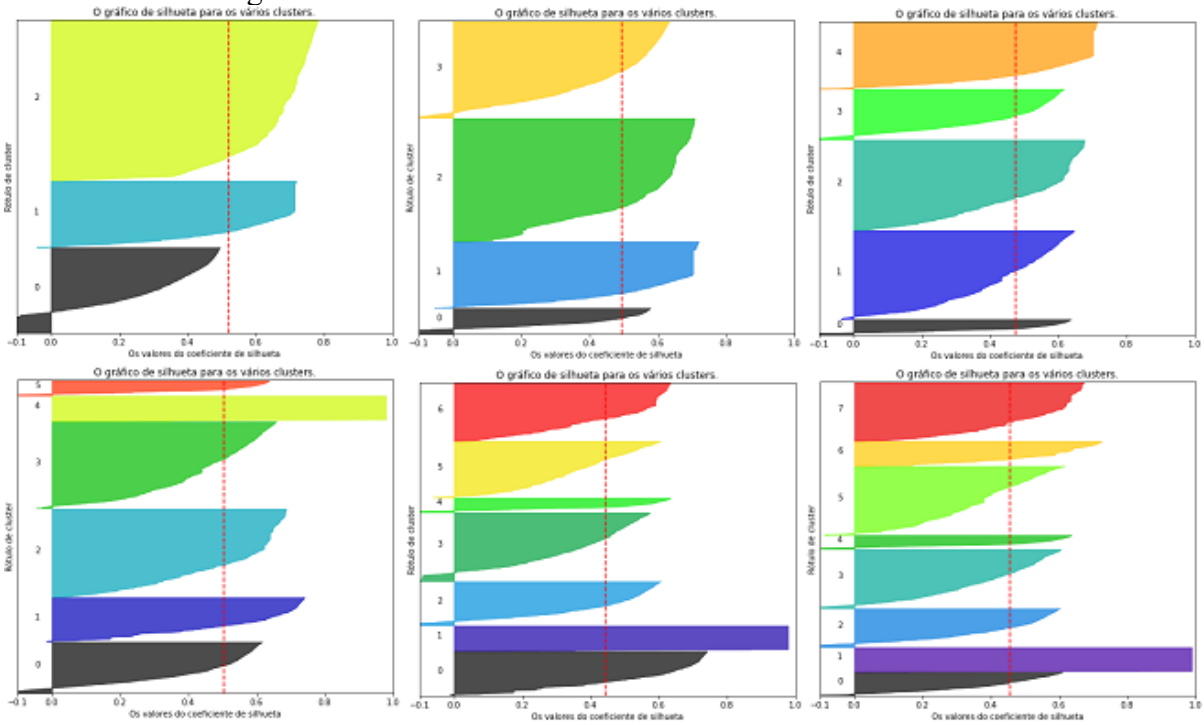
A análise da silhueta Figura 15, mostra que os resultados com 6 a 8 possuem valores acima do coeficiente de silhueta em todos os *clusters*, e a Tabela 3 mostra que o coeficiente de silhueta é maior para o número 6, logo pela análise da silhueta tem-se que o número ideal de *clusters* é 6.

Figura 14 - Método do cotovelo com distância de *Levenshtein*



Fonte: Elaborado pelo autor.

Figura 15 - Análise de silhueta com distância de Levenshtein



Fonte: Elaborado pelo autor.

Tabela 3 - Coeficiente de silhueta com distância de *Levenshtein*

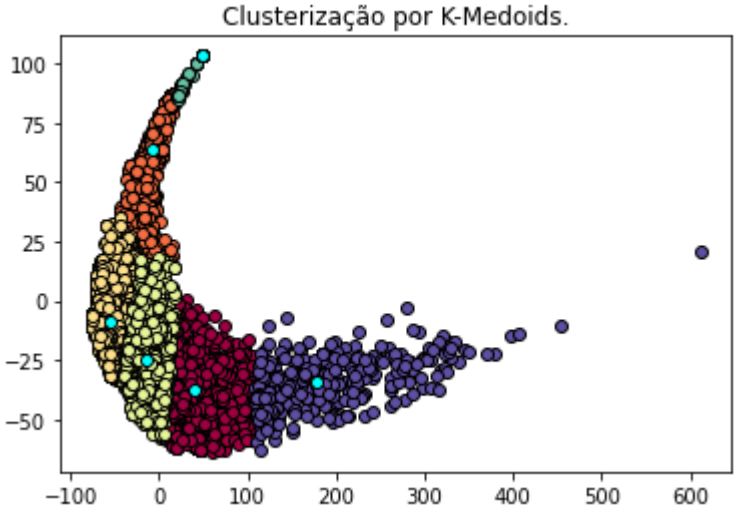
Número de *Clusters*	Coeficiente de Silhueta
3	0.5199
4	0.4955
5	0.4754
6	0.5045
7	0.4443
8	0.4554

Fonte: Elaborado pelo autor.

5.8.2 Clusterização

A matriz de dissimilaridade com distância de *Levenshtein* foi escalada e reduzida utilizando-se uma redução de dimensionalidade linear, *Principal Component Analysis* (PCA), ficando com as dimensões 7032 x 2. Então foi utilizando *KMedoids* do *scikit-learn*, passando como parâmetros a matriz reduzida e o número de clusters indicado pelo método do cotovelo e análise da silhueta. A Figura 16 mostra como ficaram organizados os *clusters.*

Figura 16 - Clusterização k-medoids com distância de Levenshtein



Fonte: Elaborado pelo autor.

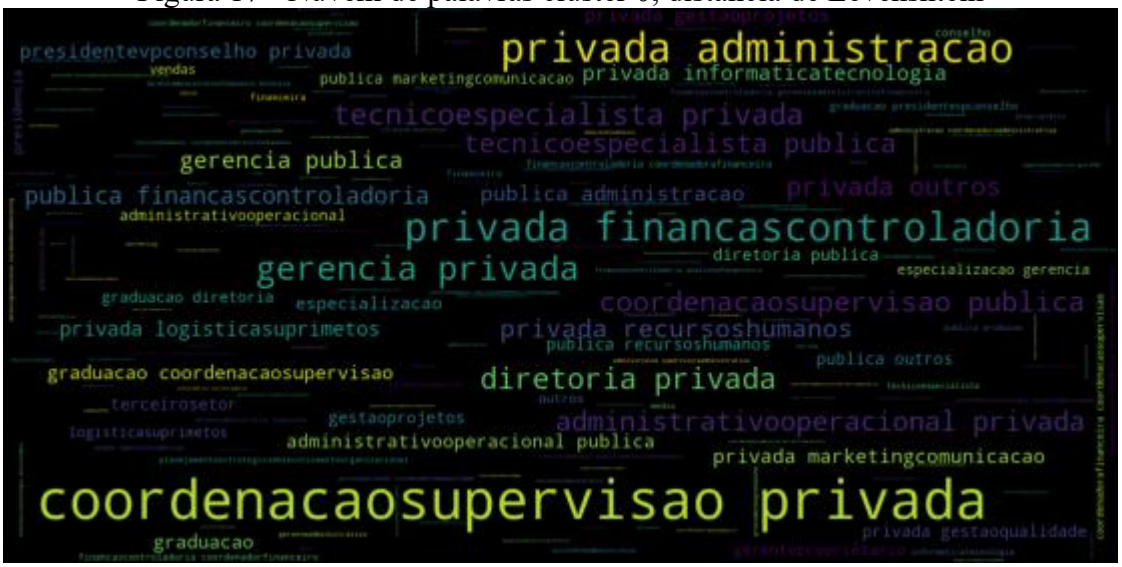
5.8.3 Nuvem de palavras

Foi gerada uma visualização de nuvem de palavras para identificação dos perfis mais comuns presentes em cada um dos clusters gerados.

Na clusterização utilizando a distância de *Levenshtein* foram gerados 6 *clusters*, e para cada *cluster* foi gerado uma nuvem de palavras. A partir das nuvens de palavras foi possível

identificar que o *cluster* 1 é composto por contatos com perfis de coordenação, supervisão. O *cluster* 2 é composto por contatos de instituições privadas, com perfis de diretor, gerente e que possuem graduação. O *cluster* 3 é composto por contatos de instituições privadas, com perfis de diretoria, administração e gerência. O *cluster* 4 é composto por contatos de instituições privadas, com perfis de gerência, presidente, diretoria e administração. Os *cluster* 5 e 6 são é composto por contatos que possuem poucas informações de perfil, sendo que o 5 possui contatos de instituições públicas. As nuvens de palavras podem ser visualizadas nas Figuras 17 a 22.

Figura 17 - Nuvem de palavras cluster 0, distância de Levenshtein



Fonte: Elaborado pelo autor.

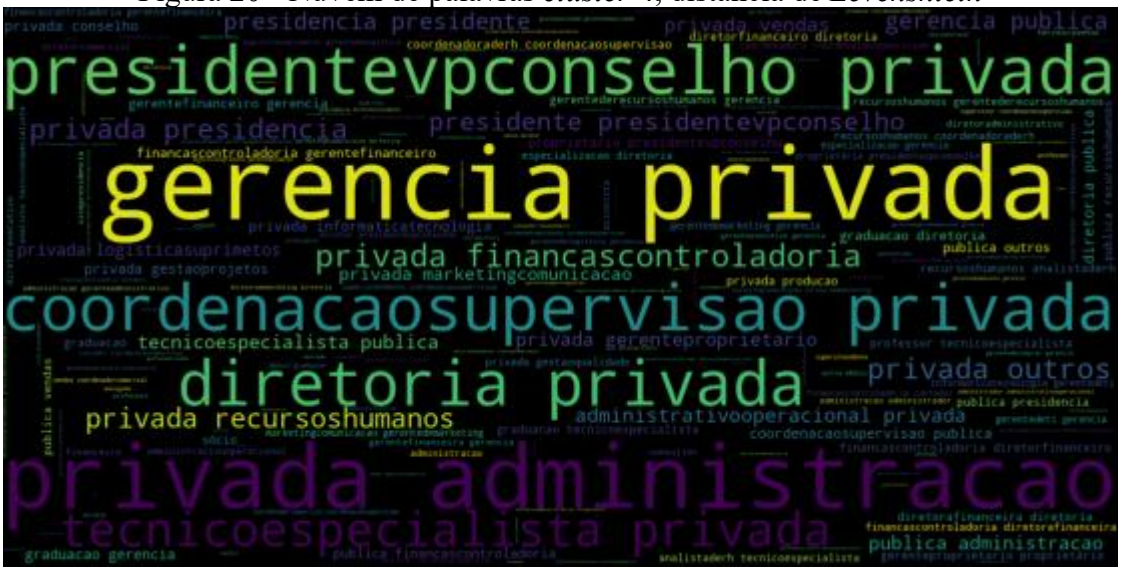
Figura 18- Nuvem de palavras *cluster* 1, distância de *Levenshtein*

Fonte: Elaborado pelo autor.

Figura 19 - Nuvem de palavras *cluster* 3, distância de *Levenshtein*

Fonte: Elaborado pelo autor.

Figura 20 - Nuvem de palavras *cluster* 4, distância de *Levenshtein*



Fonte: Elaborado pelo autor.

Figura 21 - Nuvem de palavras *cluster* 5, distância de *Levenshtein*

Fonte: Elaborado pelo autor.

Figura 22 - Nuvem de palavras *cluster* 6, distância de *Levenshtein*

Fonte: Elaborado pelo autor.

5.9 Sistemas de recomendação

Nesta seção será descrito como foram realizados os 5 tipos de recomendações de produtos.

5.9.1 Recomendação por popularidade de produtos

A recomendação por popularidade foi feita a partir do *ranking* de produtos gerados com a base de treino, os produtos recomendados foram aqueles mais vendidos de acordo com

o *ranking*. A quantidade de produtos recomendados corresponde a 20% do total de produtos presentes na lista.

5.9.2 Recomendação por filtragem colaborativa com similaridade de cosseno

A partir da matriz de similaridade gerada usando a similaridade cosseno, foi criada uma lista para cada contato, contendo os 100 contatos com perfis mais próximos. Os produtos recomendados foram aqueles produtos já adquiridos na base de treino pelos contatos próximos.

5.9.3 Recomendação por filtragem colaborativa com distância de Levenshtein

A partir da matriz de dissimilaridade gerada usando a distância de *Levenshtein*, foi criada uma lista para cada contato, contendo os 100 contatos com perfis mais próximos. Os produtos recomendados foram aqueles produtos já adquiridos na base de treino pelos contatos próximos.

5.9.4 Recomendação por filtragem colaborativa com clusterização

Para cada cluster¸ os produtos recomendados foram os produtos já adquiridos na base de treino pelos contatos que são membros do cluster.

5.10 Resultados obtidos

Após a realização das etapas anteriores, foi-se utilizada a base de testes para validar as recomendações realizadas com a base de treino para os diferentes métodos de recomendação, as oportunidades de vendas ganhas da base de testes foram comparadas com as recomendações feitas com a base de treino, dessa forma obteve-se os seguintes resultados apresentados no Quadro 5 e no Quadro 6, que representam os resultados para o cenário 1 e cenário 2 respectivamente.

Quadro 5 - Comparativo de acertos de recomendações cenário 1

	Popularidade	Distância de *Levenshtein*	Similaridade de Cosseno	*K-Medoids com distância de Levenshtein*	*K-Medoids com similaridade de* Cosseno
Acertos em %	64,75%	52,44%	62,7%	76,27%	76,85%

Fonte: Elaborado pelo autor.

Quadro 6 - Comparativo de acertos de recomendações cenário 2

	Popularidade	Distância de *Levenshtein*	Similaridade de Cosseno	*K-Medoids com distância de Levenshtein*	*K-Medoids com similaridade de* Cosseno
Acertos em %	44,94%	56,47%	60,23%	63,06%	63,76%

Fonte: Elaborado pelo autor.

Como apresentado nos Quadros 5 e 6, os métodos que se utilizaram de clusterização para realização das recomendações obtiveram melhores resultados em ambos os cenários analisados neste trabalho, sendo que as clusterizações realizadas com a similaridade de cosseno tiveram uma leve vantagem sobre as clusterizações realizadas com a distância de Levenshtein.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi descrito o processo de criação de um sistema de recomendação para a venda de produtos para contatos em um sistema de CRM. Foram utilizadas duas abordagens para a geração das recomendações, a primeira recomendava produtos com base em sua popularidade, ou seja, os produtos com um maior número de vendas realizadas eram indicados para todos os contatos.

Na segunda abordagem se utilizam de recomendação baseada em colaboração, ou seja, os contatos teriam uma maior chance de comprar produtos que foram comprados por outros contatos que tenham um perfil semelhante ao seu. Para a recomendação baseada em colaboração foram utilizadas como métricas a similaridade de cosseno, e a distância de *Levenshtein*, e também foram realizadas clusterizações utilizando-se dessas métricas.

Nos resultados observou-se que os métodos de clusterização por *k-medoids* obtiveram uma melhor performance nos dados de testes em ambos os cenários, com uma leve vantagem para a clusterização que utilizou como métrica a similaridade de cosseno.

Como sugestão de trabalhos futuros pode-se considerar o uso de outras métricas para o sistema de recomendação, como distância euclidiana. Outro possível trabalho futuro seria implementar outro tipo de recomendação, como recomendação baseada em conteúdo, ou um modelo híbrido, que faça uso de dois ou mais modelos de recomendações.

REFERÊNCIAS

ALBUQUERQUE, M. A. **Estabilidade em análise de agrupamento (cluster analysis)**. 2005. 64 f. Dissertação (Programa de Pós-Graduação em Biometria e Estatística Aplicada) - Universidade Federal Rural de Pernambuco, Recife, 2013.

BRUNIALTI, L.; PERES, S.; FREIRE, V.; LIMA, C. Aprendizado de Máquina em Sistemas de Recomendação Baseados em Conteúdo Textual: Uma Revisão Sistemática. *In*: SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO (SBSI), 11., 2015, Goiânia. **Anais [...]** Porto Alegre: Sociedade Brasileira de Computação, 2015. p. 203-210.

BURKE, R. *Hybrid Web Recommender Systems*. In: Brusilovsky P.; Kobsa A.; Nejdl W. (eds). ***The Adaptive Web***. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg, 2007. p. 377-408

GOTARDO, R. **Uma abordagem de sistema de recomendação orientada pelo aprendizado sem fim**. 2014. 105 f. Tese (Doutorado em Ciências da Computação) - Universidade Federal de São Carlos, São Carlos, 2014.

HALDAR, R.; MUKHOPADHYAY, D. *Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach.* **Computing Research Repository - CORR**. 2011.

HAN, J.; KAMBER, M.; PEI, J. ***Data Mining***. 3 ed. Morgan Kaufmann, 2012.

HENRIQUE, S. S. **Desenvolvimento de um sistema capaz de recomendar rotas seguras para ciclistas**. 2019. 38 p. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) - Universidade Federal do Ceará, Campus de Quixadá, Quixadá, 2019.

KAUFMAN, L.; ROUSSEEUW, P. *Clustering by Means of Medoids*. In: Dodge, Y. (ed.) ***Data Analysis based on the L1-Norm and Related Methods***. North Holland / Elsevier, 1987. p. 405-416.

KAUFMAN, L.; ROUSSEEUW, P. ***Finding Groups in Data: An Introduction to Cluster Analysis***. Hoboken, New Jersey: John Wiley & Sons, Inc., 1990.

LEITE, P. **Sistema de recomendação aplicado em plataformas de reserva online de restaurante**. 2019. 64 f. Monografia (Bacharelado em Ciências da Computação) - Centro Universitário Senac - Santo Amaro. São Paulo, 2019.

MACKENZIE, I.; MEYER, C.; NOBLE, S. *How retailers can keep up with consumers*. **McKinsey & Company**, 16 out. 2013. Disponível em: https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumersa. Acesso em: 08 mar. 2021.

MITCHELL, T. M. ***Machine Learning***. [S.l.]: McGraw-Hill, 1997. 414 p.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. *In*: REZENDE, S. O. **Sistemas inteligentes:** fundamentos e aplicações. Editora Manole Ltda, 2003. p. 89-114.

MONI, R. *Reinforcement Learning algorithms - an intuitive overview*. **Medium**, 18 fev. 2019. Disponível em: https://medium.com/@SmartLabAI/reinforcement-learning-algorithms-an-intuitive-overview-904e2dff5bbc. Acesso em 22 fev. 2021.

NASTESKI, V. ***An overview of the supervised machine learning methods***. 4 ed. Horizons B., 2017. p. 51-62.

RESNICK, P.; VARIAN, H. R. ***Recommender Systems***. v. 40. New York, NY. Association for Computing Machinery, 1997. p. 55-58.

REZENDE, S. O. **Sistemas inteligentes:** fundamentos e aplicações. Barueri, São Paulo: Editora Manole Ltda, 2003.

RICCI, F.; ROKACH, L.; SHAPIRA, B. *Introduction to recommender systems handbook. In:* ***Recommender systems handbook***. [S.l.]: Springer, 2011. p. 1–35.

ROUSSEEUW, P. *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis***. Journal of Computational and Applied Mathematics**, v. 20. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1987. p. 53–65.

RUSSELL, S. J.; NORVIG, P. ***Artificial intelligence:*** *a modern approach.* Third Edition. [S.l.]: Prentice Hall, 2010.

SCHUBERT, E.; ROUSSEEUW, P. *Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms*. IN: ***Similarity Search and Applications***. Cham: Springer International Publishing, 2019. p. 171-187.

SUBRAMANIAN, D. *Building a Content-Based Book Recommendation Engine.* **Towards Data Science**, 16 mar. 2020. Disponível em: https://towardsdatascience.com/building-a-content-based-book-recommendation-engine-9fd4d57a4da. Acesso em: 17 fev. 2021.

TEIXEIRA, D. S. **Preenchimento de playlists utilizando técnicas de sistemas de recomendação baseadas em filtragem colaborativa**. 2018. 35 f. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) - Universidade Federal do Ceará, Campus de Quixadá, Quixadá, 2018.

THOMPSON, C. *If you liked this, you're sure to love that*. **The New York Times**, 21 nov. 2008. Disponível em: https://www.nytimes.com/2008/11/23/magazine/23Netflix-t.html. Acesso em: 17 fev. 2021.

THORNDIKE, R. L. ***Who belong in the family? Psychometrika***. v. 18. 1953. p. 267-276,