CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS
Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U nº 198, de 14/10/2016
ASSOCIAÇÃO EDUCACIONAL LUTERANA DO BRASIL

Pedro Henrique Gomes Camargo

SENTIMENTALL: módulo de análise de agrupamentos e visualização de informação aplicado a avaliações do contexto do turismo nacional

Palmas – TO
2016

Pedro Henrique Gomes Camargo

SENTIMENTALL: módulo de análise de agrupamentos e visualização de informação aplicado a avaliações do contexto do turismo nacional

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.e Jackson Gomes de Souza.

Palmas – TO

2016

Pedro Henrique Gomes Camargo

SENTIMENTALL: módulo de análise de agrupamentos e visualização de informação aplicado a avaliações do contexto do turismo nacional

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.e Jackson Gomes de Souza.

Aprovado em: _____/_____/_____

BANCA EXAMINADORA

_____

Prof. M.e Jackson Gomes de Souza

Orientador

Centro Universitário Luterano de Palmas – CEULP

_____

Prof. Dr. Edeilson Milhomem da Silva

Centro Universitário Luterano de Palmas – CEULP

_____

Prof. M.e Parcilene Fernandes de Brito

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2016

“A melhor maneira de prever o futuro é criá-lo. ”
(Peter Drucker)

RESUMO

CAMARGO, Pedro Henrique Gomes. **SENTIMENTALL: módulo de análise de agrupamentos e visualização de informação aplicado a avaliações do contexto do turismo nacional**. 2016. 84 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Sistemas de Informação, Centro Universitário Luterano de Palmas, Palmas/TO, 2016[2].

O projeto do SentimentALL é um protótipo de ferramenta com módulos de extração de dados, mineração de dados e visualização de informação. Conduzidos por grupos de pesquisa do CEULP/ULBRA e da PUC-GO (BRITO *et al.*, 2015), os seus trabalhos se concentram no contexto do turismo nacional e têm como um dos seus principais módulos o *Módulo de Análise de Sentimentos*, que realiza o processo de análise de sentimentos utilizando a abordagem a nível de aspectos aplicadas em avaliações extraídas do site TripAdvisor. As avaliações se tratam de comentários reportados por usuários do site sobre um dado objeto (hotel, restaurante, clube, etc.) na qual são mencionadas características deste objeto – os chamados *aspectos* –, como o atendimento, preço, comida, entre outros. O resultado desta análise de sentimentos foi utilizado como entrada para o presente trabalho, na qual se objetiva o desenvolvimento de um módulo que aplica a análise de agrupamentos e interprete a saída de tal modo que seja possível identificar o quanto cada aspecto *contribui para a formação do seu agrupamento*, isto é, o quão um aspecto é estatisticamente relevante para o seu grupo. O *Módulo de Análise de Agrupamentos* é uma solução web que permite o gerenciamento e visualização em gráficos dos agrupamentos formados. Cenários de teste também foram criados para demonstrar a solução em um contexto real, utilizando para isso, os dados extraídos por Christhie (2015) no *Módulo de Extração de Dados* do SentimentALL.

Palavras-chave: Mineração de Dados. Análise de Agrupamentos. Modelo de Tópicos.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

BDMS	Database Management System
BOW	Bag-of-Words
CEULP	Centro Universitário Luterano de Palmas
CSV	Comma-separated values
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
IDE	Integrated Development Environment
KDD	Knowledge-discovery in Databases
LDA	Latent Dirichlet Allocation
PLN	Processamento de Linguagem Natural
SDQ	Soma das distâncias ao quadrado
TF-IDF	Term Frequency - Inverse Document Frequency
ULBRA	Universidade Luterana do Brasil
SGDB	Sistema de Gerenciamento de Banco de Dados

LISTA DE APÊNDICES

SUMÁRIO

1 INTRODUÇÃO

É bem conhecido que a quantidade de informações na internet é vasta e a sua expansão, há bastante tempo, segue em alto ritmo e sem previsão de ser frenada. A popular expressão de um mundo que vive a *era da informação* poderia ser indiscutivelmente aceita nos dias de hoje, no entanto, como sugerem Han, Kamber e Pei (2011), na verdade o mundo vivencia a era dos dados, a *era rica em dados, mas pobre em informação*.

Há 20 anos, Fayyad, Piatetsky-Shapiro e Smyth (1996) já ressaltavam a urgente necessidade uma nova geração de teorias e ferramentas computacionais para ajudar os seres humanos a extrair conhecimento dos volumes de dados que crescem rapidamente. Este crescimento representa 3,3 bilhões de usuários (46% da população global) (STATS, 2016; DOMO, 2015) com uma capacidade tecnológica per-capita que permite praticamente dobrar a quantidade de informações na internet a cada 40 meses (desde de 1980) (HILBERT; LÓPEZ, 2011), bem como, desde 2002, todos os dias 2,5 bilhões de *gigabytes* de dados são criados (IBM, 2016).

Todas essas estatísticas evidenciam um cenário cada vez mais expressivo: embora a capacidade de capturar e armazenar grandes quantidades de dados vem crescendo em ritmo sem precedentes, as tecnologias para recuperar, analisar e obter *conhecimento novo* acerca de determinada temática está muito aquém disso.

Exemplos do que se entende por obter conhecimento novo está na antecipação do desejo de compra de um cliente a partir de suas características, na segmentação de clientes em grupos de alta similaridade ou ainda na sumarização de opiniões de usuários sobre determinado assunto.

Sobre este último, aliás, é muito difícil encontrar pessoas que nunca procuraram informações na internet sobre um produto ou serviço antes de realizarem uma compra. Pesquisar a opinião, avaliação e conhecer a experiência de outras pessoas sobre um bem ou serviço em particular é, sem dúvidas, uma prática de uso geral e pode influenciar diretamente no processo de tomada de decisão (VERMA; PATEL; PATEL, 2013).

Este cenário motivou o campo de estudo que ficou conhecido como análise de sentimentos ou mineração de opiniões (LIU, 2012). A mineração de opiniões se preocupa com os sentimentos presentes em opiniões, atitudes, emoções, entre outras formas de se expressar.

Uma abordagem da análise de sentimentos é a dita *abordagem em nível de aspectos*, que concerne ao reconhecimento de expressões de sentimento dentro de um documento e os aspectos que eles se referem (FELDMAN, 2013). Por exemplo, para a sentença “*o hotel tem*

bom preço e atendimento, mas o café da manhã deixa a desejar", a seguinte análise pode ser obtida:

1. O **objeto** em questão é *hotel*;
2. Os **aspectos** referentes a este objeto são *preço*, *atendimento* e *café da manhã*; e
3. A **polaridade** – que tem por objetivo definir os aspectos como positivos, negativos ou neutros –, caracterizam *preço* e *atendimento* como aspectos de polaridade positiva e o aspecto *café da manhã*, como sendo de polaridade negativa. Tal percepção pode ser explicada por meio da presença das palavras (ou expressões) opinativas:
    - *Bom* para os aspectos *preço* e *atendimento;* e
    - *Deixou a desejar* para o aspecto *café da manhã*.

O resultado do processo de análise de sentimentos pode ainda ser utilizado em várias outras tarefas de mineração de dados. Por exemplo, de posse de um conjunto de aspectos extraídos de avaliações, quais deles têm a tendência de aparecerem juntos em avaliações? Ou ainda, para um grupo de aspectos similares, existe alguma maneira de determinar que um aspecto é mais importante que outro dentro um mesmo grupo?

Os questionamentos citados acima, na verdade, elucidam problemas que contemplam o projeto multidisciplinar por trás do presente trabalho. O projeto – intitulado SentimentALL – é uma parceria do CEULP/ULBRA e da PUC-GO, onde é desenvolvido pesquisas utilizando técnicas computacionais de mineração de dados e procedimentos psicológicos de análise comportamental aplicados ao contexto do turismo nacional (Brito *et al*., 2015).

Considerando o turismo nacional como sendo o principal contexto de interesse da pesquisa, os objetos de estudo se concentram em avaliações de usuários de destinos turísticos do site TripAdvisor[1]. Cada uma destas avaliações se referem a opinião que o cliente de determinado destino turístico registrou no site, fazendo referência a, por exemplo, hotéis, restaurantes, um estádio de futebol, um clube aquático, etc.

Diversos autores foram responsáveis por diferentes trabalhos desenvolvidos no projeto e o resultado de muitos deles constituíram a entrada de outros. Alguns dos trabalhos estão listados a seguir:

[1]O TripAdvisor é o maior site de viagens do mundo que, juntamente com as suas subsidiárias, formam a maior comunidade de viagens da internet, com 350 milhões de visitantes por mês, 350 milhões de avaliações e opiniões, além de cobrirem mais de 6,2 milhões de acomodações, restaurantes e atrações (TRIPADVISOR BRASIL, 2015). Mais informações em: www.tripadvisor.com.br.

A. Christhie (2015): *spider* de extração de dados do TripAdvisor utilizando técnicas de *web crawling* e *web scraping*;
B. Christhie (2015): protótipo de ferramenta de análise de sentimentos em nível de aspectos aplicado nas avaliações extraídas em A;
C. Schmitz (2015): aplicação da técnica supervisionada de regras de associação para extrair relações entre os aspectos positivos mais frequentes da análise realizada em B;
D. Roese (2016): ferramenta de visualização de informação dos dados obtidos em A e B; e
E. Araújo (2016): ferramenta para extração de regras de associação de aspectos positivos e negativos obtidos em B utilizando o algoritmo *Apriori*.

Apesar dos trabalhos possuírem uma finalidade específica, todos eles lidam com o mesmo cenário. Pensando nisso, uma das propostas do SentimentALL é agregá-los (juntamente como este trabalho) em uma ferramenta única, porém, que no momento ainda se trata de um protótipo.

Considerando todo esse contexto, uma nova necessidade levantada pelo grupo de pesquisa culminou na busca pela resposta do seguinte problema: como gerar agrupamentos de avaliações de usuários do TripAdvisor *baseadas nos aspectos* e como explicar a *contribuição de cada aspecto* nos agrupamentos formados?

As expressões *avaliações baseadas nos aspectos* e a *contribuição dos aspectos nos agrupamentos formados* se referem, respetivamente, ao conjunto de aspectos identificados em cada avaliação e ao cálculo ponderado que mensura o peso dos aspectos em um dado agrupamento. Estes conceitos são imprescindíveis para a compreensão do trabalho, mas só serão detalhados nas seções posteriores.

As hipóteses levantadas consideravam a possibilidade de se gerar agrupamentos de avaliações com base nos aspectos utilizando algoritmos de análise de agrupamentos e modelagem probabilísticas de tópicos, bem como se era possível explicar a contribuição de cada aspecto na formação dos grupos.

Como objetivo geral, teve-se o desenvolvimento do módulo para o SentimentALL que aplica a análise de agrupamentos em avaliações baseadas nos aspectos e apresente os agrupamentos de tal modo que torne evidente o quanto os aspectos contribuem para a formação de um grupo. Este módulo ainda teve os seguintes objetivos específicos:

- Aplicação dos algoritmos K-Means e DBSCAN (análise de agrupamentos) e LDA (modelagem probabilística de tópicos) em um conjunto de avaliações de usuários do TripAdvisor;
- Idealização da interpretação do conceito de *contribuição dos aspectos nos agrupamentos formados*;
- Desenvolvimento de uma ferramenta web que permite a visualização dos agrupamentos e demonstra a contribuição dos aspectos em cada agrupamento por nível de estado, cidade e objeto (detalhados adiante); e
- Avaliação dos agrupamentos formados utilizando o coeficiente de silhueta.

A solução desenvolvida foi baseada no processo de *Knowledge Discovery in Databases* (KDD) proposto por Fayyad, Piatetsky-Shapiro e Smyth (1996) e se preocupa principalmente com a etapa de mineração de dados, onde são aplicados os algoritmos. Ainda em relação ao processo de KDD adaptado, algumas etapas foram removidas e outras inseridas, de acordo com a necessidade do processo. Essas alterações estão melhor descritas na seção 3.2.

De maneira geral, a grande relevância do trabalho está na capacidade de visualizar quais aspectos possuem maior grau de similaridade e o quanto cada um deles contribui para a formação dos grupos, além dos agrupamentos propriamente ditos. Por meio desta medida será possível identificar quais aspectos são mais correlatos e, principalmente, quais são os aspectos mais importantes dentro de um grupo ou contexto.

Esta monografia está estruturada da seguinte maneira: a seção 2 apresenta uma visão geral de mineração de dados, partindo desde o clássico processo de KDD até as tarefas de aprendizagem de máquina mais conhecidas e a apresentação do algoritmo LDA, que é tido como o algoritmo estado da arte em modelagem probabilística de tópicos.

A seção 3 descreve os materiais e métodos utilizado no desenvolvimento do trabalho. A seção 4 apresenta os principais artefatos desenvolvidos no módulo e discussões levantadas a partir de alguns resultados da análise e da avaliação da qualidade dos grupos formados. A seção 5 disserta sobre algumas conclusões geradas, limitações e sugestões para trabalhos futuros.

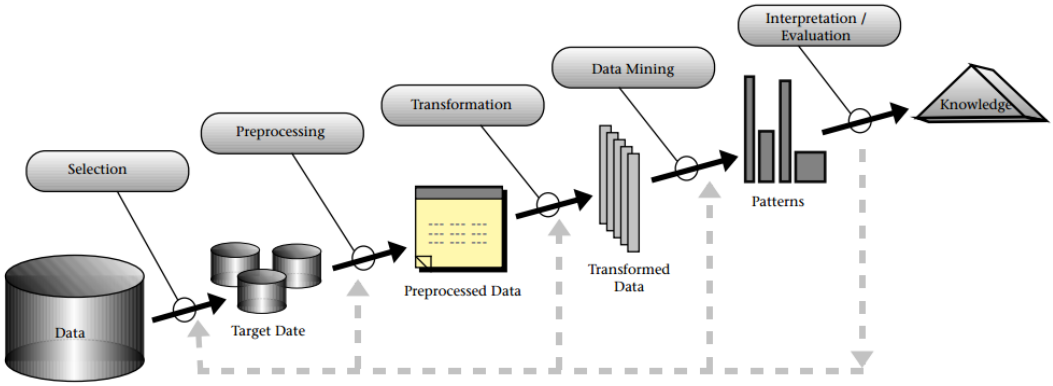
2 REFERENCIAL TEÓRICO

2.1 DESCOBERTA DO CONHECIMENTO EM BASES DE DADOS

A Descoberta de Conhecimento em Bases de Dados, do inglês *Knowledge Discovery in Databases* (KDD), é um processo não trivial de identificar informações válidas, potencialmente úteis e, finalmente, padrões compreensíveis em dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Em outras palavras, o processo de KDD é um campo que está preocupado com o desenvolvimento de métodos e técnicas para extrair conhecimento útil em dados e dar sentido a eles. Questões relacionadas incluem a coleta de dados, o desenho do banco de dados e a descrição das entradas utilizando a representação mais adequada (QIN; NORTON, 1999).

Ainda segundo Fayyad, Piatetsky-Shapiro e Smyth (1996), o processo de descoberta de conhecimento, ilustrado pela Figura 1, é iterativo, interativo (envolve muitas decisões feitas pelo usuário) e possui etapas bem definidas.

Figura 1 – Visão geral das fases do processo de KDD.



Fonte: Fayyad, Piatetsky-Shapiro e Smyth (1996).

As etapas mencionadas por Fayyad, Piatetsky-Shapiro e Smyth (1996) são as que se seguem:

1. **Aprender o domínio da aplicação**: inclui os objetivos a serem alcançados e o levantamento do conhecimento prévio relevante ao contexto de aplicação. Por exemplo, entender os conceitos básicos de análise de sentimentos e análise de agrupamentos e a sua relevância para o especialista de domínio.
2. **Criar um conjunto de dados destino**: compreende a seleção de um conjunto de dados ou concentra-se em um subconjunto de variáveis ou amostras em que a descoberta deverá ser realizada. Por exemplo, a extração das avaliações dos usuários do TripAdvisor por Christhie (2015).

3. **Limpeza de dados e pré-processamento**: inclui operações básicas como, por exemplo, a remoção de ruídos ou *outliers* (se conveniente). Aqui ainda devem ser decididas quais as estratégias serão utilizadas para lidar com as questões de BDMS (*Database Management System*) como tipos de dados, *schemas* e mapeamento de valores em falta ou desconhecidos.
4. **Redução e projeto de dados**: preocupa-se em encontrar características úteis para representar os dados.
5. **Escolhendo a função de exploração de dados**: compreende a conformidade dos objetivos traçados (etapa 1) com o método de mineração de dados utilizado, isto é, define a técnica (sumarização, agrupamento, classificação, regressão, etc.) que melhor se adeque aos objetivos.
6. **Escolhendo o(s) algoritmo(s) de mineração de dados**: define o(s) modelo(s), método(s), algoritmo(s) e parâmetro(s) de mineração de dados e que serão utilizados para encontrar padrões.
7. **A mineração de dados**: abrange a busca de padrões de interesse, utilizando alguma forma particular de representação de conhecimento ou um conjunto delas como, por exemplo, regras ou árvores de classificação, regressão e agrupamentos.
8. **Interpretação**: inclui a interpretação dos padrões descobertos que, eventualmente, pode provocar a volta para qualquer um dos passos anteriores. Permite também a visualização dos padrões extraídos, a remoção dos dados irrelevantes ou redundantes com o objetivo de, finalmente, traduzir os padrões úteis em termos compreensíveis pelos usuários.
9. **Usando o conhecimento que surgiu**: contempla a incorporação do conhecimento adquirido em outros sistemas ou simplesmente a documentação e relato às partes interessadas. Além da verificação e levantamento de possíveis conflitos na crença (conhecimento) anteriormente acreditada ou extraída com o resultado obtido.

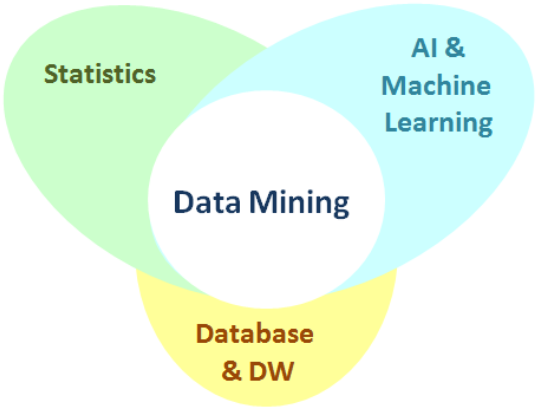
Fayyad, Piatetsky-Shapiro e Smyth (1996) apontam ainda que todas as etapas são igualmente importantes para o sucesso da aplicação do KDD na prática, entretanto, a maioria dos trabalhos focam na etapa de mineração de dados. Do mesmo modo, o presente estudo também se preocupa principalmente com a técnica não supervisionada de análise de

agrupamentos em mineração de dados e de modelo de tópicos. A mineração de dados e suas competências são assuntos da seção a seguir.

2.2 MINERAÇÃO DE DADOS

A Mineração de Dados (ou *Data Mining*, em inglês) consiste em explicar o passado e prever o futuro por meio de análise de dados e se trata de um campo multidisciplinar que combina estatística, aprendizagem de máquina, inteligência artificial, banco de dados e *data warehouse* (SAYAD, 2016), conforme ilustrado pela Figura 2.

Figura 2 – Visão geral das áreas que compõem a mineração de dados.



Fonte: Sayad (2016).

Em seu trabalho, Fayyad, Piatetsky-Shapiro e Smyth (1996) também destacou a relação entre a mineração de dados e as áreas de estatística, aprendizagem de máquina e banco de dados, bem como reforçaram que a mineração de dados é uma particular etapa no processo de KDD. Já Han, Kamber e Pei (2011), indicam que a mineração de dados é outra denominação do que é popularmente conhecido como processo de KDD e o define como sendo a extração automatizada ou conveniente de padrões que representam o conhecimento implicitamente armazenado ou capturados em grandes bases de dados, *data warehouses*, na web e em outros repositórios de informação em massa ou fluxos de dados. Jin e Lin (2011), por sua vez, conceituam a mineração de dados como uma metodologia de encontrar correlações e padrões entre dezenas de campos em grandes bancos de dados e destacam que a informação pode ser convertida em conhecimento sobre padrões históricos e tendências futuras.

A maioria dos métodos de mineração de dados são baseados em técnicas experimentadas e testadas de aprendizagem de máquina, reconhecimento de padrões e estatística. Os métodos mais comuns são a classificação, agrupamento (*clustering*), regressão,

entre outros (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Dentre as áreas relacionadas a mineração de dados, este trabalho se preocupa principalmente com o campo da aprendizagem de máquina, pois o resultado almejado é concebido por meio do estudo de reconhecimento de padrões e algoritmos. Detalhes a seu respeito estão na seção seguinte.

2.3 APRENDIZAGEM DE MÁQUINA

A aprendizagem de máquina fornece técnicas baseadas em mineração de dados. Ela é usada para extrair informação a partir de dados brutos e pode ser utilizada para uma variedade de fins. A aprendizagem de máquina é interpretada como a aquisição de conhecimento de descrições estruturais a partir de exemplos (WITTEN; FRANK; HALL, 2011). Para Russell e Norvig (2003), o aprendizado de máquina possui a competência de adaptar-se as novas circunstâncias para detectar e extrapolar padrões. Mitchell (1997) ainda forneceu uma definição formal muito usada: com alguma classe de tarefas T e uma medida de desempenho P, um programa de computador *aprende* a partir de experiências E caso o seu desempenho em tarefas em T, medida pelo P, melhora com a experiência E.

Segundo Russel e Norvig (2003, p. 650), as tarefas de aprendizagem de máquina são geralmente classificadas em três grandes categorias:

- Aprendizagem supervisionada;
- Aprendizagem não supervisionada; e
- Aprendizagem por reforço.

A **aprendizagem supervisionada** é quase um sinônimo para a tarefa de classificação (HAN; KAMBER; PEI, 2011, p. 24). Métodos nesta categoria envolvem a tarefa de aprender ou otimizar uma função a partir de exemplos de suas entradas e saídas (RUSSELL; NORVIG, 2003, p. 650).

Já a **aprendizagem não supervisionada** é o sinônimo para a tarefa de análise agrupamento (*clustering*) (seção 2.4). O processo de aprendizagem é dito como não supervisionado quando os exemplos de entrada não possuem classes rotuladas, ou seja, o objetivo é descobrir classes dentro dos dados (HAN; KAMBER; PEI, 2011, p. 25).

Em **aprendizagem por reforço**, os algoritmos são projetados com o objetivo de que as ações que um agente venha a tomar possam maximizar (ou não) uma medida de recompensa. O agente não possui informação a priori do que fazer ou qual a ação a tomar; em vez disso, ele descobre através da exploração de ações que oferecem a maior recompensa e contribuem para atingir o objetivo proposto (LUGER, 2005, p. 442).

Aliás, o termo *supervisão* refere-se a uma “coluna” (ou característica) especial do conjunto de dados de entrada que é usada para direcionar o processo de mineração de dados (aprendizagem), tal como um professor pode supervisionar o seu aluno em direção a um objetivo específico (AGGARWAL, 2015, p. 15).

Por exemplo, considere dois conjuntos de dados praticamente iguais compostos por dados de alunos de ensino fundamental, ensino médio e superior. Entre as variáveis, têm-se as notas dos alunos, o comportamento, assiduidade, cumprimento dos prazos, entre outros. O objetivo do problema é identificar a escolaridade atual do aluno a partir das suas características (variáveis), isto é, classificar um aluno como *fundamental*, *médio* ou *superior*.

Considere também que a única diferença entre os conjuntos de dados é que o primeiro possui a informação do que se deseja buscar (a *escolaridade* do aluno) e o segundo, não. A *supervisão*, neste caso, atua no sentido de que a classe desejada (variável *escolaridade*) existe e ela será usada para o algoritmo **aprender** e, posteriormente, conseguir classificar novos alunos.

Portanto, como o primeiro conjunto de dados possui o valor da classe desejada, a tarefa é dita como supervisionada. Para o segundo caso, visto que as classes dos alunos não são informadas, a tarefa é dita como não supervisionada, isto é, os algoritmos deverão inferir os grupos por meio de análise de **similaridade** entre os alunos. Critérios de similaridade e índices de avaliação de grupos poderão ser utilizados para mensurar a qualidade dos grupos formados e alguns deles são apresentados na seção 2.4.2.

Há uma série de funcionalidades em mineração de dados e tipos de padrões que podem ser minerados, estes incluem: a caracterização e discriminação; a mineração de padrões frequentes, associações e correlações; classificação e regressão; análise de agrupamento; e análise de *outliers* (HAN; KAMBER; PEI, 2011, p. 15). Os métodos especificam o tipo de padrão a ser encontrado em tarefas de mineração de dados e as mais populares são apresentadas a seguir.

A tarefa de c**lassificação** é o processo de encontrar um modelo (ou função) que descreve e distingue classes de dados ou conceitos (HAN; KAMBER; PEI, 2011, p. 18). A análise de **regressão**, semelhante a tarefa de classificação, busca aprender uma função que mapeia um item de dados a uma variável de previsão de valor real (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996, p. 44) e abrange também a identificação das tendências de distribuição com base nos dados disponíveis (HAN; KAMBER; PEI, 2011, p. 19). Embora tenha uma grande semelhança entre estas duas abordagens, há uma distinção simples: a classificação prediz

categorias (não ordenadas e discretas) enquanto regressão se preocupa com funções de valores contínuos.

As **regras de associação**, popularmente aplicadas no contexto de *análise de cesta de compras* (*market basket analysis*) e muito comum no cenário de compras de supermercado, utiliza técnicas de associação de encontrar grupos de itens que tendem a ocorrer juntos em transações, como indicam Witten, Frank e Hall (2011). Em Schmitz (2015) e Araújo (2016), por exemplo, foi aplicado a técnica de associação *Apriori* para extrair relações entre os aspectos positivos mais frequentes nas avaliações obtidas por Christhie (2015). Assim como o presente trabalho, o estudo de Schmitz (2015) e Araújo (2016) também integra o projeto de pesquisa do SentimentALL (BRITO *et al*., 2015).

A **análise de agrupamentos** (ou *clustering*) é outra tarefa de mineração de dados bastante comum. Tendo em vista que ela é técnica utilizada neste trabalho, a seção que se segue detalha melhor as suas competências.

2.4 ANÁLISE DE AGRUPAMENTO

Segundo Luger (2005, p. 435), o problema de análise de agrupamentos (*clustering analysis*, em inglês) ou simplesmente *clustering*, começa quando se tem uma coleção de objetos não classificados e uma maneira de medir a similaridade entre eles. O objetivo é segmentar um conjunto de dados (*dataset*) em subconjuntos, onde cada subconjunto é chamado de *cluster* (HAN; KAMBER; PEI, 2011), ou ainda, agrupar os objetos em classes de tal modo que os objetos de um *cluster* são semelhantes uns aos outros, mas diferentes dos objetos de outros *clusters* (HAN; KAMBER; PEI, 2011).

O agrupamento de um grande número de objetos em grupos menores ajuda muito a resumir e compreender os dados. Uma definição informal e intuitiva da análise de agrupamentos é a seguinte: dado um conjunto de pontos, aplicar a análise de agrupamentos seria dividi-los em grupos que contenham pontos muito semelhantes (AGGARWAL, 2015, p. 153).

Outras definições de análise de agrupamentos são apresentadas a seguir:

- Tan, Steinbach, Kumar (2005, p. 487): “a análise de agrupamentos divide os dados em grupos que são significativos, úteis ou ambos”.
- Han, Kamber e PEI (2011, p. 20): diferente do que ocorre com as tarefas supervisionadas de classificação e predição, a análise de agrupamentos particiona objetos sem um rótulo de classe, ou seja, é uma tarefa não supervisionada que pode ser usada para gerar rótulos de classes para um conjunto de dados.

- Hu e Hao (2012, p. 17), “a análise de agrupamentos em si não é um algoritmo específico, mas uma tarefa de mineração de dados utilizada para resolver problemas”. Além de não se ter informações prévias a respeito de classes nas quais os objetos do conjunto de dados em questão pertencem.
- Jin e Lin (2011): a análise de agrupamentos é uma técnica de aprendizagem de máquina não supervisionada usada para descobrir estrutura de grupos de um conjunto de dados.

Ainda em relação a definição de análise de agrupamentos, Aggarwal (2015, p. 16) conceitua formalmente a tarefa como, “dado uma matriz de dados D, particionar suas linhas (registros) em conjuntos $C_1 \dots C_k$, de tal modo que as linhas em cada grupo são semelhantes umas às outras”.

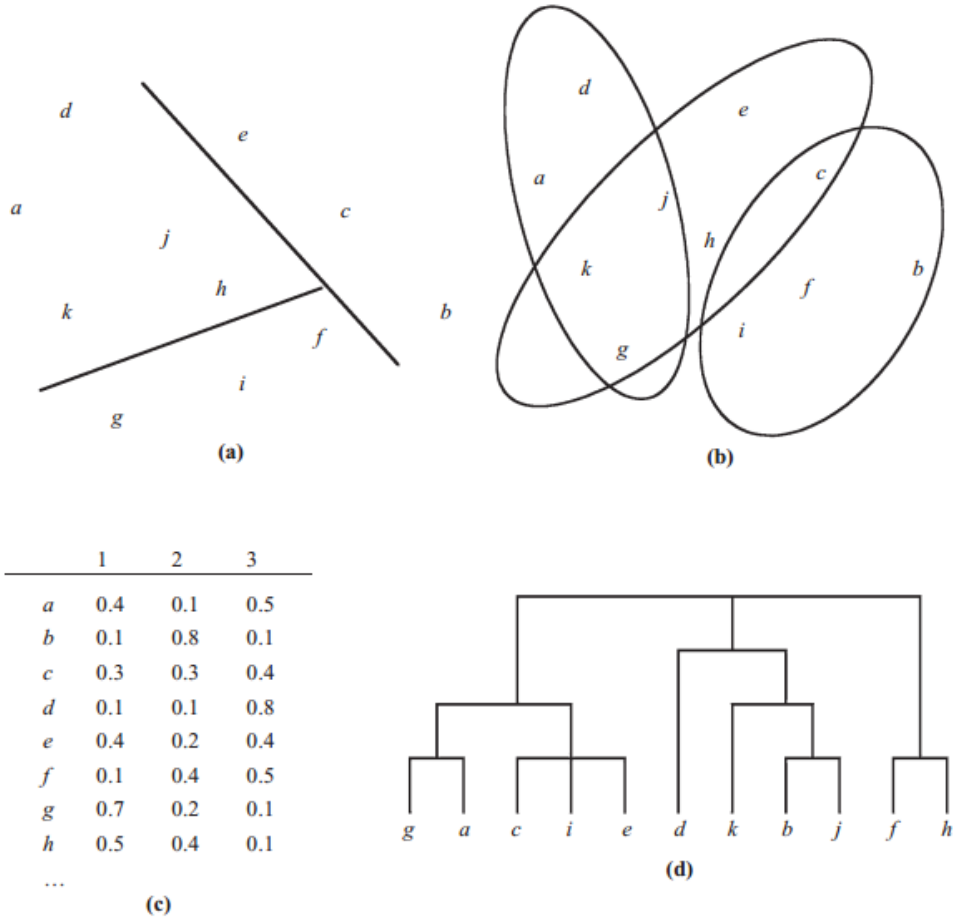
O agrupamento de um conjunto de dados em grupos intuitivamente similares é requerido em muitas aplicações como, por exemplo (AGGARWAL, 2015, p. 17):

- **Segmentação de cliente**: por meio da identificação dos perfis de clientes, uma agência bancária pode conceder diferentes tipos de planos, com diferentes limites de cartão de crédito, por exemplo.
- **Sumarização de dados**: dado que os agrupamentos podem ser considerados como grupos semelhantes de registros, estes grupos semelhantes podem ser usados para resumir os dados.
- **Aplicação em outros problemas de mineração de dados**: uma vez que o agrupamento é considerado uma versão não supervisionada de um problema de classificação, é comum ela ser utilizada como um bloco de construção para resolver este último.

2.4.1 Agrupamento como representação do conhecimento

Segundo Witten, Frank e Hall (2011), existem ainda diferentes formas em que o resultado da tarefa de análise de agrupamentos pode ser expressado, produzindo assim, diferentes visualizações que representam como os objetos estão agrupados. São eles, grupos exclusivos, grupos sobrepostos, grupos probabilísticos e grupos hierárquicos. A Figura 3 ilustra estas representações.

Figura 3 – Diferentes maneiras de representar agrupamentos.



	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1
…			

Fonte: adaptado de Witten, Frank e Hall (2011).

Cada forma de representação da Figura 3 pode ser interpretada da seguinte forma:

a. **Grupos exclusivos**: agrupamento na sua forma mais simples. Qualquer instância pertence a apenas um grupo.
b. **Grupos sobrepostos**: uma mesma instância que pode pertencer a vários grupos.
c. **Grupos probabilísticos**: indica a probabilidade de uma instância pertencer a um grupo, isto é, para cada exemplo, há uma probabilidade ou o grau de adesão dele pertencer a cada um dos agrupamentos.
d. **Grupos hierárquicos**: trata de uma divisão recursiva de instâncias, produzindo grupos hierárquicos.

Até então, muito foi dito sobre *similaridade*, *semelhança*, elementos *próximos* ou *distantes*, etc., porém, pouco deles foram explorados. Fato é que eles, na verdade, se tratam da essência dos problemas de *clustering*. Everitt *et al*. (2011, p. 43) e Aggarwal (2015, p. 17), por exemplo, apontam que o conhecimento sobre a forma na qual os objetos (ou dados) estão

próximos ou *distantes* uns dos outros é muito importante na tentativa de encontrar agrupamentos e que o projeto da função de similaridade é uma parte importante no processo de *clustering*. Logo, as próximas seções abordam estes importantes conceitos no estudo de análise de agrupamento, bem como outros temas frequentemente citados na literatura.

2.4.2 Medidas de similaridade

Segundo Everitt (2011, p. 43), muitos problemas de análise de agrupamentos têm como ponto de partida uma matriz $n \times n$, onde os elementos refletem uma medida quantitativa de proximidade, muitas vezes referido como *distância*, *similaridade*, *dissimilaridade* ou *proximidade*.

Segundo Linden (2009, p. 19), "algumas métricas calculam a similaridade, outras calculam a dissimilaridade, mas em essência elas são idênticas". O mesmo autor (LINDEN, 2009) aponta que todas as medidas de dissimilaridade são funções $d: \Gamma \times \Gamma$, onde Γ representa o conjunto de dados e permitem transformar uma matriz de dados em uma matriz de distâncias, representados pelas Equações 1 e 2, respectivamente.

$$\Gamma = \begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{il} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{nl} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix} \tag{1}$$

$$\Gamma = \begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,1) & 0 & & \\ \vdots & \vdots & \vdots & \vdots & \\ d(n,1) & d(n,1) & \cdots & \cdots & 0 \end{bmatrix} \tag{2}$$

Onde:

- x_{ij} é a posição do elemento na matriz. Já i é a característica do objeto (coluna) e j o valor (linha);
- $d(i,j)$ representa a distância entre os elementos i e j.

Ainda de acordo com Linden (2009, p. 19), as funções de similaridade devem obedecer aos seguintes critérios:

1. $d_{ij} \geq 0, \ \forall i,j \in \Gamma$
2. $d_{ij} = d_{ji}, \ \forall i,j \in \Gamma$
3. $d_{ij} + d_{jk} \geq d_{ik}, \ \forall i,j,k \in \Gamma$

Os critérios acima descritos podem ser interpretados da seguinte maneira:

1. A distância entre qualquer ponto deve ser um número positivo não nulo;
2. A distância entre dois elementos não é afetada pelo ponto a partir da qual ela é medida; e
3. A menor distância entre dois pontos é uma reta. Esta propriedade é conhecida como desigualdade triangular.

Em medidas de distância, a **euclidiana** é a mais conhecida (PRASS, 2004, p. 27). Considerando os objetos p e q com n dimensões (variáveis), a distância euclidiana em um espaço n-dimensional é definida por:

$$d(p_i, q_i) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \qquad (3)$$

Onde:

- n representa o número de variáveis do conjunto de dados; e
- p_i e q_i denotam os valores dos pontos p e q para a variável (dimensão) i.

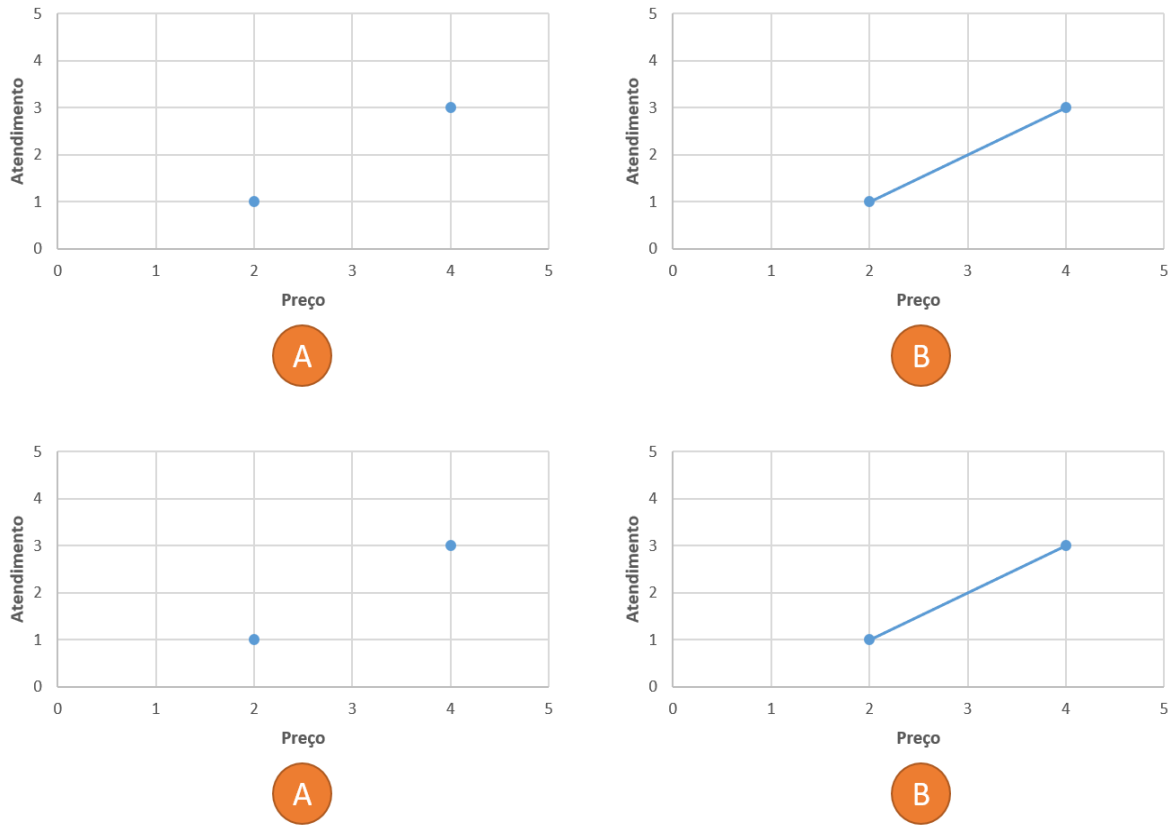
Para demonstrar a sua utilização, considere um exemplo bastante simples, na qual um conjunto de dados possui apenas dois objetos e duas variáveis, conforme apresentado no Quadro 1.

Quadro 1 – Conjunto de dados de exemplo para demonstrar a distância euclidiana.

atendimento	preço
1	2
3	4

Uma vez que o conjunto de dados acima se trata de um espaço bidimensional, uma representação gráfica pode ser feita por meio de um diagrama de dispersão, apresentado na Figura 4-A.

Figura 4 – Diagramas de dispersão para a demonstração da distância euclidiana.



Na Figura 4-B, a distância entre os dois pontos a e b, de acordo com a medida euclidiana e desprezando a unidade, pode ser calculada como $d(a,b) = \sqrt{(1-2)^2 + (3-4)^2} \cong 2{,}83$. Portanto, a distância entre os pontos a e b, representados pelo conjunto de dados do Quadro 1 e utilizando a medida euclidiana, é de aproximadamente de 2,83.

A distância euclidiana será a principal medida utilizada para avaliar os agrupamentos gerados neste trabalho. Aliás, a tarefa de avaliação de agrupamento é considerada uma etapa imprescindível para a relevância dos resultados obtidos. As competências desta avaliação estão detalhadas a seguir.

2.4.3 Avaliação de agrupamentos

A avaliação de agrupamento ou validação de agrupamento define formalmente como realizar avaliações objetivas para resultados quantitativos da análise de agrupamentos (WU, 2012, p. 5). A partir dela, a viabilidade e qualidade da análise de agrupamentos em um conjunto de dados pode ser medida (HAN; KAMBER; PEI, 2011, p. 491).

Segundo Aggarwal (2015, p. 195), “a validação de *cluster* é muitas vezes difícil em conjuntos de dados reais porque o problema é definido de uma forma sem supervisão”. Em

alguns casos, podem existir um indicador chamado *ground-truth*, onde os verdadeiros agrupamentos são conhecidos (AGGARWAL, 2015). Neste caso, o problema é dito que possui índices externos, visto que eles usam informações que não estão presentes no conjunto de dados (supervisionada)(TAN; STEINBACH; KUMAR, 2005, p. 535). Outra terminologia utilizada para indicar índices externos são os métodos extrínsecos, tal como indicado por Han, Kamber e Pei (2011).

Juntamente dos índices externos, as medidas de validação internas são os tipos de validação de agrupamento mais tradicionais (AGGARWAL, 2015; HAN; KAMBER; PEI, 2011; WU, 2012). Os índices internos de validação de agrupamento serão melhor detalhados a seguir, devido ao conjunto de dados do presente trabalho não possuir uma *ground-truth* e serem conduzidos por um problema não supervisionado.

Os índices de critério interno medem o quão bom é um agrupamento desconsiderando informação externa (*ground-truth*) (WU, 2012, p. 535). Segundo Tan, Steinbach e Kumar (2005, p. 535), os índices de critério interno podem ainda ser divididos em:

- **Medidas de coesão**: que determina como os objetos de um agrupamento são semelhantes; e
- **Medidas de separação**: que definem o quão distinto e bem separados um agrupamento é de outro agrupamento.

De acordo com Aggarwal (2015, p. 196), os critérios de avaliação interna comumente utilizados são os seguintes:

- Soma do quadrado da distância para os centroides;
- Relação de distância *intra-cluster* para *inter-cluster*; e
- Coeficiente de silhueta.

O índice da **soma do quadrado das distâncias para os centroides**, como o próprio nome sugere, tem como função objetivo a soma do quadrado das distâncias (SQD) entre os pontos e os centroides. A melhor qualidade de agrupamento é indicada pelos menores valores da SQD. Logo, esta abordagem é mais adequada para algoritmos baseados em distância, como o K-Means (seção 2.5).

No índice **relação de distância *intra-cluster* pra *inter-cluster***, a ideia é provar pares r de pontos de dados a partir dos dados subjacentes. Destes, seja P o conjunto de pares que pertencem ao mesmo conjunto encontrado pelo algoritmo, os pares remanescentes são

denotados por Q. A distância média *inter-cluster* e a distância média *intra-cluster* são definidas respectivamente como:

$$Intra = \sum_{(\overline{X_i},\overline{X_j})\in P} dist(\overline{X_i},\overline{X_{ij}})/|P| \quad (4)$$

$$Inter = \sum_{(\overline{X_i},\overline{X_j})\in P} dist(\overline{X_i},\overline{X_{ij}})/|Q| \quad (5)$$

Em seguida, a razão entre a distância média *intra-cluster* e distância *inter-cluster* é dada pela razão *intra-cluster* por *inter-cluster*. Pequenos valores desta medida indicam melhores agrupamentos.

Já o **coeficiente de silhueta** é uma popular medida que combina as medidas de coesão e de separação (TAN; STEINBACH; KUMAR, 2005, p. 451), isto é, ela indica o quão similar é um objeto do seu próprio agrupamento (coesão) em comparação com outros agrupamentos (separação).

Considere dois agrupamentos C e D, e C_i como sendo os objetos pertencentes ao agrupamento C. Dado que se deseja calcular o coeficiente de silhueta de C e que D é o agrupamento mais próximo dele, o coeficiente de silhueta é calculado (NUNES, GIOPPO, 2015) da seguinte maneira:

$$s(i) = \frac{b(i) - a(i)}{\max\big(a(i), b(i)\big)} \quad (6)$$

Onde:

- $s(i)$ é o coeficiente de silhueta dos objetos $i, \dots, n$;
- a(i) é a dissimilaridade média do objeto i a todos os objetos do agrupamento C; e
- $b(i)$ é a dissimilaridade média do objeto i a todos os objetos do agrupamento D.

O coeficiente de silhueta ($s(i)$) varia no intervalo de -1 e 1 e é interpretado da seguinte maneira:

- $s(i) \approx 1$, objeto i foi bem classificado no grupo, ou seja, ele está muito próximo dos objetos do seu grupo em comparação com o seu vizinho;
- $s(i) \approx -1$, objeto i foi mal classificado no grupo, tendo em vista que ele está mais próximo de D do que em comparação a C; e
- $s(i) \approx 0$, objeto i está em um ponto intermediário entre os dois grupos.

Outro critério interno citado por Aggarwal (2015, p. 196) é a medida probabilística. O objetivo dela é usar um modelo misto para estimar a qualidade de um particular agrupamento. Seu uso é aplicado a algoritmos de agrupamento de abordagem estatística (como o *Expectation-maximization)*, logo, não é de competência deste trabalho.

2.4.4 Métodos de agrupamento

Alguns algoritmos de análise de agrupamentos integram vários métodos diferentes, fato que dificulta a classificação de um determinado algoritmo como sendo pertencente exclusivo de uma única categoria. Além disso, algumas aplicações podem ter critérios de agrupamento que exigem a integração de várias técnicas de agrupamento (HAN; KAMBER; PEI, 2011, p. 450).

Geralmente os principais métodos de agrupamento podem ser classificados em hierárquicos, de particionamento, baseados em densidade e baseados em grade (HAN; KAMBER; PEI, 2011). Os métodos citados acima estão descritos a seguir:

- **Métodos hierárquicos**: criam uma decomposição hierárquica de um dado conjunto de observações (Figura 3-C). Baseado em como essa decomposição é formada. Métodos dessa categoria podem ainda ser divididos em:
    - **Aglomerativos (*bottom-up*)**: iniciam com cada objeto formando um grupo separado. Então os dados são particionados sucessivamente até que uma representação hierárquica dos agrupamentos – chamada dendograma – seja produzida (Figura 3-D).
    - **Divisivos (*top-down*)**: abordagem oposta ao método aglomerativo. Um único agrupamento contém todas as observações que, passo a passo, sofrerá sucessivas divisões.

- **Métodos de particionamento**: o método de análise de agrupamentos por particionamento obtém uma única partição dos dados em vez de uma estrutura de grupos (Figura 3-A). A maioria dos métodos de particionamento agrupam os objetos com base na distância entre eles (seção 2.4.2). Eles conseguem encontrar agrupamentos de formato esférico e tem dificuldades em descobrir agrupamentos de formato arbitrário. Um problema que acompanha o uso de um algoritmo de particionamento é a escolha do número de agrupamentos desejados (JAIN; MURTY; FLYNN, 1999). Este trabalho utilizará o método de particionamento chamado K-Means e ele será melhor explicado na seção 2.5.

- **Métodos baseados em densidade**: outros métodos de agrupamento foram desenvolvidos com base na noção de densidade. A sua ideia geral é a que a forma do agrupamento possa se expandir (utilizando densidade) enquanto sua "zona" não exceda a duas limiares popularmente conhecidas como *Eps* e *MinPts* (HAN; KAMBER; PEI, 2011, p. 449). Por exemplo, para cada ponto de dados dentro de um grupo, a vizinhança de um dado raio (*Eps*) tem de conter, pelo menos, um número mínimo de pontos (*MinPts*). Este método pode ser usado para filtrar ruídos e *outliers*, bem como descobrir agrupamentos de formato arbitrário. Este trabalho utilizará o método baseado em densidade DBSCAN e ele será melhor explicado na seção 2.6.
- **Métodos baseados em grade**: segundo Han, Kamber e Pei (2011), os métodos baseados em grade quantificam o espaço do objeto em um número finito de células que formam uma estrutura de grade. Todas as operações de agrupamento são realizadas sobre a estrutura da rede, isto é, no espaço quantificado. A principal vantagem desta abordagem é o seu tempo rápido de processamento, que geralmente é independente do número de objetos de dados e depende apenas do número de células em cada uma das dimensões no espaço quantificado.

As seções a seguir tratam dos algoritmos de análise de agrupamentos K-Means e DBSCAN que serão utilizados no presente trabalho.

2.5 K-MEANS

Apesar do K-Means ter sido proposto pela primeira vez há mais de 50 anos, ele ainda é um dos algoritmos de análise de agrupamentos mais utilizados e tem sido estendido de muitas maneiras (JAIN, 2010, p. 3). Suponha que o conjunto de dados D contém n objetos, o método de particionamento K-Means irá distribuir os objetos de D em k agrupamentos ($C_1, \dots, C_k$), onde $C_i \subset D$ e $C_i \cap C_j = \emptyset$, para ($i \leq k, j \leq k$).

No algoritmo K-Means, a soma do quadrado das distâncias euclidianas (seção 2.4.2) dos seus objetos para os seus representantes mais próximos – chamados de centroides – é usada para quantificar a função objetivo do agrupamento (AGGARWAL, 2015, p. 162), ou seja, a partição é encontrada de tal modo que o erro quadrático médio entre a média empírica de cada cluster e os pontos do cluster é minimizada (JAIN, 2010, p. 3).

Ainda segundo Jain (2010, p. 653), seja μ_k a média de um cluster C_k. O erro quadrático entre μ_k e os pontos em C_k é definida como:

$$J(C_k) = \sum_{x_i \in C_k} \left||x_i - \mu_k|\right|^2 \quad (7)$$

Onde:

- $J(C_k)$ representa o erro quadrático médio do *cluster* C_k;
- x_i é o i-ésimo objeto do conjunto de dados;
- μ_k é a média do *cluster* C_k, ou seja, o centroide do *cluster* C_k.

O objetivo de K-Means é minimizar a soma do erro quadrático sobre todos os k agrupamentos:

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in C_k} \left||x_i - \mu_k|\right|^2 \quad (8)$$

Onde:

- $J(C_k)$ representa o erro quadrático médio do *cluster* C_k;
- x_i é o i-ésimo objeto do conjunto de dados;
- μ_k é a média do *cluster* C_k, ou seja, o centroide do *cluster* C_k.

Segundo Han, Kamber e Pei (2011, p. 452), o algoritmo K-Means é resumido da seguinte maneira:

Entrada:

- D: o conjunto de dados contendo n objetos;
- k: o número de *clusters.*

Saída: um conjunto de k *clusters.*

Método:

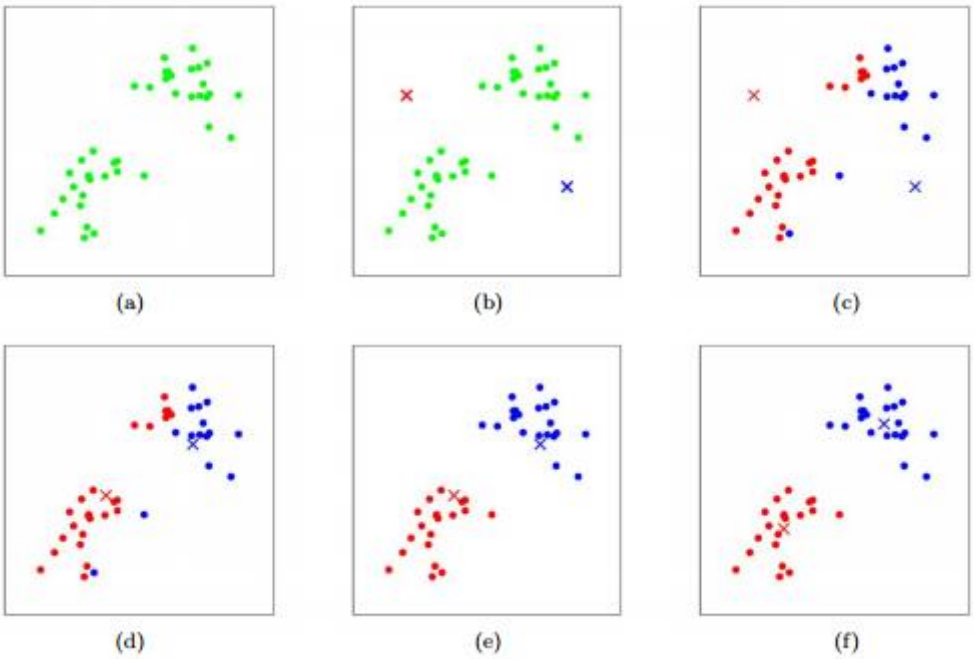
Passo 1: seleção de k objetos do conjunto de dados D como os centroides iniciais.
Passo 2: **repete**
Passo 3: cada objeto é atribuído ao cluster na qual a distância entre o objeto e o cluster (representado pelo centroide) é a menor.
Passo 4: atualizar as médias dos clusters, isto é, calcular a média dos objetos para cada cluster.
Passo 5: **enquanto** não houver convergência (não ocorrer mudanças).

Tendo como base o algoritmo acima, a seção a seguir seguinte demonstra um exemplo prático da utilização do K-Means.

2.5.1 Demonstração do K-Means

Para demonstrar o funcionamento básico do K-Means, considere o algoritmo descrito na seção anterior e os elementos da Figura 5. Considere também que a quantidade de agrupamentos desejados são 2 ($k = 2$).

Figura 5 – Diagramas de dispersão para a demonstração do K-Means.



Fonte: Piech e Ng (2016).

Na Figura 5-A, o conjunto de dados está disposto em um gráfico (bidimensional) de dispersão. Nele, sob a ótica humana, a identificação de dois agrupamentos é bastante intuitiva, tendo em vista que os aglomerados de objetos estão bem definidos e distantes um do outro. O objetivo do K-Means é exatamente este, identificar grupos particionados de dados de acordo com a metodologia descrita na seção anterior. Os passos para a identificação dos dois *clusters* ilustrados na Figura 5 são os seguintes:

1. K pontos são selecionados, de forma aleatória, como centroides inicias. Os centroides são representados pelo elemento “x” na Figura 5-B;
2. Com a métrica de distância definida, cada um dos objetos são atribuídos ao *cluster* – que é representado pelo centroide – mais próximo (Figura 5-C);
3. O valor dos centroides é atualizado. Assim, a posição dos centroides na Figura 5-D e Figura 5-F já é diferente das anteriores.
4. Os passos 2 e 3 são repetidos até que não ocorram mudança nos grupos.

O K-Means é um algoritmo simples, contudo, representa bem a intuição geral de um problema de análise de agrupamentos: um grupo é formado pelo conjunto de objetos mais próximos uns dos em comparação aos objetos de outros grupos. A maneira do K-Means e de outros algoritmos baseados em particionamento realizar essa atribuição, é feita por meio de uma medida de tendência central que, no contexto de análise de agrupamentos, é chamada de centroide. Portanto, o centroide é um dos principais elementos de problemas deste âmbito. Inclusive, como algoritmos de aprendizagem de máquina, a ideia de *aprender* diz respeito justamente ao valor do centroide, que é a média dos elementos de um grupo e que torna possível a definição do que se compreende por *proximidade*.

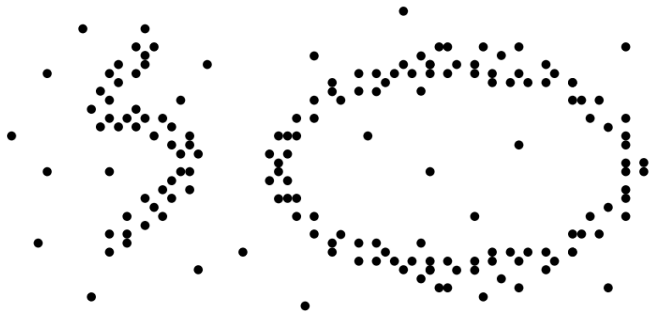
O outro algoritmo de análise de agrupamentos utilizado neste trabalho é DBSCAN. Ele utiliza a abordagem baseada em densidade (seção 2.4.4) e está detalhado a seguir.

2.6 DBSCAN

DBSCAN ou Agrupamento Espacial Baseada em Densidade de Aplicações com Ruído – do inglês *Density Based Spatial Clustering of Application with Noise* – é um método de *clustering* não paramétrico baseado em densidade proposto por Ester *et al*. (1996). A noção de *cluster* baseado em densidade é utilizada para descobrir *clusters* de formato arbitrário, visto que clusters podem ser esféricos, lineares, alongados, etc.

Agrupamentos também podem ser modelados como sendo regiões de densidade no espaço de dados e separados por regiões dispersas. A principal estratégia por trás de métodos de agrupamento baseados em densidade – como o DBSCAN – é sua habilidade de descobrir agrupamentos de formato não esféricos como, por exemplo, os agrupamentos em forma de “S” e ovais, conforme exemplificado na Figura 6 (HAN; KAMBER; PEI, 2011).

Figura 6 – Exemplos de agrupamentos de formato arbitrário.

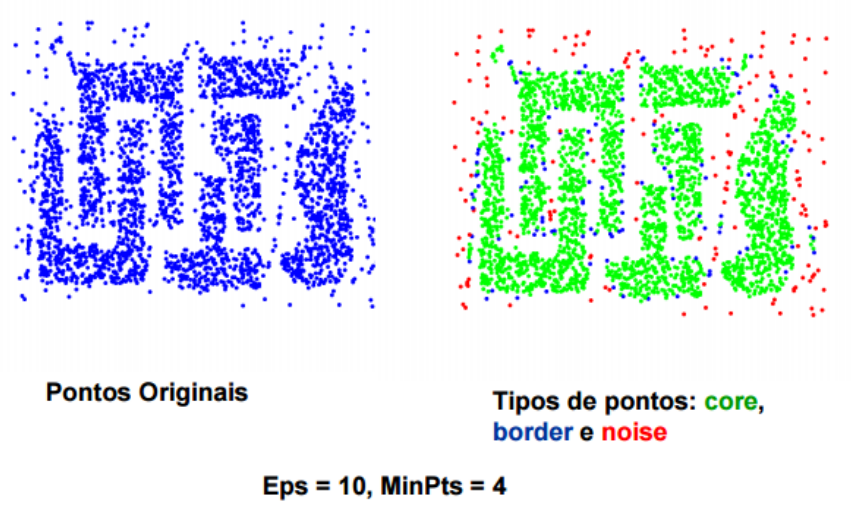


Fonte: Han, Kamber e Pei (2011, p. 471).

Segundo Tan, Steinbach e Kumar (2005, p. 527) a abordagem de baseada em densidade permite classificar um ponto como sendo (1) o interior de uma região densa (*core point*), (2) uma borda da região densa (*border point*) ou (3) uma região pouco ocupada (*noise point* ou *background point*), que estão ilustradas na Figura 7 e apresentadas a seguir:

- ***Core points***: São os pontos no interior do *cluster* baseado em densidade. Um ponto é um classificado como *core point* se ele possuir o número mínimo de pontos (*MinPts*) necessários na sua vizinhança, que é determinado por um parâmetro (*Eps*), ou seja, o conceito de vizinhança é uma função de distância que limita o alcance da região densa em volta de um ponto.
- ***Border points***: um *border point* não é um *core point*, mas fica dentro da vizinhança de um *core point*.
- ***Noise points***: um *noise point* qualquer ponto que não é *core point* e nem *border point*.

Figura 7 – DBSCAN: *core*, *border* e *noise points*.



Fonte: Oliveira (2016).

Segundo Han, Kamber e Pei (2011, p. 452), o algoritmo do DBSCAN é o que se segue:

Entrada:

- D: conjunto de dados contendo n objetos;
- Eps: parâmetro de raio; e
- $MinPts$: limiar de densidade de vizinhança.

Saída: Um conjunto de *clusters* baseados em densidade.

Método:

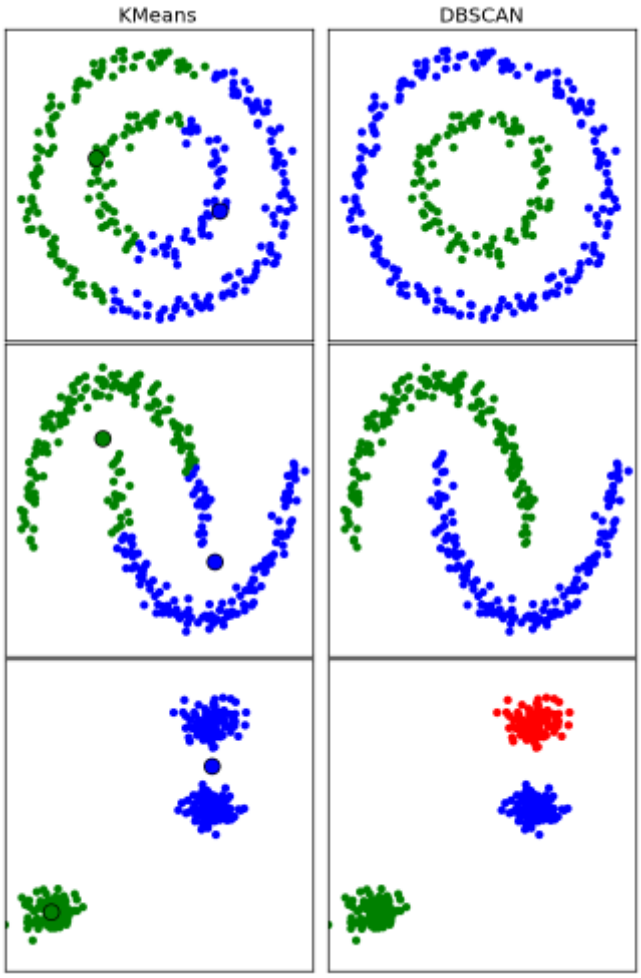
Passo 1: marcar todos os objetos como não visitados.

Passo 2: **faça**

Passo 3: selecionar randomicamente um objeto não visitado p;
Passo 4: marcar p como visitado;
Passo 5: **se** a *Eps*-vizinhança de p tem pelo menos $MinPts$ objetos
Passo 6: cria um novo cluster C e adiciona p em $C;$
Passo 7: seja N o número de objetos na *Eps*-vizinhança de $p;$
Passo 8: **para** cada ponto p' em N
Passo 9: **se** p' não foi visitado
Passo 10: marcar p' como visitado;
Passo 11: **se** a *Eps*-vizinhança de p' tem pelo menos $MinPts$
Passo 12: adicionar esses pontos em $N;$
Passo 13: **se** p' ainda não é membro de algum cluster:
Passo 14: adicionar este ponto em $N;$
Passo 15: **fim para**
Passo 16: saída $C;$
Passo 17: **se não** marcar p como *noise point;*
Passo 18: **enquanto** existir objetivos não visitados.

Para métodos baseados em particionamento, encontrar agrupamentos de formato arbitrário (como a Figura 6) pode não ser uma tarefa fácil. Isso ocorre devido ao fato de ser difícil definir grupos de objetos que estão próximos entre si, mas distantes dos objetos de outros grupos. Essa característica é melhor compreendida na Figura 8.

Figura 8 – Comparação entre os algoritmos K-Means e DBSCAN.



Fonte: adaptado de Scikit-learn (2011).

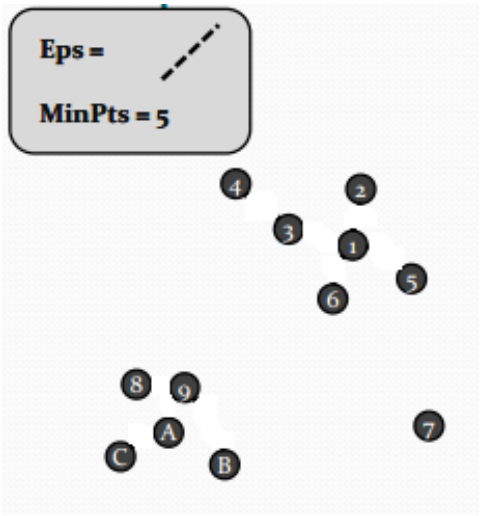
A seção a seguir demonstra DBSCAN utilizando um conjunto de dados fictício e os passos descritos anteriormente para o seu algoritmo.

2.6.1 Demonstração do DBSCAN

O exemplo de demonstração do DBSCAN foi reproduzido de Vendramin (1996).

Considere o conjunto de dados $D = [1,2,3,4,5,6,7,8,9, A, B, C]$. Considere também que o parâmetro de entrada da limiar de densidade de vizinhança seja 5 ($MinEps = 5$) e que o raio (Eps) seja representado pela linha tracejada da Figura 9. A Figura 9 também apresenta o conjunto de dados dispostos em uma representação de duas dimensões.

Figura 9 – Conjunto de dados para demonstração do DBSCAN.



Fonte: adaptado de Vendramin (1996).

Han, Kamber e Pei (2011, p. 452) apontam que a seleção do objeto a ser visitado é feita de maneira aleatória, contudo, esta particularidade foi omitida nesta demonstração para facilitar a fluidez dos passos.

Seguindo a Figura 10, o primeiro objeto a ser visitado é o (1), caso o seu raio de vizinhança (Eps) possua pelo menos 5 objetos ($MinPts$), logo ele é classificado como um *core point* e os demais objetos da sua região densa são classificados como *border point*. O detalhe é que o objeto que está sendo visitado também entra na contagem, por este motivo a vizinhança atingiu os 5 objetos mínimos (Figura 10-1).

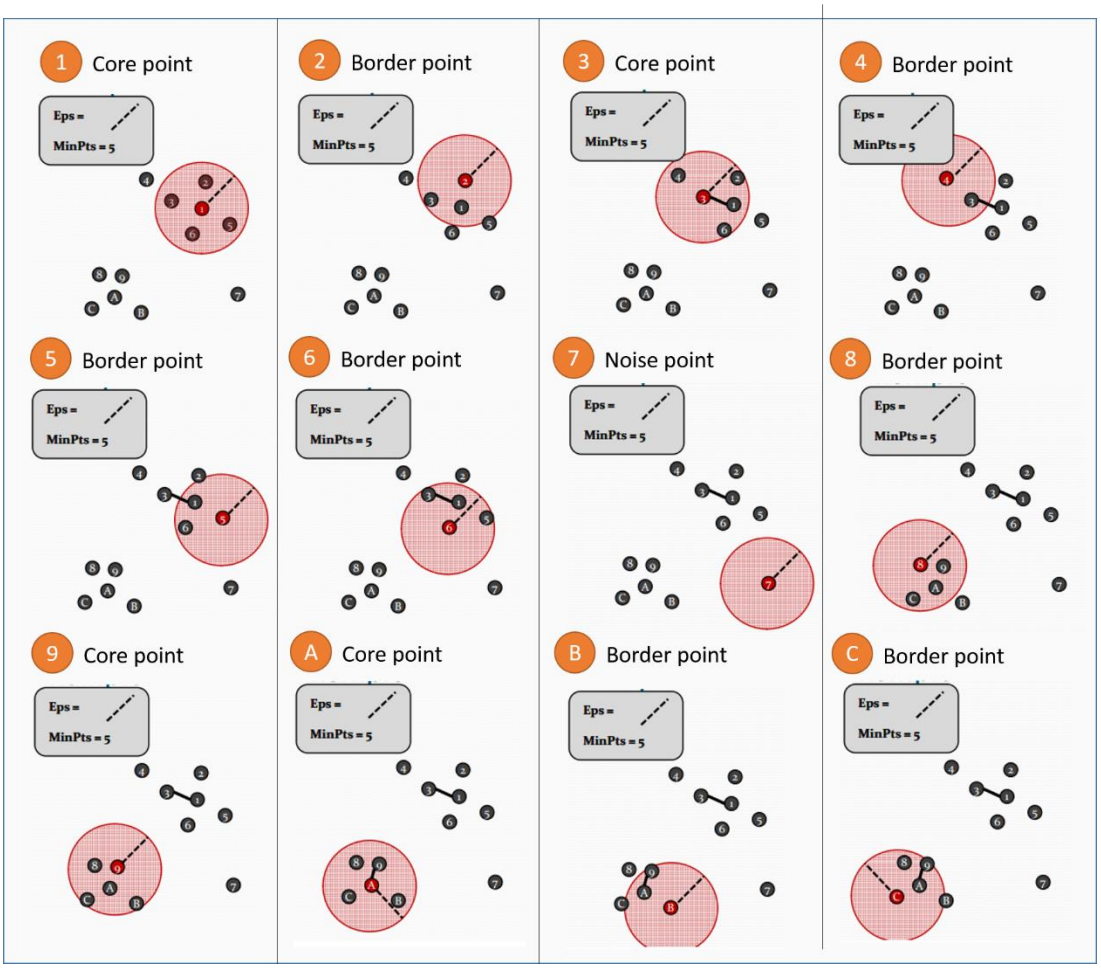
Considerando o objeto (2), a Figura 10-2 mostra que a região densa de (2) não atingiu o número de mínimo de objetos na vizinhança ($MinPts = 5$), logo, a sua classificação continua sendo como *border point*.

Considere agora o objeto Figura 10-7, classificado como um *noise point*. As razões pela qual ele é um *noise point*, são:

- Não pode ser um *core point* porque não ele não possui 5 elementos na sua região densa; e
- Não pode ser um *border point* porque nenhum outro objeto o "alcança".

Portanto, o objeto (7) é classificado como *noise point*.

Figura 10 – Estados de cada iteração para demonstração do algoritmo DBSCAN.



Fonte: adaptado de Vendramin (1996).

Como referido no algoritmo, cada objeto deverá ser visitado e classificado como sendo um *core*, *border* ou *noise point*. Cada etapa deste processo está ilustrado na Figura 10.

A próxima seção trata do LDA, um algoritmo de modelagem probabilística de tópicos que será utilizado neste trabalho e também possui a característica de gerar grupos de dados, mas utilizando uma abordagem um pouco diferente da análise de agrupamentos pura.

2.7 MODELAGEM PROBABILÍSTICA DE TÓPICOS

Os modelos probabilísticos de tópicos têm apresentado bom desempenho em tarefas de sumarização, categorização e agrupamentos, fato que recentemente tem atraído muita atenção da comunidade científica de mineração de texto (LU; MEI; ZHAI, 2011, p. 178) e motivou a utilização do LDA no presente trabalho.

Os algoritmos desta abordagem são importantes em aplicações onde se deseja categorizar documentos em temas ou assuntos, na qual não se faz necessário nenhuma rotulação

a priori (abordagem não supervisionada) dos documentos, isto é, temas são descobertos a partir da análise de uma coleção de documentos e, para tal finalidade, são utilizados métodos estatísticos (BLEI, 2012, p. 77).

Em síntese, a noção fundamental de modelos probabilísticos de tópicos é que os documentos são misturas de temas, onde um tópico é representado por uma distribuição multinomial[2] de palavras, ou seja, um modelo de linguagem unigrama (LU; MEI; ZHAI, 2011). O objetivo é descobrir estruturas temáticas latentes (ocultas) em grandes coleções de documentos e o termo *tópico oculto* é usado para expressar um assunto ou tema que ocorre em documentos semanticamente relacionados por meio de estruturas temáticas ocultas (FALEIROS; LOPES, 2016, p. 9).

2.7.1 LDA

O *Latent Dirichlet Allocation* (LDA) é um modelo probabilístico generativo para coleções de dados discretos, como corpus de documentos e é tido como o estado da arte em modelos probabilísticos de tópicos (FALEIROS; LOPES, 2016). O LDA é um modelo *bayesiano* hierárquico de três níveis em que cada item de uma coleção é modelado como uma mistura finita sobre um conjunto subjacente de temas (BLEI; NG; JORDAN, 2003). Em outras palavras, o modelo permite que conjuntos de observações possam ser explicados por meio de grupos ocultos e também da similaridade dos seus dados. Recentemente, modelos probabilísticos de tópicos, como o LDA, têm sido fortemente utilizados em tarefas de *clustering*, obtendo bons resultados (KELAIAIA; MEROUANI, 2013).

Segundo Blei (2012, p. 3), um *tópico* é definido como uma distribuição de probabilidades sobre um vocabulário fixo. Para cada documento da coleção, as palavras são geradas em duas etapas:

1. Escolher aleatoriamente uma distribuição sobre os tópicos;
2. Para cada palavra no documento:
    a. Escolher aleatoriamente um tópico de uma distribuição sobre o passo #1.
    b. Escolher aleatoriamente uma palavra para corresponder a distribuição sobre o vocabulário.

Este trabalho aborda o LDA, o algoritmo que se encontra na literatura como sendo estado da arte em modelos probabilísticos de tópicos (FALEIROS; LOPES, 2016, p. 13). A

[2] A distribuição multinomial é uma extensão da distribuição binomial, onde cada saída possui $k \geq 2$ resultados possíveis.

fundamentação teórica por trás da modelagem de tópicos e do LDA é vasta e possui um embasamento estatístico muito forte. Modelos de linguagens, modelos generativos, distribuição multinomial, distribuição *dirichlet*, variáveis latentes, teorema de *bayes*, modelo mistura, redução de dimensionalidade, entre outros, são alguns dos exemplos de conceitos que certamente precisariam de um enfoque maior para uma boa compreensão do algoritmo. Logo, não compete a este trabalho o seu detalhamento, o foco será no resultado obtido, que permitirá entender como os tópicos (ou agrupamentos) formados são influenciados pelos seus aspectos.

2.7.2 Demonstração do LDA

Para demonstrar o funcionamento do LDA, considere um problema que se deseja obter 3 tópicos ocultos ($k = 3$) nas avaliações a seguir:

1. Gostei. Só achei um pouco caro.
2. Bem localizado e bom café.
3. Ótima localização e preço em conta.
4. Um pouco longe, mas ótimo restaurante.

O Quadro 2 apresenta as probabilidade $P(w \in t_n)$ de um termo w pertencer ao tópico oculto t_n de gerado pelo LDA:

Quadro 2 – Tópicos de exemplo gerados pelo LDA.

t_1	$P(w \in t_1)$	t_2	$P(w \in t_2)$	t_3	$P(w \in t_3)$
localização	34,8%	pouco	17,9%	café	21,3%
preço	20%	caro	15,3%	bom	17,5%
conta	14,5%	ótimo	10,3%	localização	13%
ótima	12,7%	gostei	10,3%	bem	11,4%
⋮	⋮	⋮	⋮	⋮	⋮

Onde podemos destacar:

- No tópico t_1, os termos *localização*, *preço*, *conta* e *ótima* apresentam maior relevância estatística neste grupo, isto é, eles são os quatro termos que melhor explicam o tópico (ou agrupamento).
- Ainda em relação ao tópico t_1, o termo *localização* é o elemento mais representativo daquele tópico, apresentando uma probabilidade de 34% de estar presenta naquele grupo.
- Cada termo da coleção de documentos terá uma representatividade (probabilidade) no tópico, por isso o uso das reticências verticais foi utilizado.

Uma transformação dos dados deverá ser realizada para processa-los no LDA, tendo em vista que o objetivo é gerar tópicos de avaliações de usuários baseado em aspectos, ou seja, não será utilizada a avaliação inteira. Este procedimento está melhor detalhado na seção 4.3.

A tarefa de análise de agrupamentos se torna um problema de pesquisa interessante porque envolve muitos conceitos como medidas de similaridade, coesão, separação, espaços dimensionais, entre outros, além de contar com diversas abordagens para agrupar objetos, tais como por particionamento, por regiões densas, definido estatisticamente, etc.

Outra característica bastante peculiar se dá pela capacidade de percepção e julgamento subjetivo, por exemplo, mesmo que o resultado de uma análise de agrupamentos possua um bom índice de avaliação, ou ainda que esteja em conformidade com a realidade de um determinado domínio, os grupos formados podem ainda não ser de grande valia para uma fração das partes interessadas, entre outros casos.

A seção a seguir descreve os materiais e métodos utilizados no trabalho.

3 MATERIAIS E MÉTODOS

Este capítulo apresenta os materiais e os métodos utilizados no desenvolvimento do módulo de análise e visualização de agrupamentos para o SentimentALL.

3.1 MATERIAIS

O *front-end* do módulo foi construído com as tecnologias Bootstrap, AngularJS, além do Google Chart Tools para a geração de gráficos.

O Bootstrap é um dos *frameworks* HTML, CSS e JavaScript mais populares para o desenvolvimento de aplicações móveis responsivas na web (BOOTSTRAP, 2016).

O AngularJS é um *framework* que permite o desenvolvimento de aplicações web do lado do cliente. Um dos seus principais recursos é a sincronização automática dos dados da interface gráfica (*views*) com seus objetos JavaScript (*models*) através da vinculação de dados bidirecional, também conhecida como vinculação *2-way data binding*.

O Google Chart Tools é uma biblioteca de gráficos interativos para navegadores e dispositivos móveis (DEVELOPERS, 2016).

No *back-end* foi utilizado o Flask como servidor web e o Scikit-learn[3] como biblioteca que implementa os algoritmos K-Means, DBSCAN e LDA, todos escritos na linguagem de programação Python[4].

O Flask é um *micro-framework* que permite a construção de aplicações web, definição de rotas HTTP e o consumo de dados remoto (FLASK, 2016).

O Scikit-learn é uma biblioteca *open source* que integra uma vasta gama de algoritmos de aprendizagem de máquina para problemas supervisionados e não-supervisionados de média escala (PEDREGOSA *et al.*, 2011).

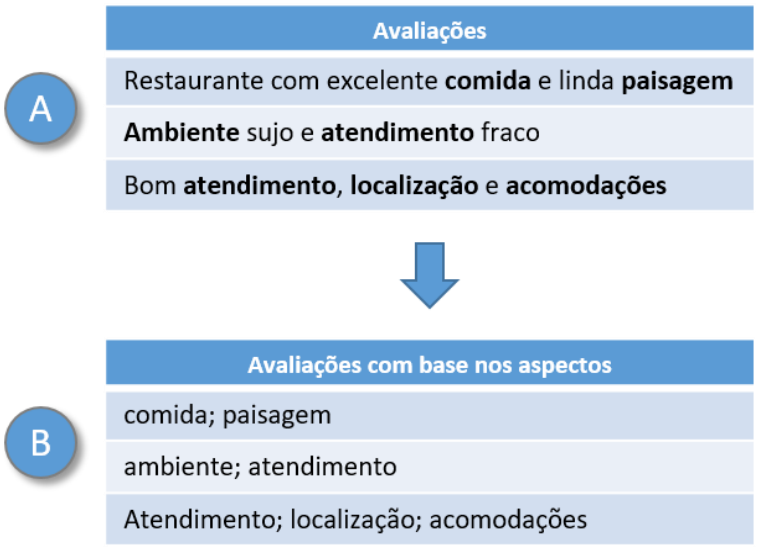
Uma visão geral da maneira na qual as tecnologias se interagem está ilustrada no Apêndice A.

Outro material importante no trabalho foi o conjunto de dados utilizado. Este conjunto possui 1.415.476 avaliações que foram aplicadas em um processo de análise de sentimentos baseada em aspectos (CHRISTHIE, 2015), onde foi realizado a extração dos aspectos das avaliações e a polarização deles. As avaliações se referem a comentários de usuários do TripAdvisor sobre destinos turísticos e retomam a expressão *avaliações baseadas nos aspectos*

[3] Versão 0.18.0.
[4] Versão 2.7 (x64).

mencionada na seção 1. Um exemplo que ilustra como as *avaliações baseadas nos aspectos* são geradas está apresentado na Figura 11.

Figura 11 – Demonstração do processo de geração das avaliações baseadas nos aspectos.

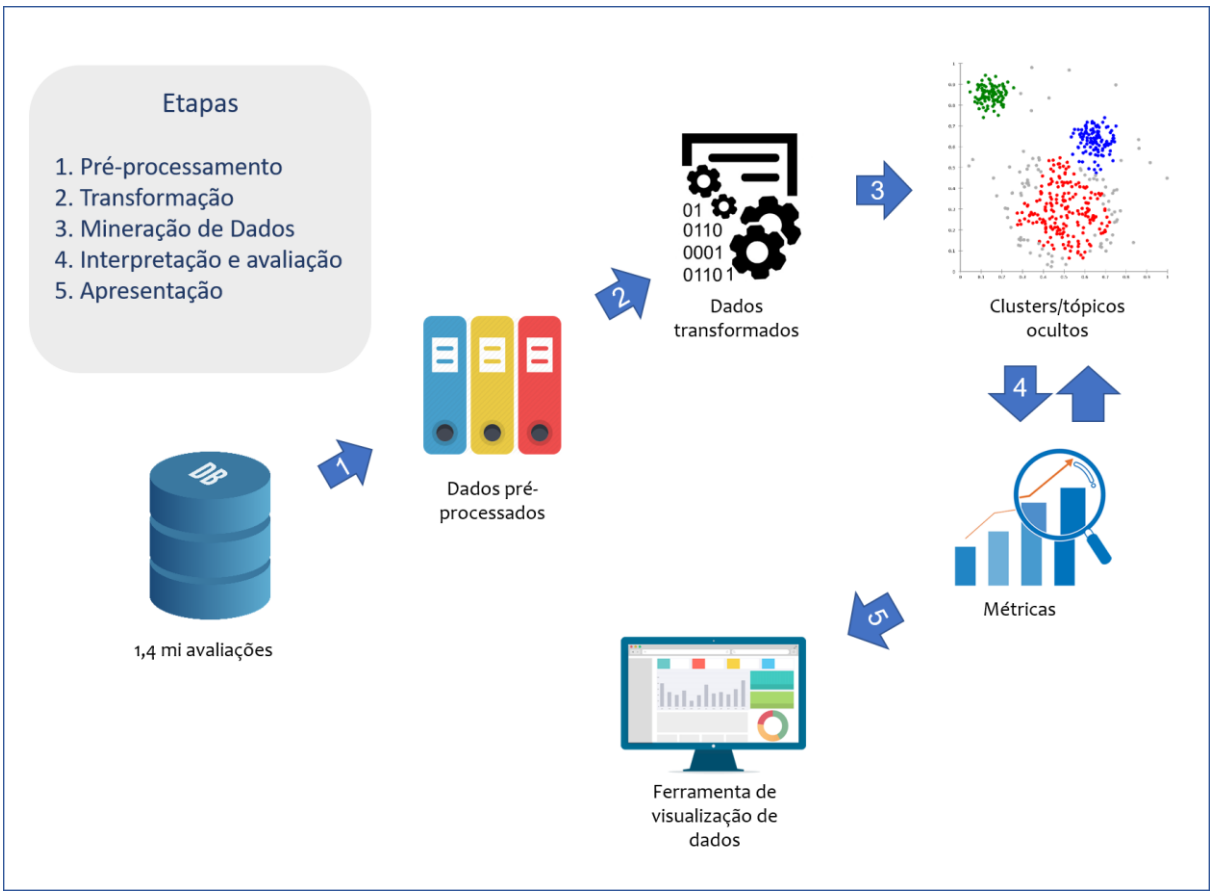


A Figura 11-A contém três avaliações escolhidas aleatoriamente e a Figura 11-B o conjunto de aspectos identificados para cada uma delas. Como pode ser observado, as *avaliações baseadas nos aspectos* indicam a definição de uma avaliação pelo seu conjunto de aspectos identificados. Mais detalhes sobre essa transformação serão apresentados na seção 4.3.2.1.

3.2 MÉTODOS

A metodologia adotada no trabalho foi baseada no processo de KDD proposto por Fayyad, Piatetsky-shapiro e Smyth (1996) e está ilustrada na Figura 12.

Figura 12 – Metodologia de desenvolvimento do trabalho**.**



Etapa 1 – Pré-processamento. Embora Christhie (2015) tenha realizado muitas tarefas de pré-processamento nas avaliações extraídas do TripAdvisor, o resultado da análise de sentimentos baseada em aspectos ainda apresentou muitos ruídos – como aspectos sem nenhum sentido ou em branco –, tornando necessário outros procedimentos de limpeza e tratamento dos dados. Os principais procedimentos realizados estão apresentados no Apêndice B.

Etapa 2 – Transformação. A transformação foi caracterizada pela adequação e formatação dos dados para que eles pudessem ser utilizados no Scikit-learn. Um exemplo de transformação é a criação de uma matriz de dados numérica a partir do conjunto de dados.

Etapa 3 – Mineração de Dados. Após o conjunto de dados ter sido tratado e transformado, eles agora podem ser analisados. Os algoritmos escolhidos para a análise foram o K-Means, DBSCAN e o LDA. Os motivos que levaram a escolha deles são os que se seguem:

A. **K-Means**: é o algoritmo clássico de *clustering*. Apesar de não ser dos mais recentes (1957), ainda é amplamente utilizado e possui um método de particionamento simples e objetivo: os objetos são definidos ao grupo *mais próximo*, de acordo com alguma medida de distância. Detalhes sobre o K-Means será apresentado na seção 2.5.

B. **DBSCAN**: é um dos algoritmos mais citados na literatura (JIN; LIN, 2011; MICROSOFT, 2016), além de trazer vantagens significativas para o processo de *clustering*, como a não especificação, *a priori*, do número de agrupamentos que se deseja obter. A seção 2.6 descreve melhor o DBSCAN.
C. **LDA**: algoritmo de modelagem probabilística de tópicos que tem apresentado resultados expressivos em tarefas de mineração de texto, tais como a sumarização, categorização e agrupamento (LU; MEI; ZHAI, 2011). Abordado mais detalhadamente na seção 2.7.1.

Etapa 4 – Interpretação e avaliação. A fase de interpretação e avaliação se propuseram em realizar duas principais tarefas: (1) a avaliação dos agrupamentos formados utilizando o coeficiente de silhueta e a (2) interpretação da *contribuição dos aspectos nos agrupamentos formados*.

Sobre o primeiro, o resultado da avaliação forneceu quantitativamente as melhores configurações de parâmetros para cada algoritmo, utilizando para isso, o coeficiente de silhueta (seção 2.4.3). Através desta medida, um questionamento muito comum em problemas de aprendizagem não supervisionada pode ser explorado: em uma análise de agrupamentos, qual é a quantidade de grupos ideal para um determinado conjunto de dados?

Para a segunda tarefa, a interpretação da *contribuição dos aspectos nos agrupamentos formados* é tratada como uma das vertentes de maior relevância deste trabalho e possui uma seção (4.6) inteiramente responsável por detalhá-la. Em linhas gerais, a *contribuição dos aspectos nos agrupamentos* pode ser compreendida como uma interpretação realizada pelo autor para identificar o quanto um aspecto é importante em um determinado grupo de aspectos.

Aliás, devido ao fato de que diferentes parâmetros de configuração possuem uma tendência natural de gerar resultados também diferentes, as etapas de mineração de dados (3) e avaliação (4) ocorreram inúmeras vezes, justificando a relação encadeada da Figura 12.

Etapa 5 – Apresentação. Os resultados obtidos são apresentados em uma ferramenta web que permite ao usuário a visualização dos agrupamentos de aspectos, bem como o quão um aspecto contribui para a formação dos agrupamentos gerados – conforme dito na etapa anterior. Detalhes sobre a ferramenta serão apresentadas na seção 4.5.

A metodologia de desenvolvimento deste trabalho foi inspirada no processo de KDD e modelada de acordo com a necessidade da solução, fato que culminou na inserção e remoção de algumas fases do processo padrão. Os ajustes realizados estão listados a seguir:

- **Fase de seleção**: foi removida porque não houve a necessidade de se realizar a seleção e/ou definição das fontes de dados nas quais as variáveis de interesse estariam presentes, tendo em vista que o trabalho utilizou somente o banco de dados criado por Christhie (2015).
- **Fase de Apresentação**: fase incluída por se tratar de uma etapa importante no cenário de descoberta de conhecimento. Muitos resultados de problemas de mineração de dados acabam por não serem tão úteis em um contexto justamente pela ausência de uma visualização de mais alto nível. Sem uma visualização mais intuitiva, o resultado ainda fica restrito a públicos com conhecimentos mais específicos em ciência de dados e áreas correlatas.

Vale ressaltar ainda que a metodologia acima se preocupou em descrever as etapas realizadas no processo de desenvolvimento **da solução como um todo** e não somente do **módulo de análise de agrupamentos**. Sobre este segundo, aliás, a maneira na qual o processo foi conduzido está descrito nas seções a seguir.

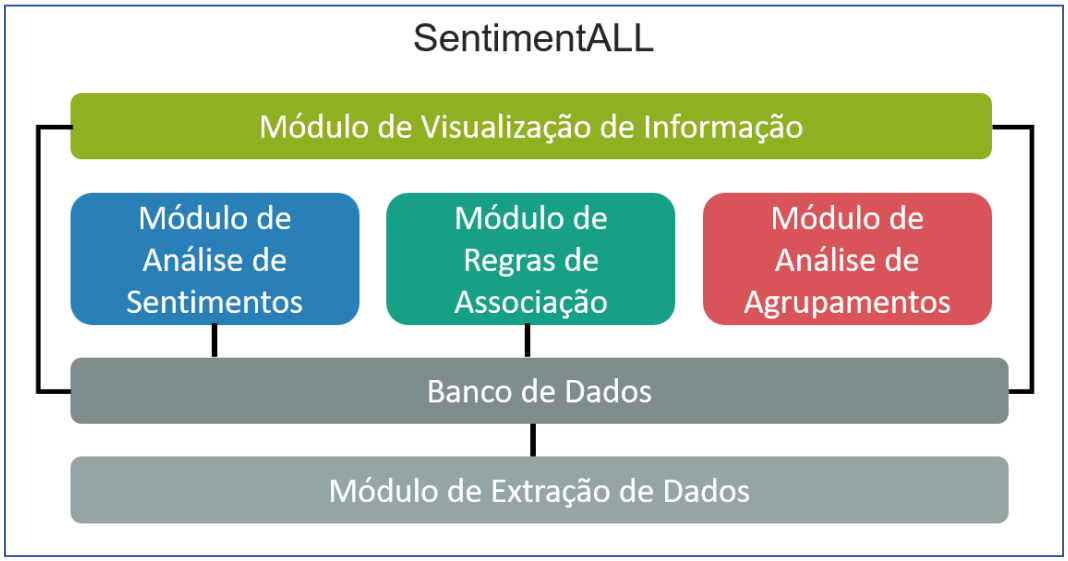
4 RESULTADOS E DISCUSSÃO

Esta seção se preocupa em apresentar os principais resultados do trabalho. Ela aborda assuntos como a visão sistêmica do módulo desenvolvido – com entrada, processamento e saída –, a utilização e funcionalidades do módulo, a definição da *contribuição dos aspectos nos agrupamentos formados*, bem como alguns cenários de teste para demonstrar a solução e gerar discussões.

4.1 ARQUITETURA

Considerando a importância do contexto no qual este trabalho está inserido, o primeiro assunto apresentado nesta seção descreve um breve resumo dos outros módulos[5] (e trabalhos) que compõem o SentimentALL. Sabendo que o SentimentALL possui módulos com funções de extração de dados, mineração de dados e visualização de informação (ROESE, 2016), a sua arquitetura está ilustrada na Figura 13.

Figura 13 – Arquitetura do SentimentALL.



Fonte: adaptado de Roese (2016).

Primeiramente, o *Módulo de Extração de Dados* foi responsável pela recuperação de 1.415.476 avaliações de usuários do TripAdvisor e a sua persistência em uma base de dados (CHRISTHIE, 2015).

[5] Uma descrição mais detalhada dos módulos está descrita no Apêndice C.

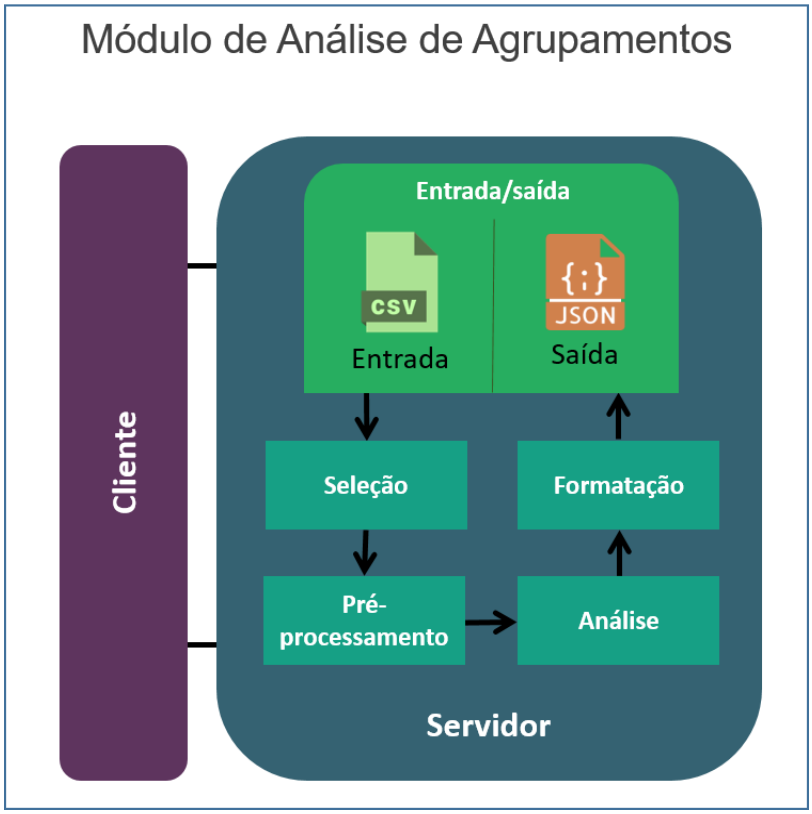
Posteriormente, essas avaliações tiveram seus aspectos identificados e suas polaridades definidas através do *Módulo de Análise de Sentimentos* (CHRISTHIE, 2015). O resultado desta análise foi armazenado na mesma base de dados e utilizado pelos módulos subsequentes.

O *Módulo de Regras de Associação* aplicou o algoritmo *Apriori* para extrair regras associativas entre os aspectos positivos mais frequentes nos comentários (ARAÚJO, 2016; SCHMITZ, 2015).

O *Módulo de Visualização de Informação* se preocupou em apresentar as informações do banco de dados de maneira intuitiva para o usuário, com gráficos e geradores de taxonomia (ROESE, 2016).

Já o *Módulo de Análise de Agrupamentos*, em síntese, teve por objetivo o desenvolvimento de uma ferramenta web que aplica a análise de agrupamentos nas *avaliações baseadas nos aspectos* e apresenta os resultados em gráficos. A sua arquitetura está apresentada na Figura 14.

Figura 14 – Arquitetura do módulo de análise de agrupamentos.



Praticamente todo o processamento é realizado no servidor, onde um arquivo CSV (*Comma-separated values*) que é enviado pelo usuário contêm os dados que serão utilizados como **entrada** para o módulo.

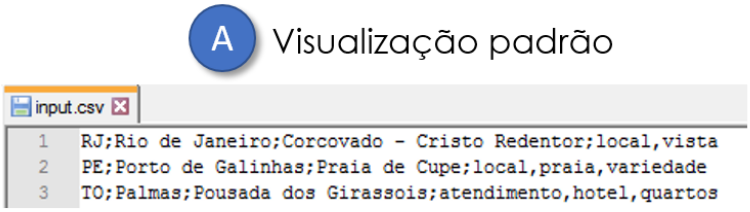
Primeiramente, o *Componente de Seleção* se encarrega de recuperar os subconjuntos de dados dos níveis de *estado*, *cidade* e *objeto* e os passam para o Componente de Pré-processamento. No *Componente Pré-processamento* é realizado a transformação do subconjunto em uma matriz numérica n-dimensional para que ela possa ser utilizada no *Componente de Análise*. Os resultados obtidos pelo Componente de Análise são então repassados ao *Componente de Formatação*, que armazena em um arquivo JSON uma estrutura de dados em árvore (**saída**) pronta para ser apresentada ao usuário em forma de gráfico do tipo *TreeMap*.

As próximas seções apresentam o módulo de análise de agrupamentos (chamado a partir daqui apenas de *módulo*) utilizando uma abordagem sistêmica – com entrada, processamento e saída – e evidenciando cada um dos componentes citados acima. Uma visão macro do processo pode ser visualizado no diagrama de sequência do Apêndice D.

4.2 ENTRADA

A entrada do módulo se trata de um arquivo CSV contendo o resultado de uma análise de sentimentos baseada em aspectos e estruturado nesta ordem: (1) a avaliação propriamente dita, (2) o destino que a avaliação se refere (ou *objeto*), (3) o estado, (4) cidade e (5) o conjunto de aspectos presentes na avaliação separados por vírgula, tal como ilustrado na Figura 15.

Figura 15 – Arquivo CSV para demonstrar a estrutura de dados de entrada.



B Visualização em tabela

Estado	Cidade	Objeto	Aspectos
RJ	Rio de Janeiro	Corcovado – Cristo Redentor	Local, vista
PE	Porto de Galinhas	Praia de Cupe	Local, praia, variedade
TO	Palmas	Pousada dos Girassóis	Atendimento, hotel, quartos

Um conjunto de dados no formato CSV geralmente têm suas amostras separadas por quebra de linha e suas características (variáveis) separadas por vírgula, no entanto, a vírgula

pode ser substituída por ponto e vírgula dependendo do *separador de lista* do território, justificando a estrutura da Figura 15-A.

É importante destacar ainda que apesar desta pesquisa ter utilizado somente avaliações de destinos turísticos, outras avaliações de qualquer natureza poderiam ter sido utilizadas como entrada para o módulo, desde que, obrigatoriamente, estivessem armazenadas em um arquivo CSV separado por ponto e vírgula e estruturado em função do objeto, seguindo o modelo:

```
estado ; cidade ; objeto ; lista de aspectos separadas por vírgula
```

4.3 PROCESSAMENTO

4.3.1 Componente de Seleção

Considerando a estrutura que o arquivo CSV de entrada deve ter, o processamento adota uma representação hierárquica de três níveis: *estado*, *cidade* e *objeto*, da seguinte maneira:

1. Primeiramente, todas as amostras de cada um dos *estados* disponíveis são selecionadas; depois
2. Todas as amostras de cada *cidade* e *estado* também são selecionadas; e, por fim
3. Todas as amostras de cada *objeto*, *cidade* e *estado* são selecionadas.

O algoritmo a seguir detalha o funcionamento deste processo:

Algoritmo 1 – Seleção dos subconjuntos de dados.

Algoritmo: Seleção(A)

1. $S = \{\}$
2. Para cada $estado \in Estados(A)$ faça
3. $\quad S_{estado} = \{a \in A \mid a.estado == estado\}$
4. $\quad$ Para cada $cidade \in Cidades(estado)$ faça
5. $\quad\quad S_{(estado,cidade)} = \{a \in A \mid a.cidade == cidade\}$
6. $\quad\quad$ Para cada $objeto \in Objetos(cidade)$ faça
7. $\quad\quad\quad S_{(estado,cidade,objeto)} = \{a \in A \mid a.objeto == objeto\}$
8. Retorne S

Onde:

- A é o conjunto de avaliações na qual cada elemento é uma tupla contendo os atributos de uma avaliação: $(estado, cidade, objeto, listaDeAspectos)$;
- S_{estado} é um subconjunto de A para cada estado;
- $S_{(estado,cidade)}$ é um subconjunto de A para cada cidade de um estado;
- $S_{(estado,cidade,objeto)}$ é um subconjunto de A para cada objeto, de cada cidade, de cada estado;
- $Estados(A)$ é uma função que retorna os estados presentes em A;
- $Cidades(estado)$ é uma função que retorna todas as cidades de um estado; e
- $Objetos(cidade)$ é uma função que retorna todos os objetos de uma cidade.

Para exemplificar, considere o conjunto de dados apresentado no Quadro 3:

Quadro 3 – Conjunto de dados de exemplo para demonstrar o algoritmo *Seleção(S)*.

#	Estado	Cidade	Objeto	Lista de aspectos
1	GO	Caldas novas	diRoma Acqua Park	ambiente, piscina, brinquedos
2	GO	Goiânia	Castro's Park Hotel	localização, espaço
3	TO	Palmas	Hotel dos Girassóis	atendimento, café da manhã
4	TO	Palmas	Hotel dos Girassóis	localização
5	TO	Palmas	Praia da Graciosa	clima, vista
6	TO	Paraíso do Tocantins	Restaurante Ecológico	comida, ambiente, clima

Seja A o conjunto de todas as avaliações a do Quadro 3, o processamento hierárquico geraria os subconjuntos:

1. Subconjunto a nível do estado:
   a. $S_{go} = \{a_1, a_2\}$
   b. $S_{to} = \{a_3, a_4, a_5, a_6\}$

2. Subconjuntos a nível de cidade:
   a. $S_{go,caldas\ novas} = \{a_1\}$
   b. $S_{go,goiania} = \{a_2\}$
   c. $S_{to,palmas} = \{a_3, a_4, a_5\}$
   d. $S_{to,paraíso\ do\ tocantins} = \{a_6\}$

3. Subconjuntos a nível de objeto:
   a. $S_{go,caldas\ novas,diroma\ acqua\ park} = \{a_1\}$
   b. $S_{go,goiania,castro's\ park\ hotel} = \{a_2\}$
   c. $S_{to,palmas,hotel\ dos\ girassóis} = \{a_3, a_4\}$
   d. $S_{to,palmas,praia\ da\ graciosa} = \{a_5\}$
   e. $S_{to,paraíso\ do\ tocantins,restaurante\ ecológico} = \{a_6\}$

A definição e seleção destes subconjuntos de estados, cidades e objetos é justamente a função do *Componente de Seleção*, que ainda repassa estes subconjuntos ao Componente de Pré-processamento.

Cabe a ressalva de que os subconjuntos são simplesmente estruturas de dados em Python armazenadas em memória e não são gravados em arquivo ou algo similar, apenas são passados por parâmetro para o Componente de Pré-processamento.

4.3.2 Componente de Pré-processamento

De posse dos subconjuntos de cada estado, cidade e objeto, o Componente de Pré-processamento tem a função de transformar estes subconjuntos em estruturas de dados numéricas para que os algoritmos do Componente de Análise possam ser aplicados.

O Componente de Pré-processamento pode ainda ser dividido em duas etapas: a extração da lista de aspectos e a atribuição binária dos aspectos dimensionados.

4.3.2.1 Extração da lista de aspectos

A extração da lista aspectos consiste em extrair todos os aspectos presentes em cada um dos subconjuntos de dados. O algoritmo a seguir detalha o funcionamento deste processo:

Algoritmo 2 – Extração da lista de aspectos.

Algoritmo: Aspectos(S)

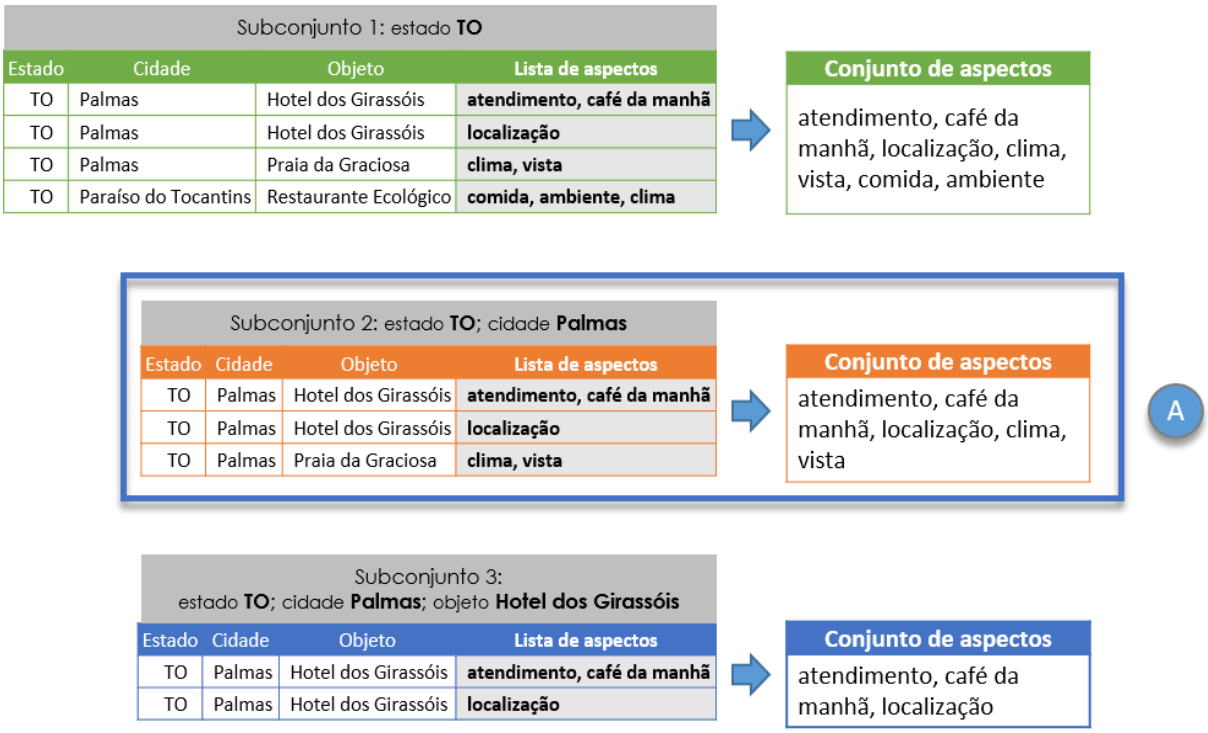
1. $L = \{\}$
2. Para cada $a \in S$ faça
3. $\quad L \leftarrow a.listaDeAspectos$
4. Retorne L

Onde:

- S é um conjunto de avaliações (um subconjunto resultante de $Seleção(A)$); e
- L é um conjunto com a lista de aspectos presentes em S.

O algoritmo $Aspectos()$ é aplicado aos conjuntos definidos pelo algoritmo $Seleção()$. Para demonstrar esse processo, a Figura 16 ilustra a extração das listas de aspectos para avaliações do *estado TO* (nos níveis de estado, cidade e objeto).

Figura 16 – Exemplo de extração da lista de aspectos de subconjuntos.



Na Figura 16, há três subconjuntos (um para cada nível) com suas respectivas listas de aspectos. A lista contém o conjunto de todos os aspectos presentes nas amostras. Por exemplo: o subconjunto "*estado TO; cidade Palmas*" tem os aspectos *atendimento*, *café da manhã*, *localização*, *clima* e *vista*, indicado na Figura 16-A.

4.3.2.2 Atribuição binária dos aspectos dimensionados

De posse da lista de aspectos de um determinado subconjunto de avaliações, uma matriz em que cada elemento da lista se torna uma coluna (ou dimensão) é então criada. O algoritmo a seguir detalha o funcionamento deste processo:

Algoritmo 3 – Atribuição binária dos aspectos dimensionados.

Algoritmo: AtribuiçãoBinária(S)

1. $L = []$
2. $aspectos = Aspectos(S)$
3. Para cada $s \in S$ faça
4. Para cada $a \in aspectos$ faça
5. $linha_a = (a \in s.listaDeAspectos ? 1 : 0)$
6. $L \leftarrow linha$
7. Retorne L

Onde:

- S é um conjunto de avaliações (um subconjunto resultante de $Seleção(A)$); e
- L é uma matriz contendo $|S|$ linhas e $|Aspectos(S)|$ colunas. Os valores das células dessa matriz contêm 1 ou 0, conforme os aspectos de cada avaliação presente em S.

Para demonstrar, a Figura 17 ilustra a criação da matriz resultante a partir do uso do algoritmo $AtribuiçãoBinária()$ aplicado no subconjunto de avaliações da cidade de Palmas, no estado do Tocantins (TO) e indicado na Figura 16-A.

Figura 17 – Exemplo do processo de atribuição binária.

Os aspectos identificados na Figura 17-A (*atendimento*, *café da manhã* e *localização*) e indicados pela Figura 17-B são transportados para a nova matriz de dados (Figura 17-C). Para ela, é atribuída o valor "1" caso o aspecto esteja presente na avaliação em questão ou "0" caso esteja ausente (Figura 17-D). Portanto, a primeira avaliação será representa pelo valor "1" em todas as colunas, enquanto a segunda amostra, terá o valor "1" somente na última coluna, para o aspecto *localização.*

4.4 SAÍDA

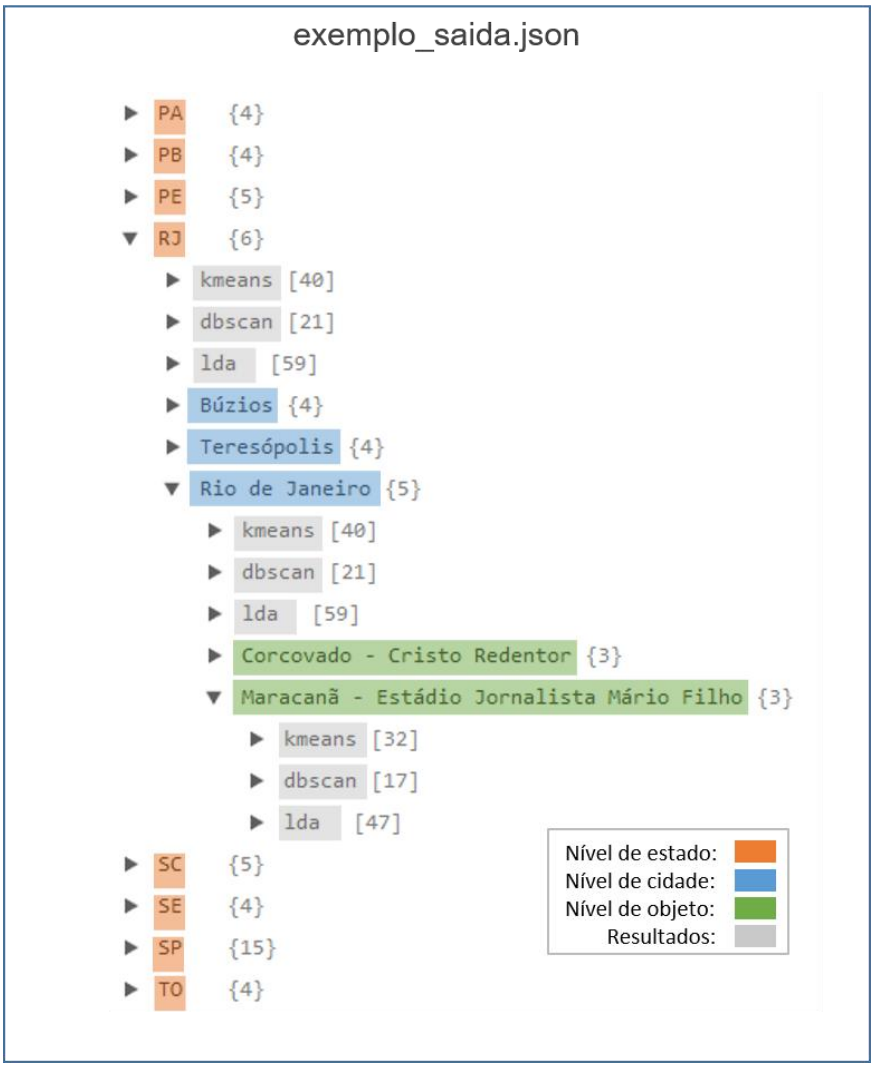
Após as matrizes de dados terem sido criadas, elas agora são passadas aos Componentes de Análise e Formatação para, respectivamente, executar os algoritmos nos dados e preparar o arquivo de saída que será apresentado ao usuário.

4.4.1 Componentes de Análise e Formatação

O Componente de Análise executa os algoritmos nas matrizes de dados e retornam os resultados obtidos. O problema é que estes resultados, da forma como são gerados, ainda são de difícil visualização para o usuário final, tornando necessário que uma formatação seja realizada. Logo, o Componente de Formatação tem a função de preparar uma estrutura de dados que facilite a apresentação e interpretação dos resultados dos algoritmos para usuários não técnicos.

Esta estrutura de dados é armazenada em um arquivo JSON e preserva a representação hierárquica de *estado*, *cidade* e *objeto* mencionada anteriormente. Um exemplo de arquivo JSON visualizado em árvore está apresentado na Figura 18.

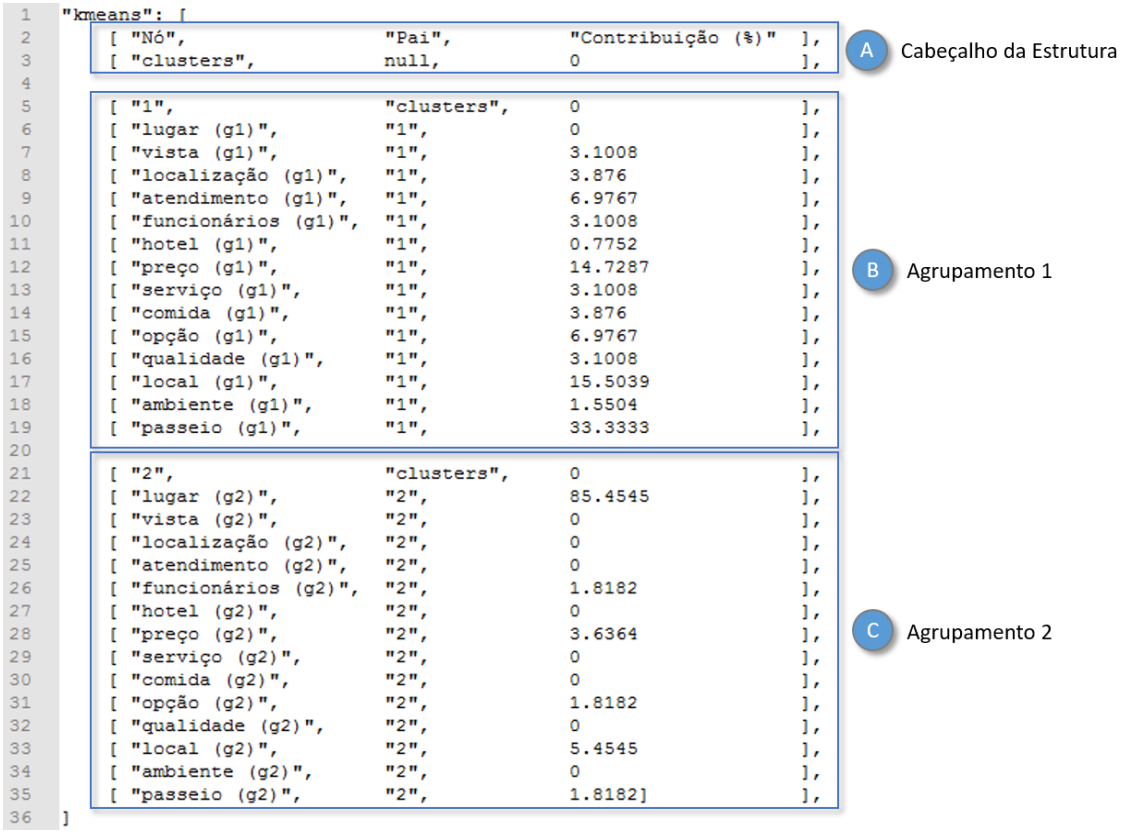
Figura 18 – Visualização em árvore de arquivo JSON de exemplo.



Os níveis de hierarquia estão ilustrados de acordo com a cor: os *estados* em laranja, as *cidades* em azul e os *objetos* na cor verde. Em cinza, estão os resultados dos algoritmos e, como pode ser observado, cada nível hierárquico possui o seu próprio conjunto de chaves (e resultados). O objetivo desta abordagem é evitar que usuário tenha que esperar por qualquer tipo de processamento no lado cliente, permitindo assim, a visualização instantânea dos resultados para qualquer nível desejado.

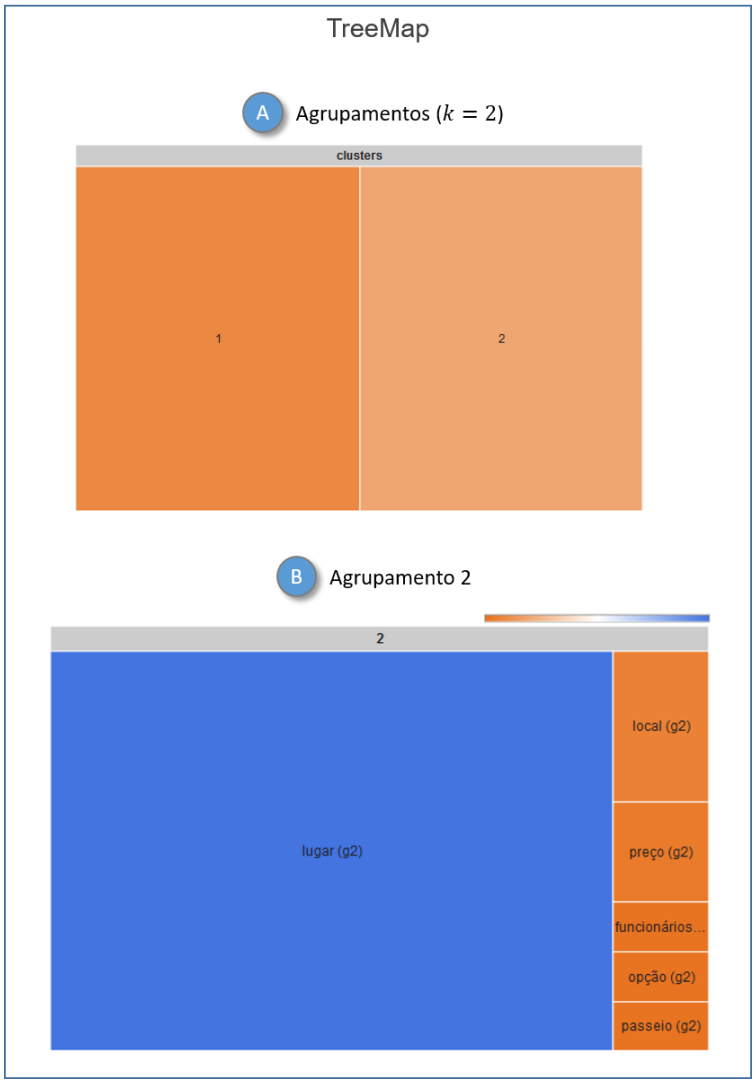
As chaves indicadas em cinza contêm uma estrutura de dados em *DataTable* utilizada para formar a visualização dos resultados nos gráficos *TreeMap*. Para exemplificar e ilustrar, considere a Figura 19 e a Figura 20, que representam o *DataTable* e a sua visualização em *TreeMap*, respectivamente, de uma análise utilizando o algoritmo K-Means e com $k = 2$.

Figura 19 – Exemplo de DataTable.



De acordo com a Figura 19 e considerando que o *DataTable* possui uma representação hierárquica de árvore, em A está indicado os elementos que compõem o cabeçalho e o nó raiz da árvore. Em B, tem-se o primeiro agrupamento gerado e em C os elementos do segundo agrupamento. Este *DataTable* é então utilizado pela biblioteca do Google Chart Tools (vide seção 3.1) para gerar o gráfico apresentado na Figura 20.

Figura 20 – Visualização em *TreeMap* a partir de um *DataTable* de exemplo.

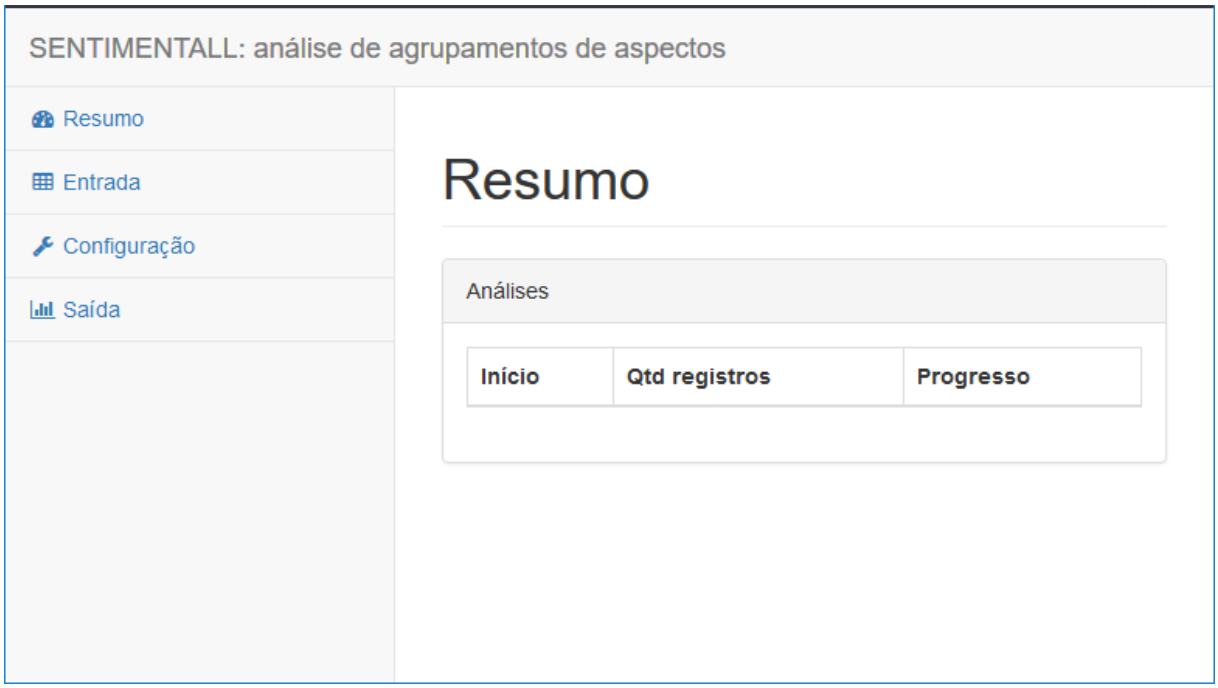


A Figura 20-A indica os dois agrupamentos gerados e a Figura 20-B, a composição do segundo agrupamento. Ambas utilizam a representação em *TreeMap*, que será explicado melhor na seção 4.5.3.

4.5 FUNCIONALIDADES DO MÓDULO

As seções anteriores abordaram procedimentos técnicos desde a leitura do arquivo CSV até a geração do JSON de saída. Esta seção também apresenta o módulo, porém, desta vez utilizando uma abordagem mais próxima da perspectiva de usuário, ou seja, se preocupando mais em apresentar as telas e as principais funcionalidades.

Figura 21 – Tela inicial da aplicação web.

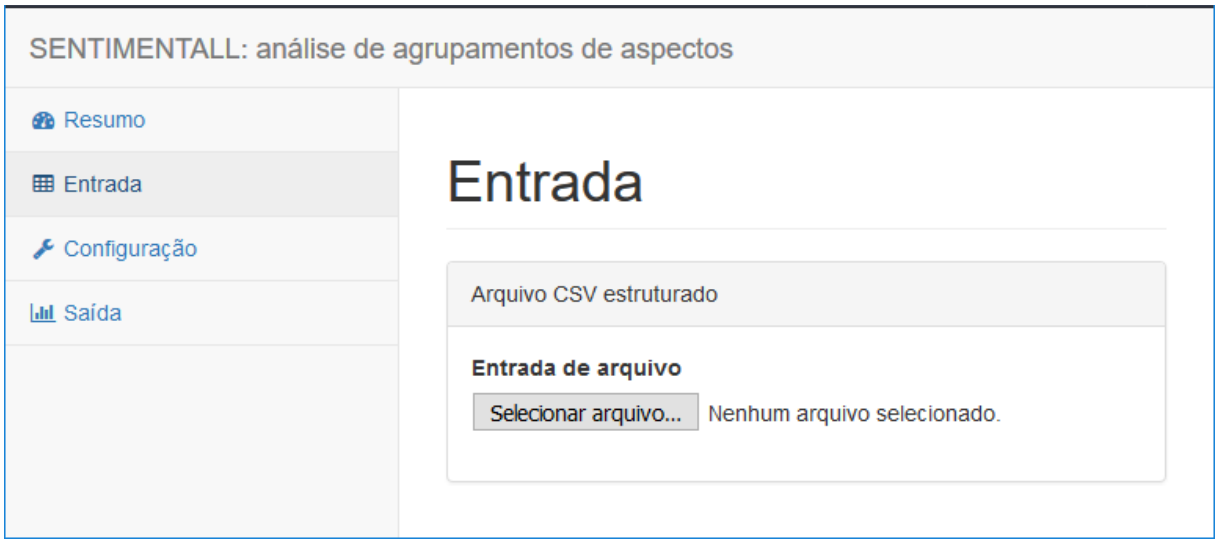


A Figura 21 apresenta a tela inicial do módulo. À esquerda, o menu principal dá acesso as telas de *Entrada*, *Configuração* e *Saída* onde, respectivamente, três grandes funcionalidades do módulo estão definidas: (1) o *envio do arquivo CSV de entrada*, a (2) *configuração e execução da análise de agrupamentos* e a (3) *visualização dos resultados*. As seções a seguir descrevem detalhadamente cada uma delas.

4.5.1 Envio do arquivo CSV de entrada

O envio do arquivo CSV de entrada é caracterizado basicamente pelo *upload* do arquivo na tela de entrada, apresentado na Figura 22.

Figura 22 – Tela de *upload* do arquivo CSV da aplicação web.

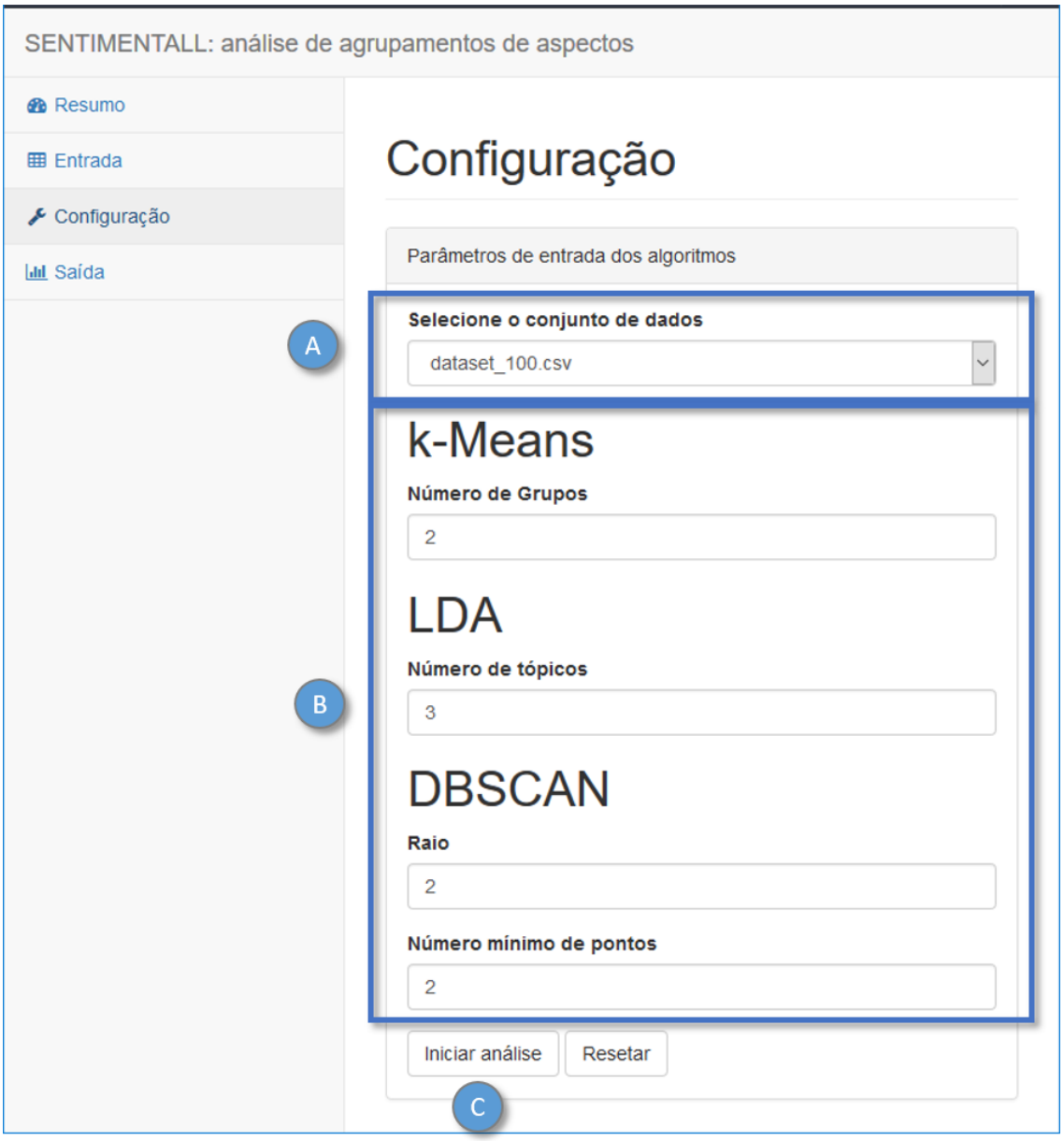


Por meio do botão *Selecionar arquivo*, o usuário pode escolher e enviar um arquivo CSV. Lembrando que o arquivo deve estar estruturado exatamente como definido na seção 4.2.

4.5.2 Configuração e execução da análise de agrupamentos

A configuração consiste na definição dos parâmetros dos algoritmos e na seleção do conjunto de dados (CSV anteriormente enviado) que se deseja aplicar a análise. A tela de configuração está apresentada na Figura 23.

Figura 23 – Tela de configuração da aplicação web.



Considerando a Figura 23, o conjunto de dados é selecionado em A e os parâmetros dos algoritmos, em B. A partir do momento em que o usuário aciona o início do processo pelo botão *Iniciar análise* (C), todo o processamento explicado na seção 4.3 é executado.

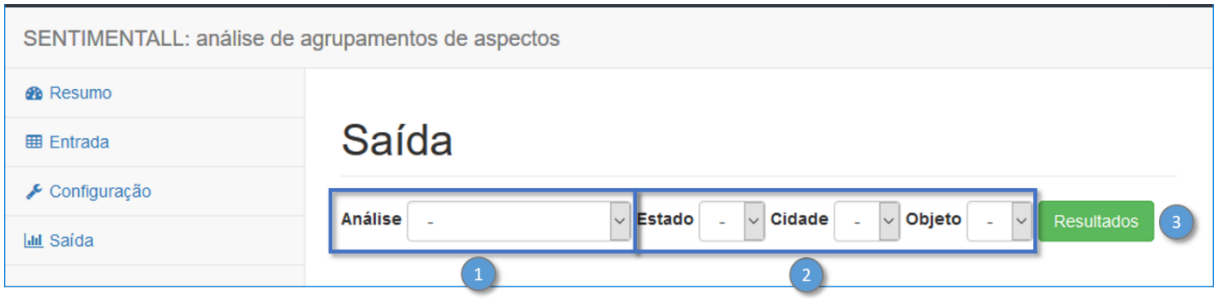
Ao final, o componente de formatação utiliza os resultados obtidos na análise para gerar um arquivo estruturado hierarquicamente (vide seção 4.4) e pronto para ser exibido ao usuário. O resultado de cada análise é armazenado em um arquivo JSON no servidor, ficando disponível para ser acessado na tela de saída, conforme apresentado na seção a seguir.

4.5.3 Visualização dos resultados

Os gráficos *TreeMap* apresentam uma árvore disposta no formato de um mapa navegável. Cada nó da árvore é representado por um retângulo, dimensionado e colorido de acordo com os valores intrínsecos àquele nó.

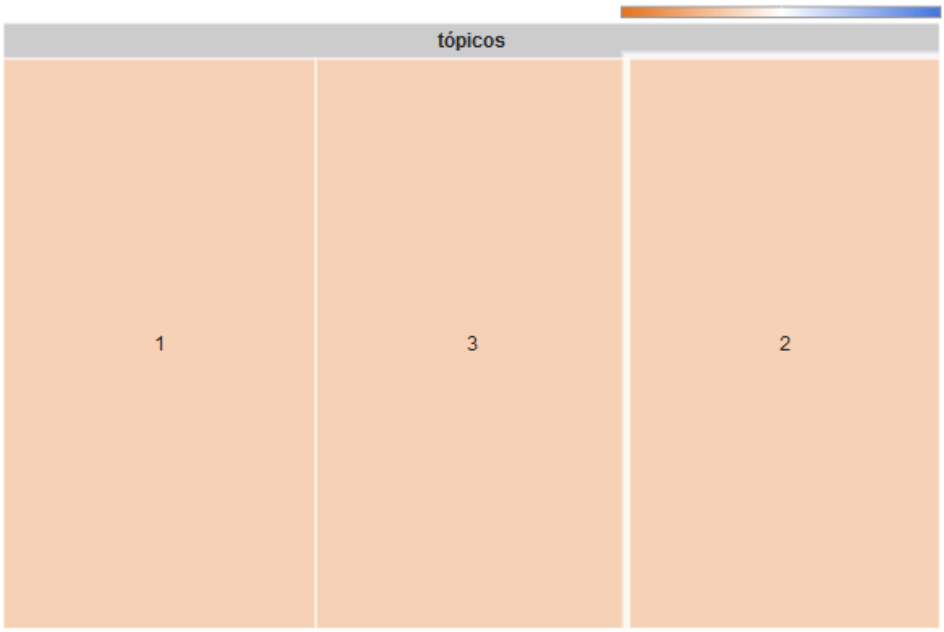
Para visualizar o gráfico, o usuário deverá primeiramente selecionar alguma análise já finalizada (Figura 24-1) e o nível hierárquico (estado, cidade ou objeto) (Figura 24-2).

Figura 24 – Tela de visualização dos resultados da aplicação web.



Feito isso, após clicar no botão *Resultados* (Figura 24-3), os agrupamentos gerados **para o nível selecionado** são exibidos, conforme apresentado no exemplo da Figura 25.

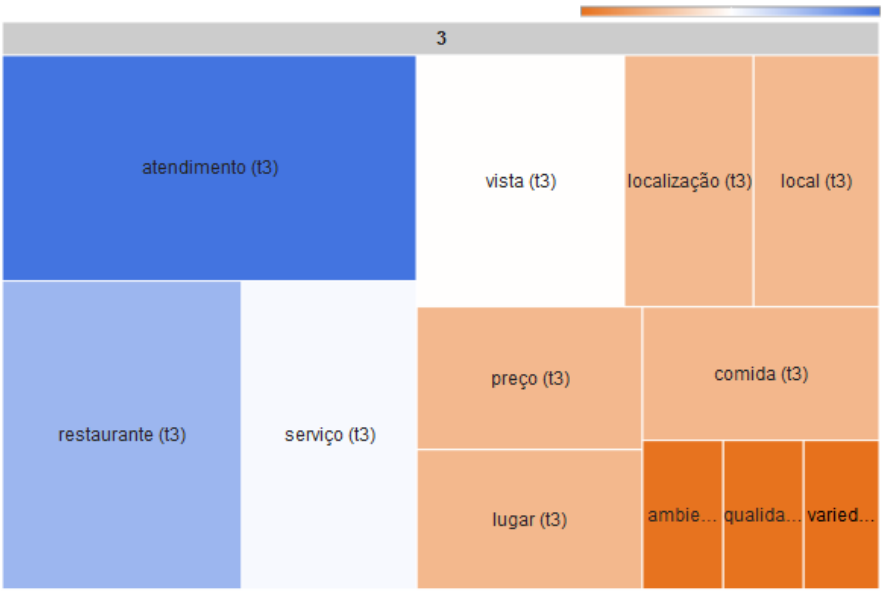
Figura 25 – Raiz de gráfico *TreeMap* exibido na tela de visualização da aplicação web.



A Figura 25 apresenta o primeiro nível da árvore (e do mapa) para o exemplo de uma análise de agrupamentos que obteve três grupos. O usuário pode acessar um agrupamento

clicando com o botão esquerdo do *mouse* no grupo de interesse. Caso se desejasse acessar o terceiro grupo, por exemplo, o gráfico exibido seria o que está apresentado na Figura 26.

Figura 26 – Gráfico *TreeMap* exibido na tela de visualização da aplicação web.



De acordo com a Figura 26, os nós são dispostos na área do mapa por meio de retângulos. O tamanho e a cor dos retângulos são atribuídos em relação a todos os outros nós: os retângulos de maior área e com a coloração mais próxima do azul são os nós mais *representativos da estrutura*.

A relevância dos nós na estrutura de acordo com a cor e tamanho dos retângulos, na verdade, só passam a fazer sentido quando um dos conceitos mais importantes deste trabalho é esclarecido: a *contribuição dos aspectos na formação dos agrupamentos*. A seção a seguir detalha como esse conceito foi idealizado.

4.6 DEFINIÇÃO DA CONTRIBUIÇÃO DOS ASPECTOS NOS AGRUPAMENTOS

A seção anterior apresentou o *TreeMap* como um recurso gráfico para tornar a percepção da contribuição dos aspectos mais evidente, onde os retângulos maiores representam os elementos mais significativos do seu grupo. Contudo, é muito importante compreender também a maneira pela qual essa informação de interesse foi concebida.

A *contribuição dos aspectos nos agrupamentos* diz respeito a uma interpretação do autor para determinar o quanto uma característica do conjunto de dados é relevante para um

dado agrupamento, isto é, no contexto do SentimentALL, o quão cada um dos aspectos é representativo para o seu grupo em comparação com os outros aspectos daquele grupo.

Neste sentido, ainda tendo como exemplo o gráfico da Figura 26, o *atendimento* é o aspecto que melhor contribui para a formação do agrupamento, já *ambiente*, *qualidade* e *variedade*, são os que menos contribuem.

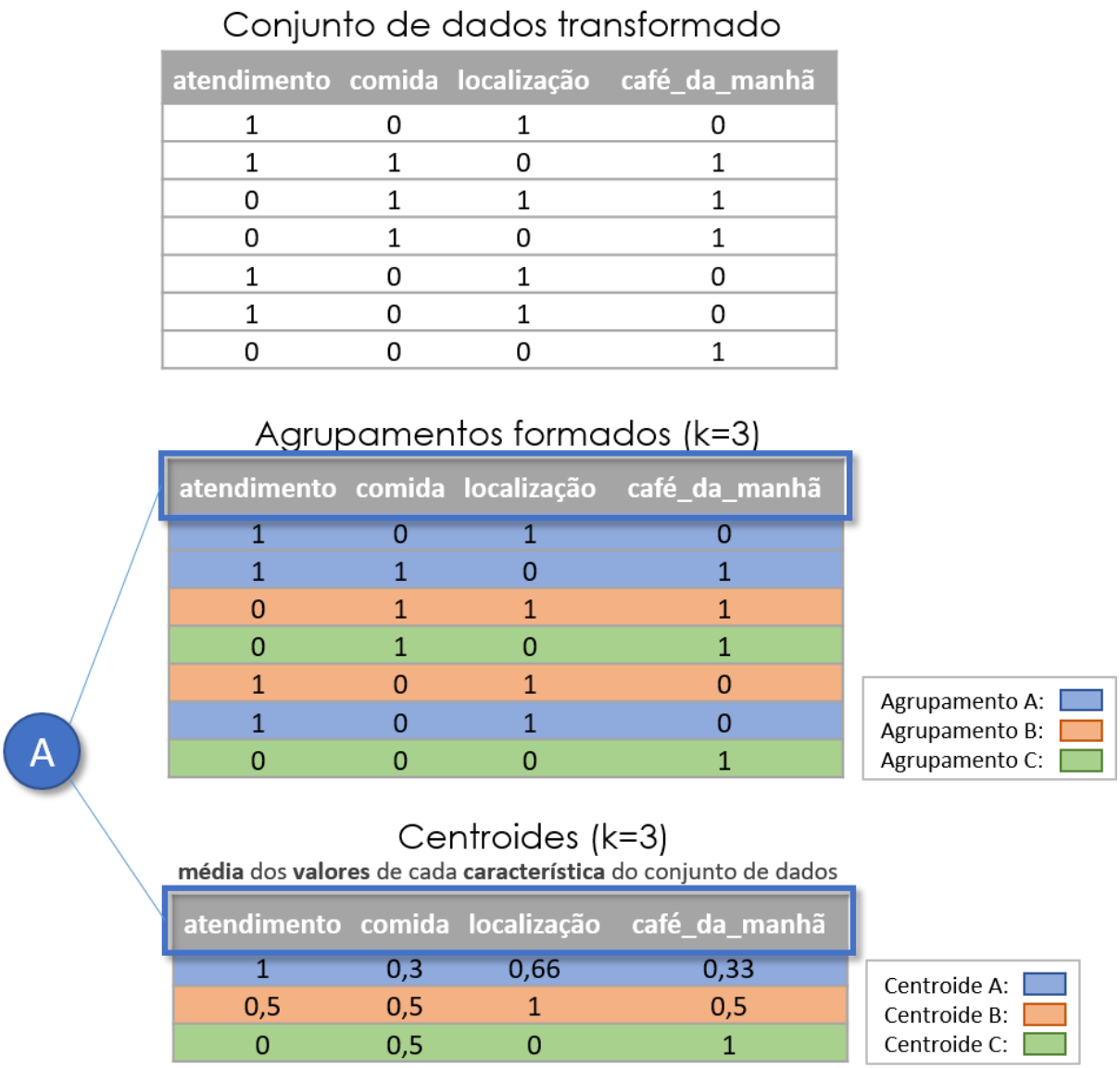
As seções a seguir descrevem como essa informação foi concebida para cada um dos algoritmos.

4.6.1 Intepretação para o K-Means

A seção 2.5 introduziu o conceito de *centroide*. A definição mais comum de um centroide no cenário da mineração de dados é: um ponto formado pela média das características de um conjunto de dados.

Pensando de uma maneira mais intuitiva, um centroide pode ser entendido como um elemento que indica o ponto central de um agrupamento de dados. Este ponto central é semelhante a todas a outras amostras do conjunto de dados, isto é, possui o **mesmo número de dimensões**, as mesmas características e tipo, mas não necessariamente os mesmos valores – já que os valores do centroide são definidos como a média de **cada característica** de um mesmo grupo. A Figura 27 ajuda a compreender melhor este conceito.

Figura 27 – Exemplo para auxiliar a demonstração da interpretação para o K-Means.



A Figura 27 apresenta um exemplo do que foi explicado acima. Nele, uma análise de agrupamentos é realizada com o objetivo se obter 3 grupos. Os centroides dos três agrupamentos formados são uma representação fiel da estrutura de dados (Figura 27-A) e o valor de cada característica (aspecto) é atribuído com a média das amostras da característica e do grupo em questão. Por exemplo, para a característica *localização* do *Agrupamento A* (em azul), é atribuído o valor da operação $\frac{1+0+1}{3} \cong 0{,}66$.

Trazendo para o contexto do trabalho, onde cada característica do conjunto de dados é um aspecto, por meio dos centroides dos agrupamentos se torna possível identificar o quão cada um deles contribui e são relevantes para o seu agrupamento que, em outras palavras, significa

dizer que o centroide carrega uma evidência do quanto um aspecto representa/influencia/impacta em cada centroide, ou seja, em cada agrupamento.

4.6.2 Intepretação para o DBSCAN

A intepretação da contribuição dos aspectos para o DBSCAN também utiliza o conceito de centroide, porém, como é bem sabido, algoritmos baseados em densidade não trabalham diretamente com ele, aliás, a própria implementação do DBSCAN na biblioteca Sckit-learn não disponibiliza essa informação.

No entanto, esse empecilho foi resolvido da seguinte maneira: a partir da definição dos grupos de cada amostra, foi possível realizar o cálculo do centroide exatamente como foi é feito para o K-Means, isto é, os centroides foram calculados a partir da média dos elementos contidos nos próprios agrupamentos formados.

4.6.3 Interpretação para o LDA

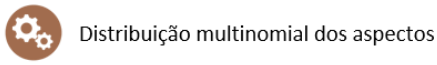
O LDA, diferentemente dos outros algoritmos utilizados, já fornece nativamente a probabilidade de cada aspecto pertencer a um grupo. Para entender como isso funciona, é preciso recapitular o que foi apresentado na seção 2.7.1: o *Latent Dirichlet Allocation* é um modelo de tópicos que gera grupos (chamados de tópicos) baseados na frequência de palavras de um conjunto de documentos.

Trazendo esses conceitos para o contexto de aplicação deste trabalho, o *conjunto de documentos* se refere ao conjunto de avaliações de destinos turísticos e a *frequência de palavras*, a atribuição binária dos aspectos dimensionados (vide seção 4.3.2.2).

Sabendo que o LDA já fornece a probabilidade de cada aspecto pertencer a um grupo a partir da distribuição multinomial dos seus aspectos, a análise do conjunto de dados da Figura 28-A – na qual se deseja obter dois grupos – resultaria na probabilidade de cada um dos quatro aspectos pertencer aos dois agrupamentos formados, conforme ilustrado na Figura 28.

Figura 28 – Exemplo para auxiliar na demonstração da interpretação para o LDA.

atendimento	comida	localização	café_da_manhã
1	0	1	0
1	1	0	1
0	1	1	1
0	1	0	1
1	0	1	0
1	0	1	0
0	0	0	1
0	1	1	1
0	1	0	0
1	0	1	0



Aspecto	Probabilidade de pertencer ao Tópico A	Probabilidade de pertencer ao Tópico B
atendimento	0,78	4,44
café_da_manhã	2,79	0,81
comida	2,92	0,69
localização	2,11	2,27

A partir da Figura 28, pode ser notado que *comida* é o aspecto mais significativo no T*ópico A* (probabilidade de pertencer ao tópico é 2,92%), enquanto o aspecto *atendimento* é o que melhor contribui na formação do *Tópico B* (probabilidade de pertencer ao tópico é 4,44%).

4.6.4 Percentual de contribuição do aspecto

Cada procedimento realizado acima resultou em um valor que representa a contribuição do aspecto no agrupamento. Contudo, é interessante saber também o quanto aquele valor é representativo em comparação aos outros, isto é, o percentual de representatividade de um aspecto em relação aos outros aspectos de um mesmo agrupamento.

Deste modo, o percentual de contribuição dos aspectos é calculado a partir do algoritmo a seguir.

Algoritmo 4 – Definição do percentual de contribuição do aspecto.

Algoritmo: ContribuiçãoAspecto(C)

1. $L = []$
2. Para cada $c \in C$ faça

3. $contribuicao$ = []
4. Para cada $aspecto \in c$ faça
5. $contribuicao[aspecto] = \frac{v}{soma(c)} * 100$
6. $L \leftarrow contribuicao$
7. Retorne L

Onde:

- C são os centroides ou a distribuição multinomial (LDA) dos agrupamentos; e
- L é uma matriz contendo $|L|$ linhas e $|C|$ colunas. Os valores das células dessa matriz contêm o percentual de contribuição de um aspecto em relação aos outros aspectos do mesmo grupo/centroide em C.

A partir deste algoritmo, os valores da contribuição dos aspectos definidos para cada algoritmo são ponderados e apresentados por um percentual que indica o quão um aspecto é representativo para o seu agrupamento em relação aos outros aspectos do mesmo agrupamento.

4.7 CENÁRIOS DE TESTE E DISCUSSÃO

Esta seção tem o objetivo de demonstrar o módulo desenvolvido utilizando alguns cenários de teste. A partir da análise de sentimentos de Christhie (2015), foram selecionadas avaliações em que pelo menos um dos aspectos extraídos estavam entre os 20 mais frequentes e que possuíam polaridade positiva. Aliás, os 20 aspectos positivos mais frequentes estão listados no Quadro 4.

Quadro 4 – 20 aspectos positivos mais frequentes na análise de Christhie (2015).

#	Aspecto	Ocorrências	#	Aspecto	Ocorrências
1	atendimento	820681	11	praia	169.398
2	comida	621040	12	café_da_manhã	165.228
3	lugar	420537	13	vista	163.738
4	ambiente	389801	14	qualidade	151.006
5	hotel	324341	15	serviço	118.401
6	restaurante	298934	16	pratos	102.464
7	localização	276113	17	passeio	101.024
8	preço	250238	18	funcionários	95.872
9	local	233354	19	variedade	91.500
10	opção	200178	20	quartos	90.649

A partir do processo de seleção citado acima, 1.025.944 avaliações baseadas nos aspectos formaram o conjunto de dados que serviu como base para os cenários de teste. É

importante destacar também as especificações da máquina em que os experimentos foram realizados:

- Sistemas Operacional: Windows 10.
- Processador: Intel Core I7-3770 (terceira geração).
- Memória RAM: 16Gb.

Considerando o ambiente descrito acima e o fato de que o processamento hierárquico de estado, cidade e objeto (vide seção 4.3.1) pode gerar centenas de agrupamentos, apenas dois casos foram escolhidos para discussão, conforme apresentado nas seções a seguir.

4.7.1 Cenário de teste com todo o conjunto de dados

O primeiro cenário de teste realizado utilizou os seguintes parâmetros (vide seção 4.5.2):

- **Conjunto de dados:** todas as 1.025.944 avaliações baseadas nos aspectos mencionadas acima.
- **Número de grupos (K-Means):** 2.
- **Número de tópicos (LDA):** 2.
- **Raio (DBSCAN):** 2.
- **Número mínimo de pontos (DBSCAN):** 2.

Os primeiros experimentos resultaram em erros de estouro de memória (classe *MemoryError* do Python) e em muitas vezes no travamento por completo da máquina.

A primeira tentativa de resolução foram configurações do sistema operacional, como o aumento do tamanho da memória virtual e a remoção de alguns recursos gráficos. Apesar destas medidas, porém, o experimento ainda não foi possível de ser realizado.

Após ser notado que o problema acontecia no momento em que o DBSCAN era aplicado nas avaliações do estado do Rio de Janeiro, a segunda tentativa foi variar os parâmetros do algoritmo, principalmente o raio de densidade (*eps*), a quantidade mínima de pontos (*minPts*) e o tipo de algoritmo utilizado (*ball_tree*, *kd-tree* ou *brute*) (SCIKIT-LEARN 0.18.1 DOCUMENTATION, 2016), porém, nenhuma combinação de parâmetros resolveu o problema.

A partir de então, foi necessário tomar algumas decisões de maior impacto para que o processamento pudesse ser realizado: ou a abdicação do uso do DBSCAN[6] para este conjunto de dados em específico ou a remoção dos subconjuntos que apresentavam erro, ou seja, o primeiro caso utilizaria somente os algoritmos K-Means e LDA, enquanto o segundo

[6] Fóruns de discussão na internet indicaram que o problema pode ser fruto de uma baixa qualidade da implementação do algoritmo DBSCAN pela biblioteca do Scikit-learn (STACKOVERFLOW, [s.d.]).

descartaria as avaliações dos estados do Rio de Janeiro e São Paulo (que posteriormente também viria a apresentar erro).

Estas duas soluções contornam o problema, entretanto, é bem sabido que este tipo de método não é nada adequado para problemas de mineração de dados. Aliás, as avaliações dos estados do Rio de Janeiro e São Paulo representam quase a metade de todo o conjunto de dados inicial (437.579) e com certeza influenciam muito nos resultados das análises.

4.7.2 Cenário de teste a nível de estado

Para o experimento a nível de estado, todas as avaliações do estado do Tocantins foram utilizadas, totalizando 2.119 avaliações.

Quadro 5 – Quadro comparativo das avaliações de agrupamentos.

k	n	eps	$minPts$	S_k	S_d	t
2	2	2	2	0,16	*(erro)*	2,55
3	3	3	3	0,18	*(erro)*	4,92
4	4	4	4	0,19	*(erro)*	7,77
5	5	5	5	0,21	*(erro)*	10,26

Onde:

- k é o número de grupos que se deseja obter na análise utilizando o K-Means;
- n é o número de tópicos que se deseja obter na análise utilizando o LDA;
- Eps e m$inPts$ são, respectivamente, o tamanho do raio e número de mínimo de pontos para a análise utilizando o DBSCAN;
- S_k é o coeficiente de silhueta avaliado para o algoritmo K-Means;
- S_d é o coeficiente de silhueta avaliado para o algoritmo DBSCAN; e
- t é o tempo de execução em segundos.

Considerando que o coeficiente de silhueta indica que os melhores agrupamentos são aqueles em que o coeficiente se aproxima de 1, nenhum agrupamento teve uma boa avaliação. A melhor quantidade de grupos identificado nas análises foi *cinco*, com 0,21 de coeficiente, conforme apresentado no Quadro 5.

Outra característica interessante deste cenário de teste foi que a medida que a quantidade de grupos aumentou, o valor do coeficiente também. Este fato indica uma tendência de que quanto maior a quantidade de grupos, melhor é a avaliação da análise de acordo com o coeficiente de silhueta.

Ainda em relação ao Quadro 5, em todas as análises do DBSCAN ocorreram erros e não compete a este trabalho o mérito de discorrer sobre os motivos que o culminaram.

5 CONCLUSÕES

O módulo desenvolvido neste trabalho resultou em ferramenta prática para aplicar a análise de agrupamentos e auxiliar a compreensão dos resultados de uma análise de sentimentos de abordagem em nível de aspectos.

Utilizando uma interface gráfica, o usuário poderá enviar um conjunto de dados resultante de uma análise de sentimentos sobre qualquer temática (desde que esteja estruturado tal como descrito na seção 4.2), definir os parâmetros para três algoritmo de *clustering* diferentes, aplicar a análise de agrupamentos e visualizar os resultados em gráficos.

A ferramenta foi projetada para poder executar várias análises simultaneamente, de forma assíncrona e de acordo com um processamento hierárquico de três níveis, ou seja, o módulo reúne os dados por estado, cidade e objeto, e aplica uma análise de agrupamentos particular para cada um dos subconjuntos.

Todos os resultados são salvos no servidor, onde o usuário poderá consultá-los sempre que precisar e sem ter que esperar por qualquer tipo de processamento – a não ser o carregamento do próprio arquivo de resultados.

Além disso, a definição do conceito da contribuição dos aspectos se mostrou uma interpretação de grande potencial no escopo da mineração de dados, pois permitiu inferir informações que não são muito explorados em problemas clássicos de *clustering* e podem prover uma série de outros estudos.

Em relação aos cenários de teste, um ponto importante em problemas de mineração de dados também foi um dos grandes desafios encontrados no desenvolvimento deste trabalho: a insuficiência de processamento.

Mesmo com especificações de *hardware* medianas para um computador pessoal (vide seção 4.7), houve muitos erros de estouro de memória e travamentos principalmente quando era utilizado todo o conjunto de dados. Além disso, o DBSCAN foi o algoritmo mais problemático, ele não conseguiu processar todo o conjunto de dados e forçou o autor a usar algumas soluções paliativas não muito interessantes para problemas de mineração de dados (seção 4.7.1).

Para trabalhos futuros, muitas opções de *upgrade* para o módulo são válidas. A principal delas é a implementação do componente de avaliação, que calcula o quão bom é um agrupamento utilizando não somente o coeficiente de silhueta, mas também outros métodos de avaliação. Disponibilizar mais parâmetros para os algoritmos na tela de configuração (seção 4.5.2) – como a medida de similaridade – também seria importante para identificar como os agrupamentos se comportariam.

Outra funcionalidade interessante seria um componente que permitisse comparar dois ou mais agrupamentos e a contribuição dos seus aspectos para melhorar cada vez mais as análises.

REFERÊNCIAS

AGGARWAL, C. C. **Data Mining: The Textbook**. 1. ed. New York, USA: Springer, 2015.

ARAÚJO, L. G. DE A. **Desenvolvimento do módulo de extração de regras de associação para o apriori-SentimentALL**, 2016.

BLEI, D. M. Introduction to Probabilistic Topic Modeling. **Communications of the ACM**, v. 55, p. 77–84, 2012.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent Dirichlet Allocation. **Journal of Machine Learning Research**, v. 3, n. 4–5, p. 993–1022, 2003.

BOOTSTRAP. **Bootstrap - The world's most popular mobile-first and responsive front-end framework.** Disponível em: <http://getbootstrap.com/>. Acesso em: 27 nov. 2016.

BRITO, P. F. DE et al. **SentimentALL**PalmasFábrica de Software: CEULP-ULBRA, , 2015.

CHRISTHIE, W. **SentimentALL: Ferramenta para Análise de Sentimentos em Português**, 2015.

DEVELOPERS, G. **Charts**. Disponível em: <https://developers.google.com/chart/>. Acesso em: 27 nov. 2016.

ESTER, M. et al. A Density-Based Clustering Methods. **Comprehensive Chemometrics**, v. 2, p. 635–654, 1996.

EVERITT, B. S. et al. **Cluster analysis**. [s.l: s.n.].

FALEIROS, T. DE P.; LOPES, ALNEU DE ANDRADE. Modelos probabilísticos de tópicos-desvendando o Latent Dirichlet Allocation. p. 2662–2673, 2016.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI magazine**, p. 37–54, 1996.

FELDMAN, R. Techniques and applications for sentiment analysis. **Communications of the ACM**, v. 56, n. 4, p. 82, 2013.

FLASK. **Flask (A Python Microframework)**. Disponível em: <http://flask.pocoo.org/>. Acesso em: 27 nov. 2016.

HAN, J.; KAMBER, M.; PEI, J. **Data mining: concepts and techniques**. 3. ed. Waltham: Elsevier, 2011.

HILBERT, M.; LÓPEZ, P. The world's technological capacity to store, communicate, and compute information. **Science**, v. 332, n. 6025, p. 60–5, 2011.

HU, F.; HAO, Q. **Intelligent Sensor Networks: The Integration of Sensor Networks, Signal Processing and Machine Learning**. [s.l: s.n.].

IBM. **What is Big Data?** Disponível em: <http://www.ibm.com/big-data/us/en/>. Acesso em: 20 maio. 2016.

Internet Usage & Social Media Statistics. Disponível em: <http://www.internetlivestats.com/>. Acesso em: 5 jun. 2016.

JAIN, A. K. Data clustering: 50 years beyond K-means. **Pattern Recognition Letters**, v. 31, n. 8, p. 651–666, 2010.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. **ACM Computing Surveys**, v. 31, n. 3, p. 264–323, 1999.

JIN, D.; LIN, S. **Advances in Computer Science, Intelligent System and Environment**. 3. ed. Newelska: Springer Science & Business Media, 2011. v. 3

KELAIAIA, A.; MEROUANI, H. F. Clustering with probabilistic topic models on arabic texts. **Studies in Computational Intelligence**, v. 488, n. 2, p. 65–74, 2013.

LINDEN, R. Técnicas de Agrupamento. **Revista de Sistemas de Informação da FSMA**, v. 4, p. 18–36, 2009.

LIU, B. Sentiment Analysis and Opinion Mining. **Synthesis Lectures on Human Language Technologies**, v. 5, n. 1, p. 1–167, 2012.

LU, Y.; MEI, Q.; ZHAI, C. Investigating task performance of probabilistic topic models: An empirical study of PLSA and LDA. **Information Retrieval**, v. 14, n. 2, p. 178–203, 2011.

LUGER, G. F. **Artificial Intelligence: Structures and Strategies for Complex Problem Solving**. 6. ed. [s.l: s.n.]. v. 5th

MITCHELL, T. M. **Machine Learning**. [s.l: s.n.].

NUNES, GIOPPO, F. Algoritmos de clustering para separação de culturas agrícolas e tipos de uso e cobertura da Terra utilizando dados de sensoriamento remoto. **Anais XVII Simpósio Brasileiro de Sensoriamento Remoto - SBSR**, n. 1, p. 6381–6388, 2015.

OLIVEIRA, S. R. M. **Clusterização ou Agrupamento de Dados**. Disponível em: <http://www.ime.unicamp.br/~wanderson/Aulas/Aula6/MT803-Aula06-Clusterizacao.pdf>. Acesso em: 5 jun. 2016.

PEDREGOSA, F. et al. **Scikit-learn: Machine Learning in Python**. Disponível em: <http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>. Acesso em: 12 abr. 2016.

PIECH, C.; NG, A. **K Means**. Disponível em: <http://stanford.edu/~cpiech/cs221/handouts/kmeans.html>. Acesso em: 5 jun. 2016.

PRASS, F. S. Estudo Comparativo Entre Algoritmos De Análise De Agrupamentos Em Data Mining. p. 70, 2004.

QIN, J.; NORTON, M. J. **Knowledge Discovery in Bibliographic Databases**. Champaign: University of Illinois, Graduate School of Library and Information Science, 1999.

ROESE, L. H. SentimentALL: módulo de visualização de informação. 2016.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 2. ed. Upper Saddle River, New Jersey: Pearson, 2003. v. 72

SAYAD, S. **An Introduction to Data Mining**. Disponível em: <http://www.saedsayad.com/data_mining.htm>. Acesso em: 26 maio. 2016.

SCHMITZ, F. E. B. Aplicação da técnica de text mining para comentários relacionados ao contexto do turismo. 2015.

SCIKIT-LEARN. **Clustering**. Disponível em: <http://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/modules/clustering.html>. Acesso em: 1 jul. 2016.

SCIKIT-LEARN 0.18.1 DOCUMENTATION. **sklearn.cluster.DBSCAN**. Disponível em: <http://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html>. Acesso em: 29 nov. 2016.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. Introduction to Data Mining. p. 736, 2005.

VENDRAMIN, L. **DBSCAN usando o Kd-Trees**. Disponível em: <http://wiki.icmc.usp.br/images/c/c7/DBSCAN_LucasVendramin.pdf>. Acesso em: 15 jun. 2016.

VERMA, J. P.; PATEL, B.; PATEL, A. Web Mining: Opinion and Feedback Analysis for Educational Institutions. v. 84, n. 6, p. 17–22, 2013.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining: Practical Machine Learning Tools and Techniques**. [s.l: s.n.]. v. 54

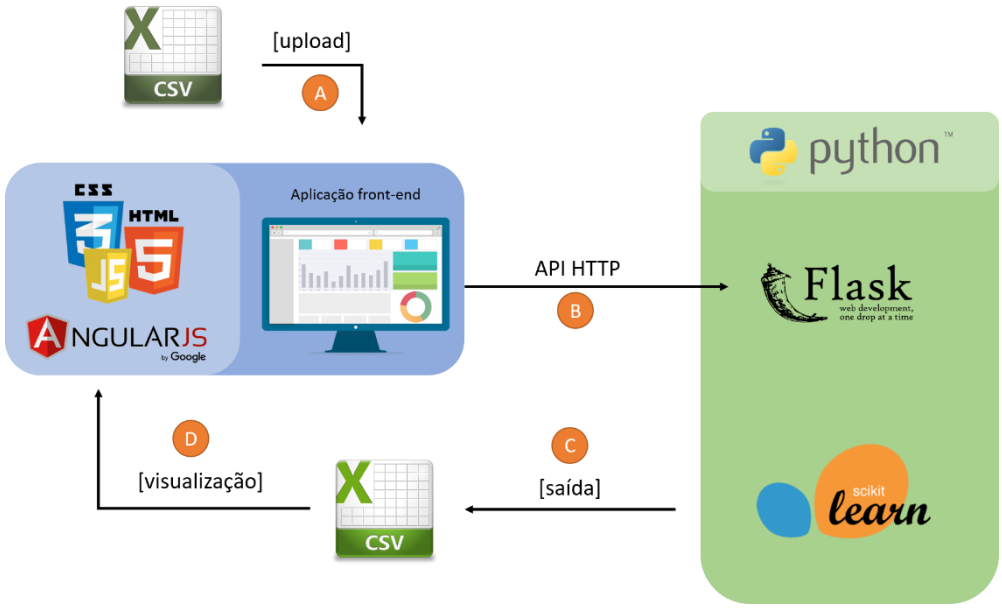
WU, J. **Advances in K-means Clustering: a data mining thinking**. 1. ed. New York, USA: Springer, 2012.

APÊNDICES

Apêndice A – Visão geral do módulo

A visão geral do módulo de análise de agrupamentos com ênfase nos materiais utilizados no processo está ilustrada na Figura 29.

Figura 29 – Visão geral do módulo de análise de agrupamentos.



A aplicação *front-end* irá executar os algoritmos nos dados de um arquivo CSV devidamente formatado e informado pelo usuário (Figura 29-A). Para iniciar o processamento, uma rota HTTP configurada com o *micro-framework* Flask (Figura 29-B) é acessada, fazendo o elo entre o código Python, Scikit-learn com a aplicação *front-end*. A análise irá gerar um outro arquivo CSV (Figura 29-C) que, por sua vez, também será carregado pela aplicação *front-end* para a apresentação dos dados utilizando a visualização *TreeMap*.

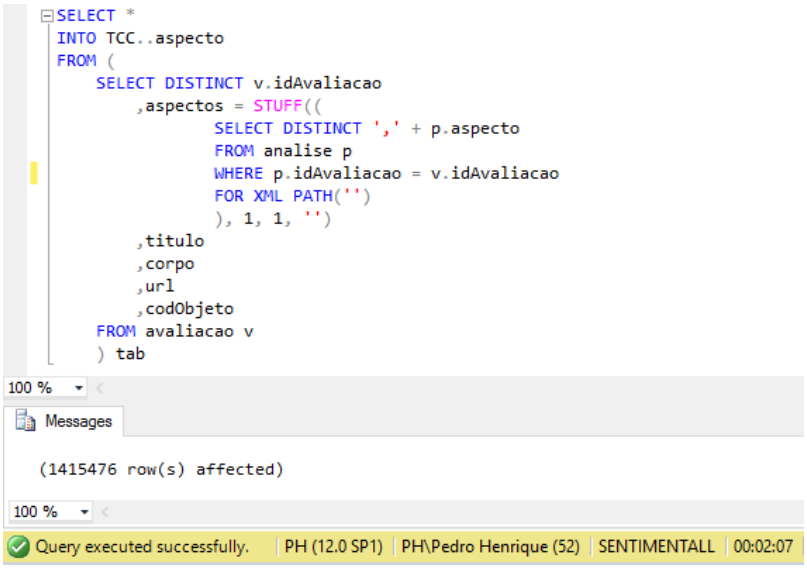
Arquivos CSV são muito comuns para armazenar dados que são utilizados em tarefas de mineração de dados. Eles são simples e leves, mas uma estrutura bem definida é fundamental para a eficiência dos algoritmos. Pensando nisso, a seção a seguir descreve a estrutura que o arquivo CSV deve ter para que o processamento do módulo ocorra normalmente.

Apêndice B – Tratamento de dados da entrada

A etapa de pré-processamento teve como objetivo acessar os dados extraídos por Christhie (2015), selecionar as variáveis de interesse, eliminar dados inconsistentes e formatar os outros dados para o prosseguimento da pesquisa. Deste modo, consultas SQL foram criadas para realizar a seleção no banco de dados.

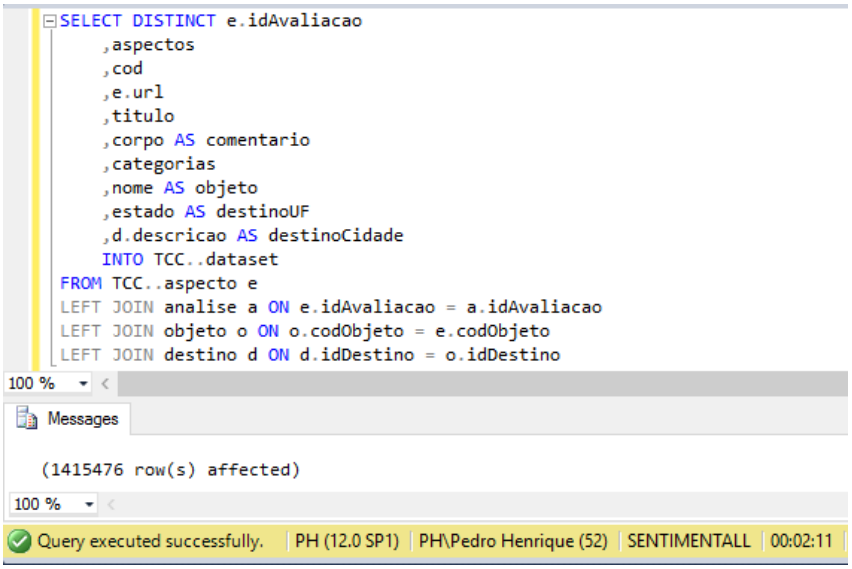
A primeira delas retorna a lista as informações relacionadas a avaliação, com destaque para o campo *aspectos*, que contém a lista de aspectos presentes na avaliação separados, separados por vírgula. A consulta em questão está apresentada na Figura 30 e foi fortemente baseada na proposta de Schmitz (2015).

Figura 30 – Captura de tela da primeira consulta SQL na etapa de seleção.



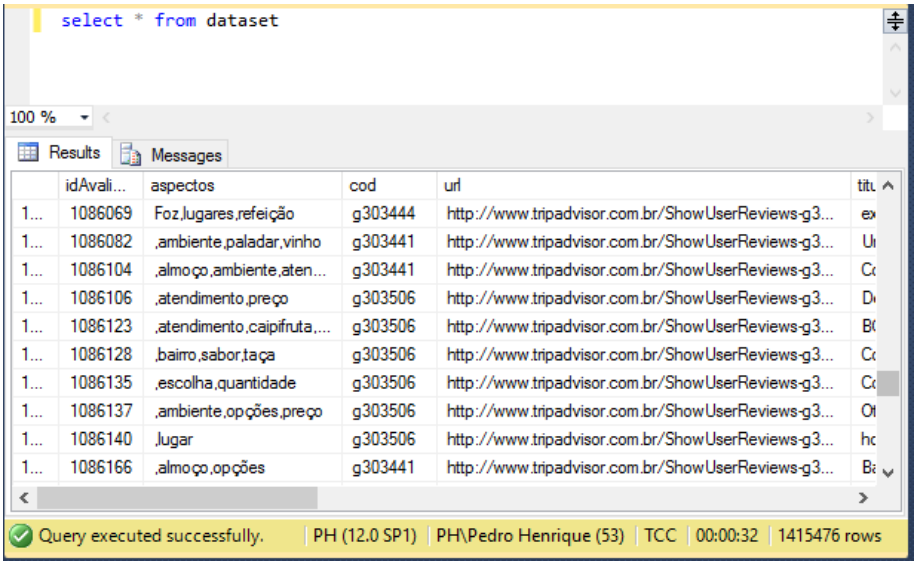
Para facilitar, a resultado da consulta é inserido uma nova tabela chamada *aspecto* que, por sua vez, é também usada na segunda consulta SQL da etapa de seleção. Nela, é acrescentado, basicamente, as informações do destino relacionado a avaliação. A segunda consulta está ilustrada na Figura 31.

Figura 31 – Captura de tela da segunda consulta SQL na etapa de seleção.



Ao final, a seleção (e conjunto) de dados que estará hábil as especificações da solução que foi desenvolvida é apresentada na Figura 32.

Figura 32 – Captura de tela da tabela final.



Vale ressaltar que a etapa de seleção será orientada pelo especialista de domínio.

A formatação e armazenar adequadamente os dados para que os algoritmos possam ser aplicados, se tornou bastante simples por conta do recurso de exportação do SQL Server. Após realizar uma consulta, a própria IDE permite e exportação dos resultados em formato CSV, que será formato utilizado como entrada para os algoritmos.

Apêndice C – Resumo dos Módulos do SentimentALL

Diferentes autores foram responsáveis pelo desenvolvimento dos módulos onde, basicamente, foi utilizado uma mesma base de dados. Tendo em base a Figura 13, uma breve descrição de cada um dos seus módulos do SentimentALL está apresentada a seguir:

- **Módulo de extração de dados**:
  - *Descrição*: crawler[7] de navegação e extração sistemática de avaliações sobre destinos turísticos do TripAdvisor. Desenvolvido utilizando técnicas de *web crawling* e *web scraping* fornecidas pela aplicação Import.io[8] (CHRISTHIE, 2015).
  - *Principais tarefas realizadas*:
    - Desenvolvimento do *crawler* de extração; e
    - Limpeza, formatação e carga dos dados obtidos.

- **Módulo de mineração de dados por análise de sentimentos**:
  - *Descrição*: protótipo de ferramenta de análise de sentimentos em textos escritos em Português, no nível de aspectos e abordagem léxica (CHRISTHIE, 2015).
  - *Principais tarefas realizadas*:
    - Identificação dos aspectos presentes nas avaliações e a definição da sua polaridade; e
    - Persistência dos resultados na base de dados unificada.

- **Módulo de mineração de dados por regras de associação**:
  - *Descrição*: aplicação da técnica supervisionada de regras de associação utilizando o Weka[9] para extrair relações entre os aspectos positivos e negativos mais frequentes (ARAÚJO, 2016; SCHMITZ, 2015). Para a análise, foram utilizados somente avaliações que continham pelo menos um dos vinte aspectos positivos mais comuns no banco de dados.
  - *Principais tarefas realizadas:*

[7] Programa autônomo que extrai dados sistematicamente da web.
[8] O Import.io é uma plataforma web para extrair dados de websites que dispensa qualquer tipo de escrita de código. Mais informações em: www.import.io.
[9] O Weka é uma coleção de algoritmos de aprendizagem de máquina para tarefas de mineração de dados. Mais informações em: www.cs.waikato.ac.nz/ml/weka.

        - Criação de tabelas auxiliares para facilitar a transformação dos dados para o formato do Weka (ARFF);
        - Aplicação dos dados no algoritmo *A Priori*.

- **Módulo de visualização de informação**:
    - *Descrição*: ferramenta de visualização de informação para facilitar a verificação de padrões e tendências pelas partes interessadas (ROESE, 2016).
    - *Principais tarefas realizadas***:**
        - Criação de tabelas auxiliares na base de dados unificada para facilitar as consultas;
        - Desenvolvimento de componente para geração de dicionário de termos e taxonomia;
        - Desenvolvimento de componente para apresentações em gráficos variados.

Apêndice D – Diagrama de sequência do módulo

Figura 33 – Diagrama de sequência do módulo de análise de agrupamentos.