

Paulo Henrique Fernandes Leite

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

São Paulo – Brasil

2019

Paulo Henrique Fernandes Leite

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

Monografia apresentada na disciplina Trabalho de Conclusão de Curso, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação.

Centro Universitário Senac - Santo Amaro
Bacharelado em Ciência da Computação

Orientador: Alexandre dos Santos Mignon

São Paulo – Brasil
2019

Agradecimentos

Agradeço ao relógio por nos ter concedido tempo o suficiente para realizar este projeto. À minha mãe, Maria Lindomar, um especial obrigado por tudo. Ao meu pai, Denilson, agradeço pelos ensinamentos para toda a vida e em especial, por sempre ter incentivado meus estudos. Ao meu irmão Denilson Junior, um obrigado por me ajudar quando eu realmente precisei. Para meus sobrinhos e sobrinhas, espero que tal projeto de alguma forma possa servir de inspiração para que vocês alcancem os seus objetivos. Para meu orientador, Alexandre dos Santos Mignon, agradeço muito pela paciência, orientações e conselhos. Para a professora Cristiane Yayoko Ikenaga, agradeço pela ajuda e dicas de como seguir com o projeto. Para minha amada noiva, Laurana Franco, obrigado pela paciência. Graças ao seu apoio, meu sentimento em relação a este trabalho é de dever cumprido.

Resumo

A proposta deste trabalho é aplicar os princípios e técnicas existente na disciplina de Ciência de Dados para desenvolver um modelo preditivo de sistema de recomendação a partir do histórico de uso dos clientes e seus restaurantes favoritos. Usando o conjunto de dados fornecido pela *startup* brasileira *ChefsClub*, foram extraídos recursos baseados em conteúdo e colaboração para identificar perfis de clientes e restaurantes. A partir disto, foi implementado o agrupamento dos dados por meio do método *K-means* pelo qual utiliza o conceito de *clusters* para expor padrões baseados na proximidade das informações. Os dados *clusterizados* permitem a obtenção dos restaurantes mais condizentes com o perfil do cliente. Com o objetivo de melhorar o desempenho do algoritmo utilizado e aumentar a precisão, as recomendações são realizadas por rodadas, onde em cada uma delas é passada um base de restaurantes e outra de preferências do usuário. As bases passam pelo processo de *clustrização* do *K-means* que por sua vez retorna os *clusters* mais semelhantes ao perfil do usuário. O ciclo termina quando o total de recomendações forem no máximo 50 restaurantes ou quando o método *elbow*, responsável por analisar a variação dos dados, identificar que os dados não podem ser mais divididos, isto é, o número de *cluster* é 1 e os dados possuem baixa variação. As informações analisadas neste trabalho consistem inicialmente no aprendizado de dados não supervisionados e posteriormente no aprendizado por reforço, ou seja, semi supervisionados, por meio da ação externa o usuário. Para avaliar o desempenho do método escolhido, foram realizados testes qualitativos, seguindo o princípio de Pareto. As recomendações são avaliadas pelos clientes utilizando 2 níveis de satisfação (gostei e remover) e o aprendizado do algoritmo é obtido a partir da reação do cliente em relação à recomendação. Sempre que uma recomendação é avaliada de forma positiva, o sistema refaz a recomendação utilizando a nova informação como base. O mesmo vale para quando uma recomendação é removida, isto é, o usuário não aprovou a sugestão, neste caso, o dado e seus similares passam a ser ignorados e deixam de ser exibidos.

Palavras-chaves: Recomendação de restaurantes, Análise de Dados, Ciência de Dados, K-means, Aprendizado de Máquina, Não Supervisionado, semi supervisionado.

Abstract

The purpose of this thesis is to apply the principles and techniques existing in the Data Science discipline to develop a predictive model of recommendation system based on the usage history of customers and their favourite restaurants. Using the dataset provided by Brazilian startup ChefsClub, content-based and collaborative capabilities to identify customer and restaurant profiles were extracted. From this, data grouping was implemented through the K-means method whereby it uses the concept of clusters to expose patterns based on the proximity of the information. The Clustered data make possible to know which restaurants best match the customer profile. In order to improve the performance of the algorithm and to increase its accuracy, the recommendations are made by rounds; each round uses a restaurants database the user preferences. The cycle ends when the total number of recommendations is no more than 50 or when the elbow method, which is responsible for analyzing data variation, identifies that the data cannot be further divided meaning that the number has a low variation. The information analyzed in this thesis consists initially in learning unsupervised data and later learning by reinforcement, ie semi-supervised, through external action the user. To evaluate the performance of the chosen method, qualitative tests were performed, following the Pareto principle. Recommendations are evaluated by customers using 2 levels of satisfaction (like and remove) and the learning of the algorithm is derived from the customer's reaction to the recommendation. Whenever a recommendation is evaluated positively, the system redo the recommendation using the new information as a basis. The same goes for when a recommendation is removed, meaning that the user did not approve the suggestion, resulting in similar data being ignored or no longer displayed.

Keywords: restaurants Recomendation, Data Analysis, Data Science, K-means, Machine Learning, Not-Supervised, Semi-Supervised.

Listas de ilustrações

Figura 1 – <i>Data Never Sleeps - How much data is generated every minute?</i>	15
Figura 2 – Diagrama de Venn da ciência de dados	17
Figura 3 – Visão panorâmica da ciência de dados	18
Figura 4 – Variáveis dependentes e independentes	21
Figura 5 – Exemplo de dados não supervisionados	22
Figura 6 – Experimento de Pavlov	24
Figura 7 – <i>Do you trust online customer reviews as much as personal recommendations?</i>	26
Figura 8 – Recomendação de produto baseado em similaridade	27
Figura 9 – Amostra de dados dispostas nos eixo X e Y	29
Figura 10 – Amostra de dados dispostas nos eixo X e Y com aplicação do K-means	30
Figura 11 – Exemplo de visualização em estrutura de árvore	30
Figura 12 – Método Elbow utilizando K-means	32
Figura 13 – Representação cíclica do fluxo de recomendação	34
Figura 14 – Diagrama do banco de dados desenvolvido para o trabalho	35
Figura 15 – Distribuição de dados da tabela cuisine	39
Figura 16 – Tabela de normalização por transformação linear	40
Figura 17 – Abstração do funcionamento do algoritmo desenvolvido	41
Figura 18 – <i>Clusters separador por price e discount</i>	42
Figura 19 – Fluxograma do algoritmo de recomendação	44
Figura 20 – Exemplo de pedido de avaliação de recomendação	48
Figura 21 – Exemplo de recomendação avaliada positivamente	49

Listas de tabelas

Tabela 1 – Distribuição de dados da tabela <i>kind</i>	36
Tabela 2 – Distribuição de dados da tabela <i>moments</i>	36
Tabela 3 – Distribuição de dados da tabela <i>categories</i>	36
Tabela 4 – Distribuição de dados da tabela <i>restaurant</i> , parte 1	37
Tabela 5 – Distribuição de dados da tabela <i>restaurant</i> , parte 2	37
Tabela 6 – Distribuição de dados da tabela <i>restaurant</i> , parte 3	37
Tabela 7 – Distribuição de cidades e total de restaurantes	38
Tabela 8 – Distribuição de categorias normalizadas	40
Tabela 9 – Distribuição de recomendações por usuários	54
Tabela 10 – Distribuição de avaliações positivas (<i>like</i>) por usuário	55
Tabela 11 – Distribuição de avaliações negativas (<i>unlike</i>) por usuário	55

Lista de abreviaturas e siglas

2D: *Two-Dimensional*

API: *Application Program Interface*

BI: *Business Inteligence*

CD: Ciência de Dados

GPS: *Global Positioning System*

IA: Inteligência Artificial

IBPAD: Instituto Brasileiro de Pesquisa e Análise de Dados

IDE: *Integrated development environment*

ISO: *International Organization for Standardization*

MVVM: *Model-View-ViewModel*

RESTful: *Model Representational state transfer*

SQL: *Structured Query Language*

UI: *User Interface*

Sumário

1	INTRODUÇÃO	10
1.1	Objetivo	12
1.2	Justificativa	13
2	REVISÃO BIBLIOGRÁFICA	14
2.1	Ciência de Dados	14
2.2	Aprendizado de máquina	20
2.3	Modelos de aprendizado de máquina	20
2.3.1	Aprendizado supervisionado	21
2.3.2	Aprendizado não supervisionado	22
2.3.3	Aprendizado semi supervisionada	23
2.3.4	Síntese: qual é o melhor modelo?	25
2.4	Sistemas de recomendação	25
2.4.1	Sistemas de recomendação em grupo	27
2.5	K-means Clustering	28
2.6	O método Elbow	32
3	PROJETO	33
3.1	Visão geral	33
3.2	Base de dados	35
3.3	Normalização	39
3.4	Analizador	41
3.5	Resultados	46
3.6	Aplicativo	46
3.6.1	Onboarding	47
3.6.2	Primeira recomendação não supervisionado	48
3.6.3	Primeira recomendação semi supervisionado	48
3.6.4	Recomendações baseadas em reservas e favoritos	49
3.7	Tecnologias utilizadas	49
4	RESULTADOS E DISCUSSÕES	51
4.1	Método de avaliação	52
4.2	Teste com usuários	52
5	CONCLUSÃO	56
5.1	Sugestões para trabalhos futuros	57

REFERÊNCIAS	59
A APÊNDICE: CARTA DE SOLICITAÇÃO DE AUTORIZAÇÃO PARA USO DE DADOS	62
B APÊNDICE: CARTA DE SOLICITAÇÃO DE AUTORIZAÇÃO PARA TESTE COM USUÁRIO	63

1 Introdução

A Ciência de Dados, ou *Data Science*, começou a ser conceituada a partir de meados dos anos 1960. Em 1962, o artigo *The Future of Data Analysis*, publicado por Tukey (1962), apresenta seu interesse na análise de dados e no estudo das características científicas a partir dela e não somente da análise do ponto de vista matemático e estatístico, comum até então.

O estudo se popularizou para o restante do mundo uma vez que Naur (1974) publicou seu material sobre *Concise Survey of Computer Methods* na Suécia e Estados Unidos, em 1974, reforçando a visão de Tukey (1962) abordada anteriormente. No ano de 1977, Tukey (1977) divulgou um novo material pelo qual trouxe ênfase no estudo de *Exploratory Data Analysis* lado ao estudo sobre *Confirmatory Data*.

Nos dias atuais, a ciência de dados volta a ser alvo de pesquisas no ramo acadêmico. Por conseguir obter informações relevantes por análises de diferentes casos, não é difícil vê-las em publicações nas mais diferentes áreas do conhecimento, como é visto por Faghmous; Kumar (2014), publicando estudos de mudanças climáticas, aplicações no setor da saúde, proposto por Krumholz (2014) e na área da educação por Finzer (2013).

Além disso, a ciência de dados também está sendo enfatizada no ramo corporativo por conseguir converter dados gerados por diferentes meios, em conhecimento relevante para o mercado (AMARAL, 2016). Uma das utilizações está no sistema de recomendação, que visa indicar as melhores opções para o usuário, mesmo que o mesmo não tenha conhecimento da indicação. Hoje, esses sistemas podem ser encontrados em empresas como *Ebay*, *Amazon*, *Netflix* e *Last.fm* (TAKAHASHI, 2015).

O presente trabalho tem por objetivo aplicar os princípios e técnicas da ciência de dados para desenvolver um modelo preditivo de sistema de recomendação a partir do histórico de reserva dos clientes e seus restaurantes favoritos.

Os sistemas de recomendação de filmes, músicas e restaurantes, que é o foco deste trabalho, são exemplos de aprendizagem não supervisionada, isto é, não existe uma saída esperada como geralmente visto em modelos supervisionada (HONDA; FACURE; YAOHAO, 2017). Por conta do aprendizado ser não supervisionada, utiliza-se o método *K-means*, que é o indicado para este tipo de modelo.

Apesar da aprendizagem ser não supervisionada, isto é, o grau de confiabilidade das informações sugeridas aos usuários acabam sendo comprometidas, pois não existe uma rotulação dos dados de saída, o presente trabalho mitiga o problema por meio do aprendizado por reforço, apresentado inicialmente por Pavlov (2013) em seu experimento

conhecido por *Pavlov's Dogs*.

O primeiro modelo se aplica quando ainda não existem validações do usuário perante as sugestões passadas a ele. Neste momento a recomendação é obtida pelo histórico de reservas dos restaurantes e lista de favoritos.

Para mitigar problemas com recomendação mal classificada, isto é, quando é recomendado um dado com baixa similaridade em relação as preferências do usuário, o sistema faz uma análise cíclica utilizando como entrada do método de avaliação a base de restaurantes e as preferências de usuário, sendo elas reservas e favoritos. Os atributos utilizados para ambas as bases são: cidade, bairro, latitude, longitude, preço, avaliação, categoria, cozinha, tipo, momento, desconto, benefício, restrição, quantidades de pessoas na mesa, se é válido para feriados e se possui *wifi* gratuito.

Para escolher os atributos a serem analisados neste trabalho, o autor partiu da linha de pensamento dos artigos publicados por Sawant (2013), que propõe uma recomendação de restaurantes utilizando a base de dados da *Yelp*, Jayasimhan; Rai; Parekh; Patwardhan (2017), que por meio dos dados obtidos da plataforma *Zomato* e novamente da *Yelp*, aplica técnicas e algoritmos de aprendizagem de máquina para recomendar restaurantes, e por fim, Manoli (2015), que utilizam dados de localização do restaurante, cozinha, ambiente e avaliações como quesitos para preparar a recomendação.

Em cada ciclo, o *K-means* é chamado e junto a ele, o método *Elbow*, responsável por avaliar a variação entre os dados e informar a melhor distribuição dos dados. Durante o ciclo, é feito a separação da base de restaurante por semelhantes e em seguida, é utilizado a base de preferências do usuário para entender a quais grupos ele mais se assemelha. Os grupos que não tiverem semelhança com o usuário ou possuírem um número a baixo da média, são descartados e é dado início a um novo ciclo com os dados que sobraram. O processo se repete até que seja recomendado no máximo 50 opções de restaurantes, garantindo assim que não seja recomendado milhares de opções, perdendo o sentido da recomendação, ou quando o *Elbow* informar que os dados não possuem variações entre eles, isto é, não fazem mais sentido serem divididos.

O aprendizado por reforço se dá pela ação do usuário. Ele tem a possibilidade de validar uma recomendação, informando ao sistema que gostou de algo recomendado ou que não gostaria de receber a sugestão passada. Uma vez avaliado a recomendação, o sistema passa a trabalhar dentro do modelo semi supervisionado (HONDA; FACURE; YAOHAO, 2017). Sempre que uma recomendação é avaliada de forma positiva, o sistema refaz a recomendação utilizando a nova informação como base. O mesmo vale para quando uma recomendação é removida, isto é, o usuário não aprovou a sugestão, neste caso, o dado e seus similares passam a ser ignorados e deixam de ser exibidos.

As recomendações oferecidas pelo algoritmo, são realizadas para usuários que

utilizam soluções *mobile* para terem experiências gastronômicas pela cidade de São Paulo, onde está situado o maior número de restaurantes existentes na base de dados utilizado neste trabalho, entretanto, é possível escolher e visualizar as recomendações de até 22 regiões diferentes, isto por que a base de dados conta com mais opções e não somente São Paulo. A concepção por trás do algoritmo decorreu do paradoxo da escolha, apresentado por Schwartz (2007), onde ele aborda a dificuldade do indivíduo em escolher um conteúdo perante a existente de muitos outros, isto é, são diversas opções que ao final, muito provavelmente a escolha ficará dentro da zona de conforto. Tendo o foco no auxílio da escolha de restaurantes, para melhorar a experiência de uso do mesmo, com recomendações que tenham maior proximidade com o seu perfil, diminuindo a variedade de opções e trazendo escolhas mais assertivas.

A visualização dos dados é feita por meio da aplicação *mobile* desenvolvida em *Flutter*, que possibilita navegar tanto em ambiente *Android* quanto em *iOS*. A recomendação é exibida em forma de listas no aplicativo, similar ao que já é visto em serviços de *streaming* de música e filmes.

1.1 Objetivo

O objetivo deste trabalho é desenvolver um sistema de recomendação de restaurantes utilizando técnicas de aprendizagem de máquina para modelos não supervisionados. Além disso, também tem o objetivo de implementar uma plataforma *mobile*, onde será exibido às recomendações, além de ser responsável por simular reservas de mesa e possibilitar que o usuário possa favoritar restaurantes. É por meio do histórico de reservas e lista de restaurantes favoritos que o sistema a ser desenvolvido fará a análise, resultando em grupos de restaurantes recomendados. Para validar o modelo, será utilizado ação externa do usuário para informar se gostou da recomendação. Essa ação permite validar e tornar o modelo que inicialmente é não supervisionados para um modelo semi supervisionado.

Os objetivos específicos deste trabalho são:

- Identificar vetores dos restaurantes para criação do modelo;
- Normalizar os dados de acordo com o vetor de atributos escolhido;
- Utilizar o algoritmo *K-means* para agrupar os restaurantes por similaridade;
- Desenvolver aplicativo em *Flutter* para exibir os restaurantes recomendados;
- Utilizar o conceito de modelo semi supervisionado por meio de ação externa do usuário.

1.2 Justificativa

Como toda ciência em crescimento, na recomendação surgem cada vez mais resultados de pesquisa e aplicações que são amplamente utilizados por grandes prestadores como *Netflix*, *Amazon*, *Spotify*, entre outras¹. No caso da *Amazon*, o resultado da empresa foi o aumento do faturamento em 29% após 3 meses de implementação do sistema de recomendação.

O presente trabalho se justifica não somente pela ciência em crescimento, mas também pelo crescimento do mercado de aplicações móveis desenvolvidas para comer fora de casa². A produção deste tipo de solução chamou a atenção não apenas de consumidores, mas também de investidores neste período.

Essas aplicações oferecem incentivos financeiros para os consumidores comerem mais fora de suas casas. Desta forma, o mercado de comer fora de casa ganhou grande importância para o país, tornando-se importante analisar novos meios de melhorar e aumentar a produtividade do setor. Além disso, as soluções para este mercado estão movimentando muito dinheiro, como é o caso do *iFood* em 2018, no qual recebeu um aporte de US\$ 500 milhões³.

O presente trabalho tem o objetivo de desenvolver um sistema de recomendação de restaurantes utilizando técnicas de aprendizagem de máquina para modelos não supervisionados. A pesquisa irá procurar por possíveis correlações entre histórico de reservas, lista de restaurantes favoritos, ação externa do usuário e base geral de restaurantes. A base de dados utilizada no trabalho foi extraída do site <https://www.chefsclub.com.br/restaurantes> em outubro de 2019 e contém dados no período de 11 de julho de 2012 até o presente momento deste trabalho. Os dados foram extraídos mediante a autorização da empresa.

¹ <https://www.prophetsharing.co/streams/the-real-secret-to-amazon-netflix-and-spotifys-success>

² <https://forbes.uol.com.br/negocios/2015/06/mesmo-com-crise-setor-de-alimentacao-continua-a-crescer-no-brasil/>

³ <https://forbes.uol.com.br/last/2018/11/ifood-captta-us-500-milhoes-no-maior-aporte-da-america-latina/>

2 Revisão Bibliográfica

Neste capítulo é apresentado a fundamentação teórica essencial necessária para o tema abordado neste trabalho. Primeiramente são contextualizados os conceitos existentes na ciência de dados, suas possíveis aplicações e áreas de atuação¹. Em seguida são explorados os estudos de aprendizado computacional e a mineração de dados, frequentemente presentes em análise de dados. Também são apresentados alguns métodos matemáticos e algoritmos computacionais relevantes para a execução de análises. Por fim, é inserido o contexto de restaurantes e a aplicação de toda a fundamentação teórica para o desenvolvimento de um sistema de recomendação baseado em uma plataforma de reserva *online* de restaurantes.

O principal escopo deste trabalho é estudar as teorias e aplicações da ciência da computação para análise de grande volume de dados com o objetivo de criar um sistema de recomendação. Existem diversos recursos que podem ser utilizados no âmbito científico tecnológico para análise de grande volume de dados, como por exemplo, mineração de dados, normalização, análise estatística por meio da programação em R e a ciência de dados. Um ponto chave para a execução do trabalho é o entendimento dos modelos de mineração de dados e qual algoritmo usar para cada caso.

Os algoritmos que permitem e facilitam a análise dessas informações podem ser classificados por: Algoritmos de classificação, que preveem uma ou mais variáveis discretas, com base nos outros atributos do conjunto de dados; Algoritmos de regressão, que preveem uma ou mais variáveis numéricas contínuas, como lucro ou perda, com base nos outros atributos do conjunto de dados; Algoritmos de segmentação, que dividem dados em grupos ou *clusters* de itens que têm propriedades semelhantes; Algoritmos de associação, que encontram correlações entre atributos diferentes em um conjunto de dados; Algoritmos de análise de sequência, que resumem sequências ou episódios frequentes em dados, como uma série de cliques em um *site* da *web* ou uma série de eventos de *log* de manutenção de computador (CORTES, 2002).

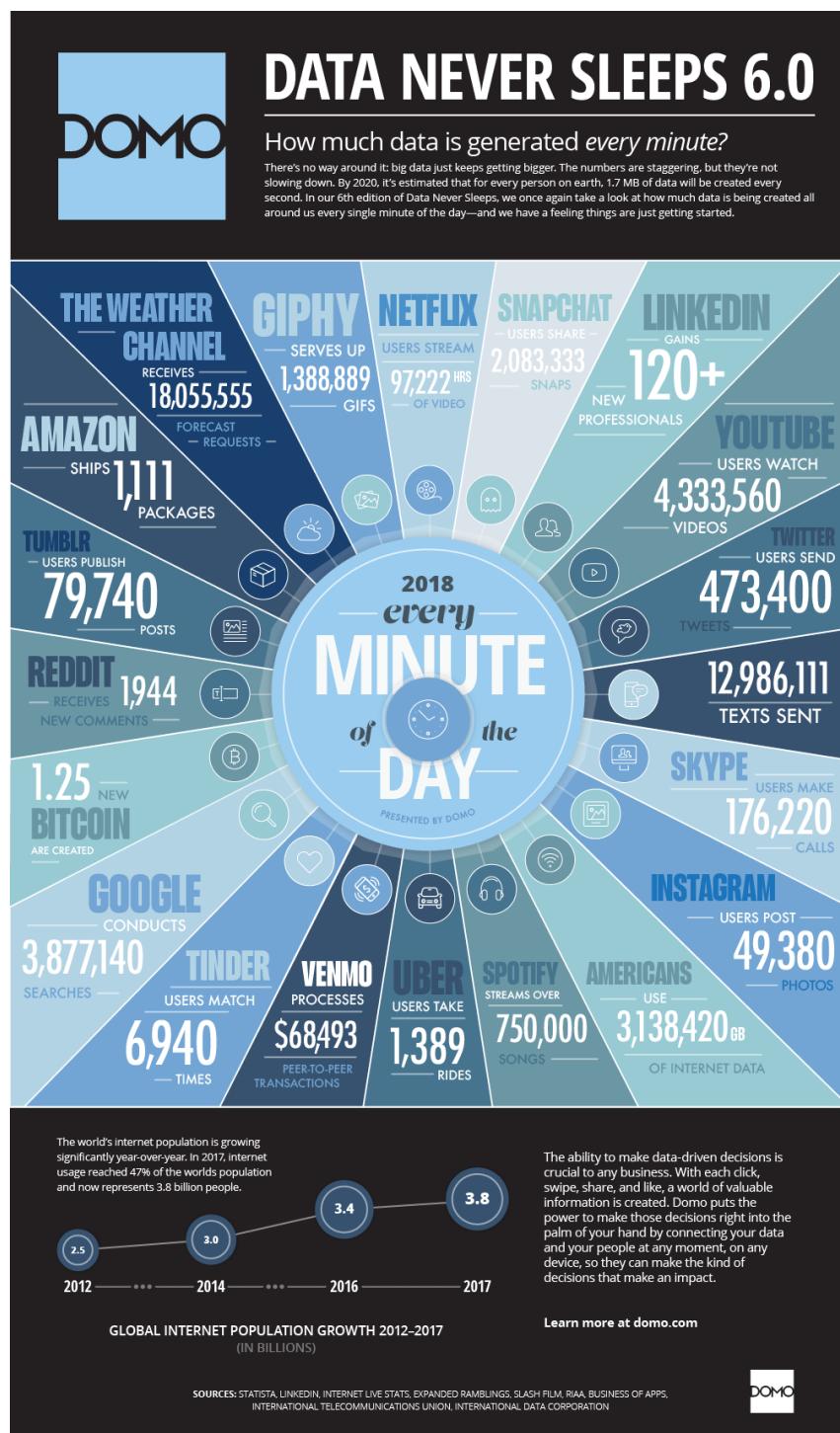
2.1 Ciência de Dados

Entende-se por Ciência de Dados (CD) os processos criados em prol de tecnologias, modelos que estudam os dados desde a sua produção até o seu descarte. Pode ser definido ainda como técnicas baseadas em Inteligência Artificial e Estatística, tendo por finalidade a descoberta de novos conhecimentos nas grandes máximas (AMARAL, 2016).

¹ <https://coggle.it/diagram/WypjBWlbpEHkyhcg/t/data-science-for-managers/c0083ef2d003b849535618df0b84e6dad73e301910c45e7eab5b72ff12f016e0>

Divulgado por *Domo*² no ano de 2017, os dados sobre *Data Never Sleeps*, mostrando a média de produção de dados feitos por minuto. Mostra ainda o seu alinhamento com a inovação. A criação desse escopo mostra ainda dados de vários setores voltados para a tecnologia, educação, serviços gastronômicos, eventos de culinária, serviços financeiros e outras áreas diversas.

Figura 1 – *Data Never Sleeps - How much data is generated every minute?*



Fonte: *DOMO* (2017)

² <https://www.domo.com/learn>

O Instituto Brasileiro de Pesquisa e Análise de Dados (IBPAD)³ orienta que a (CD) é um tipo de atividade interdisciplinar que tem por finalidade conciliar duas áreas específicas: a Ciência da Computação e a Estatística. Essa ciência pode ser aplicada ainda em muitas outras áreas de conhecimento, inclusive para análise de dados.

A Ciência de Dados também pode ser aplicada na área de *Business Intelligence*, e tem importante papel para o desenvolvimento de tecnologia voltada para suportar grande quantidade de dados, com a finalidade de contribuir com a criação de novas estratégias de negócios (TURBAN et al., 2009).

O objetivo da área de BI é o de interpretar de forma fácil um grande número de dados específicos, de maneira a buscar por novas oportunidades por meio de informação, promovendo assim uma certa vantagem competitiva e o aumento da estabilidade dos negócios (ECKERSON, 2013). Esse processo acontece por meio da busca de informações com o objetivo de fazer uma avaliação do ambiente empresarial no qual o negócio está inserido, de maneira a completar informações sobre o negócio por meio de relatórios, pesquisas financeiras e de marketing, além de outros recursos.

As fontes primárias de informação são produzidas a partir de dados básicos das empresas. Já as fontes secundárias são captadas por meio daquilo que os consumidores identificam como necessidades, levando em conta suas decisões acerca de produtos e serviços, as condições submetidas, aspectos econômicos apresentados e o nível de tecnologia utilizado pelas empresas (MILLER, 1865). De posse de todos esses dados, os sistemas de (BI) criam metas específicas, levando-se em conta os objetivos financeiros idealizados pelas empresas. Com isso, decisões inteligentes passam a ser percorridas, direcionando o foco para os resultados previamente idealizados.

Desde meados dos anos 1960, a ciência de dados tem sido abordada como a ciência dos processos de dados (TUKEY, 1962), e sob o termo datalogia de Naur (1974). Em seu trabalho, *The Future of Data Analysis* publicado em 1962, Tukey (1962) aborda seu interesse na análise de dados e no estudo das características científicas a partir dos dados e não somente a análise de dados do ponto de vista matemático e estatístico como era comum na época. Ele acreditava na abordagem do estudo dos dados como meio de aplicação para outras áreas do conhecimento. Naur (1974) deixou sua contribuição para a disciplina de ciência de dados com a publicação de seu material em 1974 sobre *Concise Survey of Computer Methods*.

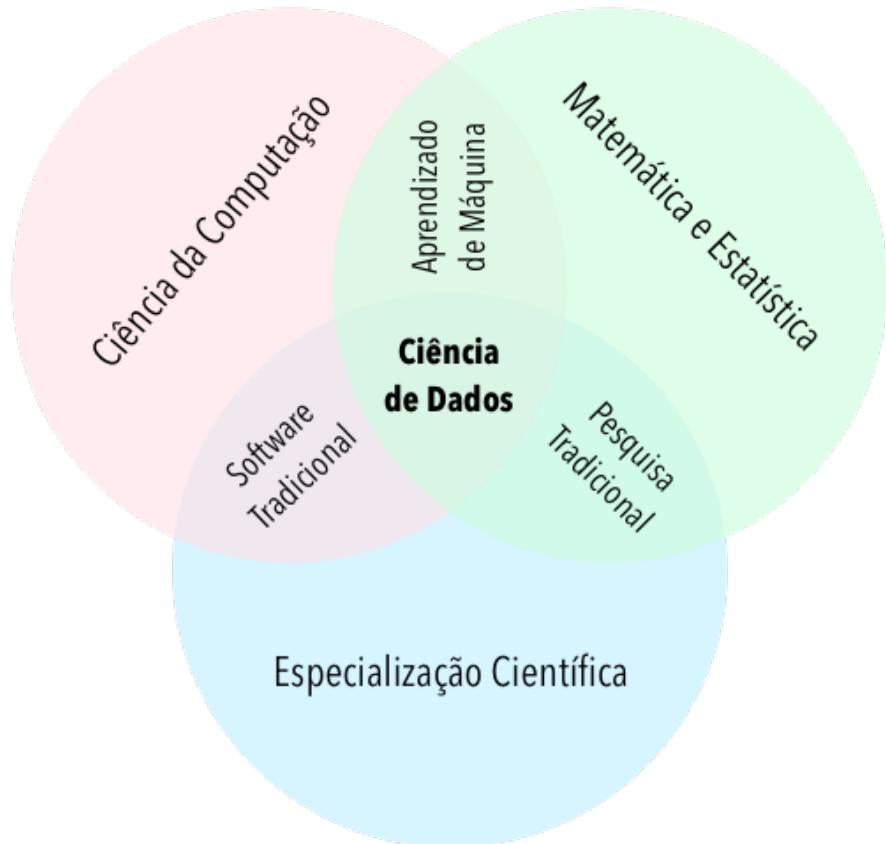
Apesar de ter surgido nos anos 1960s, ela é vista como uma ciência nova (AMARAL, 2016). Atualmente, a ciência de dados está sendo enfatizada no ramo corporativo por conseguir converter dados gerados por diferentes meios em informações relevantes para o mercado (AMARAL, 2016).

³ <https://www.ibpad.com.br/o-que-fazemos/publicacoes/historico-das-apis-no-monitoramento-e-pesquisa-em-midias-sociais>

Nos dias atuais, a ciência de dados volta a ser alvo de pesquisas no ramo acadêmico. Por conseguir obter informações relevantes por análises de diferentes casos, não é difícil vê-las em publicações nas mais diferentes áreas do conhecimento, como é visto por Faghmous; Kumar (2014), publicando estudos de mudanças climáticas, aplicações no setor da saúde, proposto por Krumholz (2014) e na área da educação por Finzer (2013).

A ciência de dados é fundamentada em recursos e teorias originadas das ciências básicas, promovendo uma nova área altamente multidisciplinar (ZIVIANI, 2015). Ela está profundamente ligada às áreas de Ciência da Computação, Matemática e Estatística, Especialização Científica e Design Gráfico (BAUDISCH, 2016).

Figura 2 – Diagrama de Venn da ciência de dados



Fonte: Baudisch (2016)

É possível discriminar e representar cada subárea da seguinte maneira:

- Ciência da Computação: trata-se da acumulação, aquisição e o processamento de dados por meio de sistemas de informação;
- Matemática e Estatística: envolve o uso de técnicas de filtragem, especialização, classificação e agrupamento de dados;
- Especialização Científica: parte de métodos científicos e questionamentos baseados na presença de padrões entre os dados;

- Design Gráfico: envolve formas de visualização e refinamento das informações;
- Aprendizado de Máquina: trata-se do estudo científico dos algoritmos e modelos estatísticas que sistemas de computador usam para realizar uma tarefa específica sem usar instruções explícitas, confiando em padrões;
- *Software* Tradicional: conjunto de instruções, data ou programas usados para operar computadores e executar tarefas específicas;
- Pesquisa Tradicional: similar ao método científico, trata-se do estudo para compreender e solucionar problemas por meio da coleta e analisar dados, fazendo questionamentos e proporcionando uma conclusão.

Em um alto nível, a ciência dos dados é um conjunto de princípios, tais como Ciência da Computação, Matemática e Estatística, Especialização Científica, Design Gráfico, Aprendizado de Máquina, *Software* Tradicional e Pesquisa Tradicional, que apoiam e orientam a extração da informação e do conhecimento dos dados. De acordo com (AMARAL, 2016), a ciência de dados é formada pelas seguintes etapas: Produção de Dados, Armazenamento, Armazenamento Analítico, Análise de Dados e Visualização de Dados. A imagem a seguir mostra essa visão panorâmica e a relação entre etapas, com exemplos de tecnologias, plataformas e técnicas existentes.

Figura 3 – Visão panorâmica da ciência de dados

Ciência de Dados	Produção de Dados	Satélites	Transportes
		Redes sociais	Smart Phones
	Armazenamento	Relacional	noSQL
		Não estruturado Semi estruturado	Mainframes
	Armazenamento Analítico	Data warehouse	Data mart
	Análise de Dados	Exploratório	Explícita
		Implícita	
Visualização de Dados		Cubos	Dashboards
		Relatórios	KPI's

Fonte: Amaral (2016)

- Produção de Dados: dados são unidades individuais de informações geradas a partir de redes sociais, celulares, satélites e demais meios de transporte de informação.
- Armazenamento: é a persistência de dados, permitindo revisitar cada unidade individual de informação a qualquer momento.
- Armazenamento Analítico: trata-se da persistência de dados dentro de um grupo específico, isto é, um repositório de possível conhecimento orientado para uma linha de negócio específico, organização, empresa, etc.
- Análise de Dados: é o estudo e transformação dos dados em informação e significado, possibilitando extrair conhecimento a partir de informação que inicialmente não tinham significado.
- Visualização de Dados: em linhas gerais, trata-se da apresentação das informações, seja ela por meio digitais, pintura, escrito em papel, etc.

A Ciência de Dados, se comparada com *Business Intelligence*, seria o equivalente a uma versão atualizada (CAVIQUE, 2014), isto é, enquanto *Business Intelligence* auxilia no estudo de dados para tomada de decisão, a Ciência de Dados possibilita a predição das próximas ações (FABRIS, 2017). Assim, um cientista de dados precisa ser capaz de visualizar problemas nos negócios à partir da perspectiva de dados.

Provost (2013) afirma que, a área de Ciência de Dados contempla diversos outros campos tradicionais do estudo, como já citado anteriormente por Amaral (2016), e que é fundamental a compreensão de alguns princípios básicos para que o cientista consiga ter um pensamento analítico dos dados. Entre os princípios básicos, é fundamental que o profissional domine os tradicionais estudos no campo das estatísticas. Além disso, conhecer métodos e metodologias para visualizar dados auxilia no entendimento dos dados. Por fim, não basta apenas entender as técnicas, também é preciso ter um olhar criativo e seguir intuições para extrair conhecimentos úteis dos dados analisados.

Além do ramo acadêmico, como já citado, e do setor varejistas, o uso da disciplina de ciência de dados também está sendo usado para a recomendação de conteúdo em plataformas *online*. Este conteúdo pode ser desdobrado em diversas outras áreas, como por exemplo, a recomendação de filmes proposta pela *Netflix* e abordada por Postmus (2018), a de alimentos pelo autor Manoli (2015) em seu artigo *FoodR*, que tem o objetivo de recomendar pratos com base no cruzamento de informações entre usuários semelhantes, e até mesmo a indicação de restaurantes baseado em suas características, como é o caso apresentado por Sawant (2013), que utiliza a base de dados da *Yelp* para fazer as recomendações.

2.2 Aprendizado de máquina

Nos últimos anos, a Inteligência Artificial (IA) ganhou um papel de protagonista no cenário mundial de estudos relacionados à computação. Mesmo sendo esta uma nova notoriedade vinda à tona somente recentemente, o estudo de (IA) não é algo tão novo assim. As pesquisas foram iniciadas logo após o fim da Segunda Guerra Mundial e o seu nome foi definido no final da década de 50 (RUSSELL; NORVIG, 2004). Os métodos de aprendizagem de máquina têm sido utilizados em diversas aplicações como veículos autônomos que aprendem a dirigir em vias expressas, reconhecimento da fala, detecção de fraudes em cartões de crédito, estratégias para a construção de jogos, programas de mineração de dados que descobrem regras gerais em grandes bases de dados, etc.

Aprendizado é a capacidade de se adaptar, de modificar e melhorar seu comportamento e suas respostas, sendo, portanto uma das propriedades mais importantes dos seres ditos inteligentes sejam eles humanos ou não (RUSSELL; NORVIG, 2004). Já na área de inteligência artificial, existem diferentes abordagens para o aprendizado de máquina, entre elas, temos: aprendizado baseada em instâncias, por regras de associação, por representação, por similaridade e métrica, entre outras. As formas mais adotadas são aprendizados por modelos supervisionados, não supervisionados e semi supervisionados, também conhecido como aprendizado por reforço (HONDA; FACURE; YAOHAO, 2017).

- Aprendizado supervisionado: são modelos onde os dados a serem treinados já estão rotulados, isto é, existe uma entrada pelo qual a saída desejada é conhecida.
- Aprendizado não supervisionado: são modelos onde os dados a serem treinados não estão rotulados, isto é, não existe uma saída desejada conhecida, sendo necessário o uso de técnicas tais como o mapeamento por proximidade ou agrupamento por similaridade para que seja possível rotular dados.
- Aprendizado semi supervisionados: são modelos onde os dados a serem treinados são submetidos a uma validação externa com o intuito de garantir a rotulação correta dos dados, isto é, inicialmente não existe uma saída desejada, ela é obtida por meio da validação externa.

2.3 Modelos de aprendizado de máquina

O aprendizado de máquina, como já foi apresentado, pode ser entendida como um sistema computacional que tem o objetivo primordial de realizar uma tarefa TT, a partir de uma experiência EE, dessa forma, melhorando a performance PP.

De uma forma mais resumida, um algoritmo consegue aprender o que está sendo desenvolvido ao seu redor e chega a um objetivo de grande volume de dados, ou seja, realiza

as suas experiências. Um exemplo do que foi apresentado pode ser observado pela solução proposta pela empresa *Betegy*, pelo qual foi preciso prever um resultado de uma partida de futebol⁴. Para prever o resultado da partida, foi analisado dados sobre a composição do time, técnico, formação, resultado das últimas partidas, etc. Dessa forma, o computador iria analisar todos os dados em forma de pares que são os resultados e as variáveis, fazendo com que ele aprenda os padrões que levam a vitória de um dos times, além de generalizar as situações que ainda não foram observadas. Logo, quanto mais dados o computador tiver para trabalhar, mais experiências serão obtidas e os resultados melhorados (HONDA; FACURE; YAOHAO, 2017).

2.3.1 Aprendizado supervisionado

O primeiro modelo a ser discutido é o aprendizado de máquina supervisionado, no qual é preciso tentar prever uma variável dependente analisando uma lista de variáveis independentes, conforme o exemplo abaixo pode demonstrar este acontecimento:

Figura 4 – Variáveis dependentes e independentes

Var. Independentes	Var. Dependentes
Anos de Carreira, Formação, Idade	Salário
Idade do Carro, Idade do Motorista	Risco de Acidente Automotivo
Texto de um livro	Escola Literária
Temperatura	Receita de venda de sorvete
Imagen da Rodovia	Ângulo da direção de um carro autônomo
Histórico escolar	nota no ENEM

Fonte: Honda; Facure; Yaohao (2017)

Uma variável dependente, tal como nota no ENEM, como mostra a figura apresentada, representa um dado pelo qual o valor depende de uma ou mais variáveis independentes, sendo para o exemplo apresentado, histórico escolar o dado independente a ser estudado.

Ao analisar este ato, é notável que os dados utilizados para treinar o computador sempre apresentam uma resposta desejada em vista do sistema supervisionado, ou seja, há a variável dependente que resulta a partir da criação das variáveis independentes e as suas devidas observações. Logo, os dados são anotados com as classes ou respostas que devem ser previstas pelo estudo feito no computador.

Diversas são as técnicas utilizadas para que se possam resolver os problemas de aprendizagem supervisionada (HONDA; FACURE; YAOHAO, 2017). As mais conhecidas são: *artificial neural networks*, *decision tree*, *vettorial support*, *k-nearest neighbors* e *naive bayes*. O modelo de aprendizagem de máquina supervisionado, portanto, é a área no qual

⁴ Predição no futebol: https://betegy.com/betegy_b2b_solution.pdf

existem as mais diversas aplicações bem sucedidas e a maioria dos problemas já estão definidos (HONDA; FACURE; YAOHAO, 2017).

2.3.2 Aprendizado não supervisionado

Por mais interessante que seja o modelo de aprendizagem supervisionado de máquina, nem todos os problemas podem e devem ser resolvidos por intermédio deste ato. Em determinados casos, como será visto, é praticamente impossível (ou extremamente custoso) conseguir dados que já estão anotados (HONDA; FACURE; YAOHAO, 2017).

Dessa forma, é preciso analisar o seguinte exemplo: “Você é o dono de um grande comércio e, a partir disto, está a fim de descobrir o perfil dos consumidores que são os seus clientes e para que isso seja possível, há a possibilidade de existirem clientes que comprem vinho e queijo, outros carne e carvão e alguns até mesmo apenas leite em pó e fralda, no qual caracteriza uma linha de produtos, dessa forma, você pode colocar os produtos longes um dos outros (o que fará com que o cliente passe mais tempo dentro do supermercado), mas ainda assim não estará sendo anotado para qual compra e à qual perfil o consumidor pertence e nem sabemos quais são os perfis consumidores que existem”.

Logo, o computador terá em seu escopo o objetivo de descobrir quais são os perfis sem os dados anotados dos clientes, fazendo com que sejam utilizados os métodos de aprendizado não supervisionados. Para isto, a opção mais viável é a de observar os registros de compras repetitivas e, a partir deste ato, inferir tais dados a um grupo de consumidores ou um consumidor em especial (HONDA; FACURE; YAOHAO, 2017). Outra solução para esta problemática é de analisar diretamente os produtos que são sempre comprados juntos (como cerveja carne e carvão ou leite em pó, fraldas e brinquedos) e a partir disto aprender uma regra associativa entre eles.

Em um âmbito geral, portanto, no modelo de aprendizagem não supervisionado o objetivo primordial é de achar uma representação que proporciona mais informação acerca dos dados que existem, fazendo com que sejam observados os pontos relevantes para este ato se tornar mais simples, como demonstra o exemplo abaixo:

Figura 5 – Exemplo de dados não supervisionados

Dados	Forma Representativa
Transações bancárias	Normalidade da transação
Registros de Compras	Associação entre produtos
Dados Multidimensionais	Dados com dimensão reduzida
Registros de Compras	Perfil dos consumidores
Palavras em um texto	Representação matemática das palavras

Fonte: Honda; Facure; Yaohao (2017)

Os sistemas de recomendação de filmes, músicas, restaurantes (o que está sendo discutido neste texto), detecção de anomalias e visualização de dados são os exemplos mais conhecidos de aprendizagem não supervisionada. Os problemas que eles proporcionam, na maior parte das vezes, são consideravelmente mais complicados do que os problemas supervisionados em vista de que nesta modalidade não há uma resposta anotada nos dados. Logo, é extremamente complicado e controverso pegar todo um modelo de aprendizagem que não é supervisionado e o avaliar na aprendizagem de máquinas.

2.3.3 Aprendizado semi supervisionada

Ao longo deste trabalho já foram apresentadas duas modalidades de aprendizagem e agora, salientando sobre a terceira, ela é chamada de semi supervisionada (ou reforço), no qual a máquina tem como objetivo primordial aprender a melhor decisão a se tomar em vista das circunstâncias que giram a problemática a ser executada.

Como bem se sabe, o futuro é sempre uma variável aleatória, pois não há o conhecimento de quando tal ato irá acontecer, se acontecerá ou se é viável acontecer, fato. Logo, é necessário que haja uma abordagem que entenda toda esta problemática (a incerteza constante) e a partir disso analise todas as eventuais mudanças ambientais que podem acontecer no processo de tomada da melhor decisão. Levando em consideração esta ideia, ela é desenvolvida a partir da aprendizagem por reforço (que é feita na psicologia), no qual há sempre uma punição ou recompensa dada a um agente acerca da decisão que é tomada, por exemplo, no adestramento de animais (PAVLOV, 2013). Durante este processo, o adestrador avalia o animal e quando ele realiza o que é determinado, é recompensado. Quando não, punido. Logo, em vista do anseio de sempre receber recompensas, o agente tenta associar as ações de acordo com o ambiente que é apresentado, evitando recompensas menores ou punições.

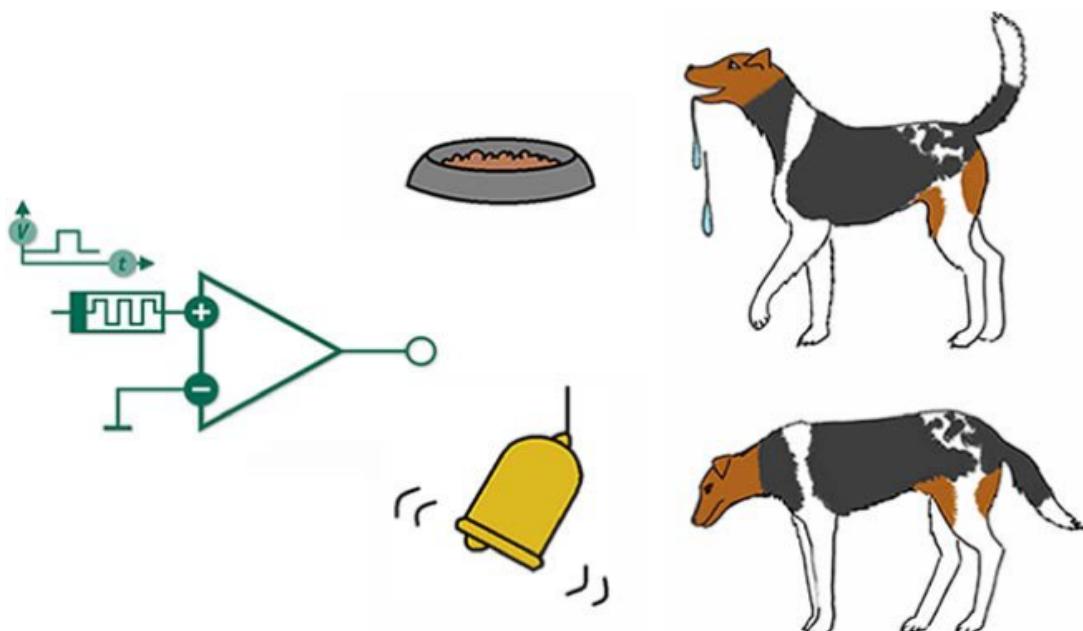
No modelo de aprendizagem de máquinas semi supervisionado, como é o caso deste texto em questão, a máquina tenta observar todo o “estado da natureza” e os conjuntos de cenários futuros possíveis para que, dessa forma, consiga analisar qual ação a se tomar é mais viável buscando uma recompensa por este ato, no caso, a informação desta combinação (HONDA; FACURE; YAOHAO, 2017). O processo é realizado inúmeras vezes para que seja possível analisar a melhor decisão a ser tomada em cada cenário possível que foi observado e pode acontecer no futuro.

Recapitulando a ideia proposta anteriormente, com o adestramento do cachorro, é interessante analisar a seguinte situação: Utilizando a abordagem semi supervisionada, é necessário fazer com que o cachorro obedeça a um comando e sente ao interpretá-lo. A princípio é quase impossível alguém fazer com que o animal execute o comando de primeira, levando em consideração de ele nunca ter tido esta prática (PAVLOV, 2013), no entanto, é necessário o repreender verbalmente, facial, ou com uma punição de pancada de

jornal (algo que não o machuque, mas o faça entender que ele está errado). Incentivando-o a executar o comando de forma correta, o cão pode receber sinais de aprovação ao se aproximar do ato e finalmente ser recompensado ao realiza-lo, por exemplo, ao sentar ele pode receber um biscoitinho (HONDA; FACURE; YAOHAO, 2017). Sendo o experimento realizado diversas vezes, com o tempo o cão consegue analisar a relação entre “causa-efeito”, no qual ele aprenderá a obedecer ao comando em vista da recompensa ou punição que ele receberá. Pavlov (2013) tem um experimento muito interessante no qual apresenta este paradigma de aprendizagem.

O russo apresentou em suas obras e experimentos a ideia do reflexo condicionado, no qual há o experimento: com um pedaço de carne, o cão começa a salivar (pois deseja o alimento). Pegando este pedaço de carne e soando uma campainha, com muita repetição, o cão associa a carne e a campainha, logo, ele salivava toda vez que ouvia o barulho da campainha e ia em busca da sua carne, representando um experimento que é associável ao sistema de aprendizagem por reforço (semi supervisionado).

Figura 6 – Experimento de Pavlov



Fonte: Honda; Facure; Yaohao (2017)

Utilizando este ideal proposto por Pavlov (2013) e o adicionando a ciência de dados, por exemplo, ao invés de adestrar cachorros seria necessário construir uma máquina de portfólios que trabalha no mercado financeiro e tem o objetivo de ajustar a combinação de ativos vendidos/comprados a depender do retorno financeiro “recompensa”, seria possível fazer com que a aprendizagem funcionasse e houvesse evolução do mercado (HONDA; FACURE; YAOHAO, 2017). Outro exemplo que pode ser associado são dos automóveis que dirigem “sozinhos”, pois eles são desenvolvidos para analisar todo o cenário em volta e tomar a melhor decisão, sendo que recebem estímulos negativos ao colidirem

com veículos/obstáculos, aprendendo, dessa forma, a os contornar e se manter de forma autônoma.

Logo, levando em consideração a aprendizagem por reforço em um âmbito mais avançado, fazer com que a máquina encontre o meio termo ótimo entre a “*exploration* e a *exploitation*” é uma tarefa muito difícil de ser realizada em vista das aplicações mais complexas como, por exemplo, fazer com que uma máquina jogue xadrez (é preciso para um bom jogador ganhar uma partida, analisar as suas jogadas e os acontecimentos futuros que podem ser concebidos a partir dela) (HONDA; FACURE; YAOHAO, 2017). Logo, neste sistema de aprendizagem levando em consideração as recomendações de restaurante, serão analisados todos os meios possíveis e os resultados que eles possam apresentar, buscando realizar propostas viáveis e que gerem “recompensas” para as pessoas que dirigem este processo através da aprendizagem de máquinas.

2.3.4 Síntese: qual é o melhor modelo?

Não existe uma regra fixa para todos os tipos de recomendação, uma vez que elas dependem de tipos diferentes de dados e chegam à diferentes conclusões (HONDA; FACURE; YAOHAO, 2017).

Como visto anteriormente, método não supervisionado, considera atributos e não necessariamente sabe o que esperar na sua saída, dificultando então a separação em grupos de forma tão eficaz como no método supervisionado. O algoritmo então faz um agrupamento por semelhantes, mas sem a certeza que o agrupamento é o esperado.

Já o método supervisionado sabe de antemão qual é a saída esperada dada as características recebidas. Então, no momento que um dado novo é obtido o algoritmo já tem as informações necessárias para inseri-lo em seu grupo determinado.

Levando em consideração o contexto deste trabalho e as tecnologias atuais, a opção que mais se adapta para a realização de recomendações de restaurantes é o modelo semi supervisionado, pois é um método que une os demais ao utilizar a validação externa do usuário para ter certeza que a sugestão foi eficaz ou não, e inseri-la de forma automática em seu novo grupo, uma vez que ela já sabe a sua saída. Fazendo assim, com que as sugestões futuras sejam cada vez mais precisas.

2.4 Sistemas de recomendação

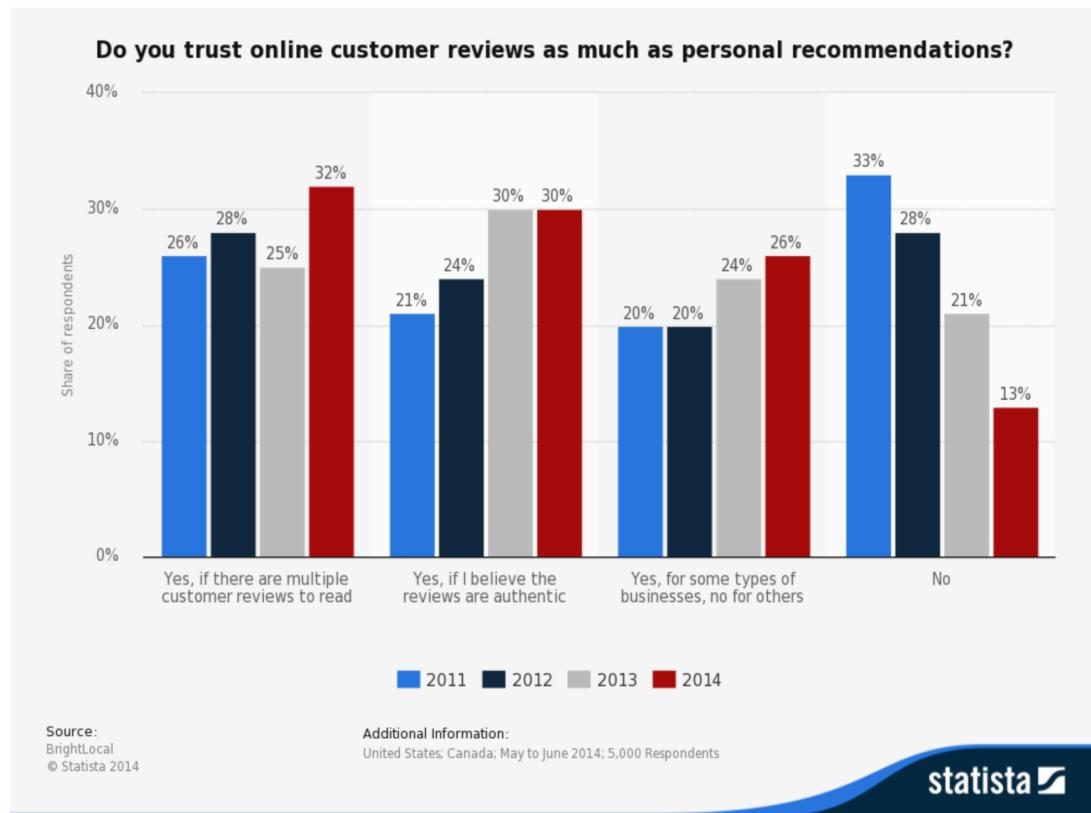
Os sistemas de recomendação estão sendo utilizados em uma variedade de áreas e são comumente encontrados na maioria dos geradores de *playlist* de serviços de música como *Spotify* ou em serviços de *streaming* de vídeo como *Netflix* e *Amazon* (RICCI, 2011). Estes sistemas podem operar usando uma única entrada de dado, como música por exemplo,

ou utilizando múltiplas entradas. Há também modelos populares para tópicos específicos como restaurantes e namoros *online*.

Estes sistemas geralmente fazem uso de filtragem colaborativa e filtragem baseado em conteúdo para recomendar dados. Filtragem colaborativa é orientado a construir um modelo baseado no comportamento passado de um ou mais usuários (MELVILLE, 2010). Já a filtragem baseado em conteúdo utiliza uma série de valores discretos, características pré-determinadas de um item para recomendar outros itens com propriedades similares (MOONEY, 1999).

A figura 7 a seguir exibe um gráfico comparando a confiança dos clientes em relação a recomendação baseada em filtragem colaborativa. O gráfico traz uma visão ano contra ano, sendo possível ver um crescimento relevante de clientes que utilizam de avaliações de terceiros como recomendação pessoal. Isso mostra a importância e relevância que a filtragem colaborativa têm no momento de recomendar conteúdo.

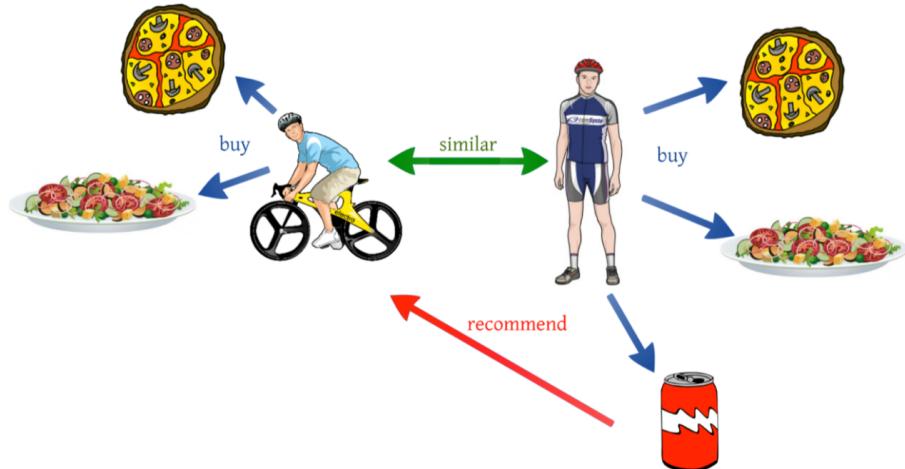
Figura 7 – *Do you trust online customer reviews as much as personal recommendations?*



A figura 8 a seguir exibe um sistema de recomendação em alto nível, comparando dois ciclistas que efetuaram a compra de um prato de *pizza* e uma salada em um restaurante, entretanto, apenas um deles efetuou a compra de um refrigerante. Com base no histórico de compras de cada ciclista, o sistema encontrou uma similaridade, isto é, ambos os perfis estavam muito próximos, com exceção do refrigerante que apenas um deles comprou.

Ampliando a ideia dos ciclistas, imagine que ao invés de dois, existem diversos outros que apresentam o mesmo padrão encontrado, a partir desta análise ampliada, é possível recomendar a compra do refrigerante para o ciclista que não havia comprado.

Figura 8 – Recomendação de produto baseado em similaridade



Fonte: Manoli (2015)

O mesmo padrão pode ser observado na recomendação de restaurantes, como explica (SAWANT, 2013) em seu sistema de recomendação. Em geral, os estabelecimentos possuem similaridades entre eles, seja pela culinária, preço médio ou simplesmente a avaliação do restaurante, é possível encontrar padrões, permitindo assim recomendar novas experiências para os clientes a partir de seus históricos.

2.4.1 Sistemas de recomendação em grupo

Muitas atividades, assim como assistir um filme ou jantar, são feitas acompanhadas. Um sistema de recomendações em grupo tem como objetivo recomendar um item para um grupo de pessoas. Isto frequentemente é de difícil medida de efetividade quanto a uma recomendação, já que todos os membros do grupo poderão ter opiniões diferentes sobre o item recomendado (BERKOVSKY; CANTADOR; TIKK, 2018). Se por exemplo é aniversário de um dos membros do grupo é razoável que ele ou ela possa não se desapontar com o item escolhido, e sua conceituação possa ter um peso maior (BERKOVSKY; CANTADOR; TIKK, 2018).

Algoritmos de aprendizagem de máquinas podem analisar dinâmicas de grupos e construir modelos de preferências de grupo. Para a agregação de conceituação há várias abordagens, um subconjunto delas seria: média, mínima pobreza, desacordo médio e mais prazer (BERKOVSKY; CANTADOR; TIKK, 2018). Estas são todas abordagens psicológicas e são frequentemente agregadas à conceituação final:

- Média - É a abordagem padrão, esta maximiza a felicidade em um grupo. A desvantagem deste método é que quando um usuário tem um grande apreço por um item e os outros não gostam deste item, ele pode ser conceituado bem acima da média;
- Mínima pobreza – É a abordagem que minimiza a insatisfação individual no grupo. A desvantagem deste método é que se um item tem um alto conceito no grupo, mas com um elemento que tenha um conceito mais baixo pode levar a um desempenho viciado;
- Desacordo médio – É justo para todos os membros do grupo se um item for selecionado pela maioria ele é mais bem conceituado. Assim minimizando desacordos sobre os itens;
- Mais Prazer – Esta medida é oposta a mínima pobreza. Ela tenta maximizar a satisfação do grupo.

2.5 K-means Clustering

O algoritmo *K-means* (também conhecido como K-médias) tem o objetivo de fornecer uma classificação de informações de acordo com os próprios dados. Moore (2001) explica em seu texto (*K-means and Hierarchical Clustering*) que essa classificação é baseada em comparações entre os valores numéricos da amostra informada. Além disso, esse método trabalha com o conceito de agrupamento de dados segundo o seu grau de semelhança, e é considerado um algoritmo de aprendizado não supervisionado. Isto por que ele consegue fornecer uma classificação dos dados sem nenhuma pré-classificação existente, ou seja, sem a necessidade de supervisão humana.

Outro ponto interessante do *K-means* é que ele permite não só trabalhar com dados contínuos, mas também com valores discretos, desde que eles sejam mapeados para valores numéricos. Este processo de transformação geralmente ocorre durante a normalização dos dados. O algoritmo original utiliza a distância euclidiana para calcular o quanto distante uma instância está da outra. Porém, existem variações deste método que usam técnicas diferentes para obter essa distância, entre elas, as mais conhecidas são: *Minkowski*, *Manhattan* e *Kernelized* (MOORE, 2001).

Finalizado a etapa para obter as distâncias, o algoritmo calcula centróides para cada uma das classes. A cada interação o valor das centróides é refinado. Esse refinamento ocorre pela média dos valores de cada atributo de cada ocorrência pertencente a centróide. Por fim, é gerado k centróides e organiza as ocorrências da tabela de entrada conforme a sua distância dos centróides (MOORE, 2001). O algoritmo *K-means* pode ser representado da seguinte maneira:

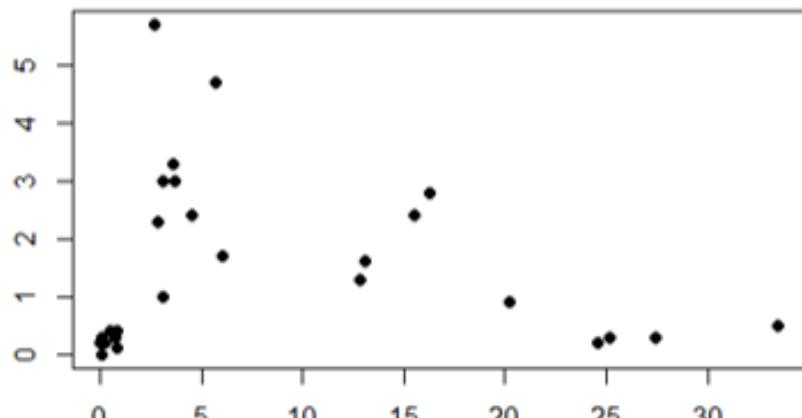
- Etapa 1: Fornecer valores para os centróides;
- Etapa 2: Gerar uma matriz de distância entre cada ponto e os centróides;
- Etapa 3: Colocar cada ponto nas classes de acordo com a sua distância do centróide da classe;
- Etapa 4: Calcular os novos centróides para cada classe;
- Etapa 5: Repetir até a convergência.

A distância euclidiana entre os pontos $P = (p_1, p_2, \dots, p_n)$ e $Q = (q_1, q_2, \dots, q_n)$, em um espaço euclidiano de n-dimensional, é dada pelo somatório:

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

Para exemplificar, suponha uma amostra disposta nos eixo X e Y, conforme mostra a figura 9. O objetivo é agrupar estes dados baseado em sua similaridades, ou seja, quanto mais próximos os dados estiverem espalhados no gráfico, maior é a similaridade entre eles.

Figura 9 – Amostra de dados dispostas nos eixo X e Y



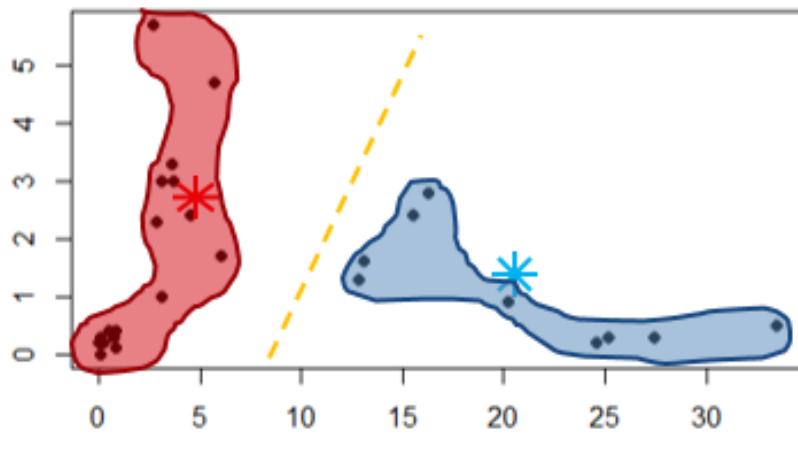
Fonte: Nogare (2015)

Observe que no gráfico da figura 9 é possível efetuar diversas separação da amostra, bastaria considerar o número de grupos e separar de tal forma que os pontos estejam o mais próximos de cada grupo. Após a aplicação do algoritmo *K-means*, o método separou os dados em dois grandes *cluster*, como mostra a figura 10.

É possível analisar na figura 10 que, após a aplicação da técnica, dois grandes grupos foram destacados na imagem, o vermelho, mais a esquerda da figura 10 e um azul, mais a direita. Além disso, observe que nesta mesma imagem existe dois asteriscos destacados, também nas cores vermelho e azul. Esses dois pontos são denominados de

centróides. Cada *cluster* foi criado a partir do cálculo da distância entre essas *centróides*. O método executa diversas rodadas, onde em cada uma delas é recalculado o valor das *centróides*. Enquanto o algoritmo não atingir o número de grupos solicitados, as *centróides* são atualizadas no gráfico e uma nova rodada é iniciada. Ao final do processo, será obtido o total de grupos solicitados ao método (NOGARE, 2015).

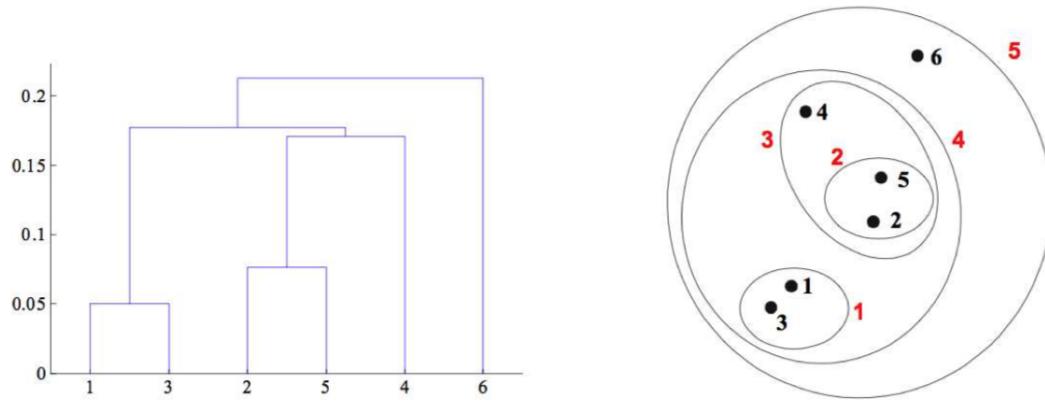
Figura 10 – Amostra de dados dispostas nos eixo X e Y com aplicação do K-means



Fonte: Nogare (2015)

O método *K-means* permite que as partições possam ser visualizadas usando uma estrutura de árvore. Além disso, é possível ver partições em diferentes níveis de granularidade. Esse recurso é conhecido como *Hierarchical Clustering*. A figura 11 mostra como essa visualização ocorre:

Figura 11 – Exemplo de visualização em estrutura de árvore



Fonte: Nogare (2015)

É importante ressaltar que esse tipo de visualização facilita enxergar a distribuição dos dados em cada *cluster*. Isto é, esse padrão de gráfico em forma de árvore ajuda o profissional visualizar se o total de grupos gerados é suficiente ou se a divisão está gerando ruído na classificação. Para entender melhor esse ruído, imagine que na figura 10 existam mais pontos espalhados pelo gráfico. Além disso, suponha que eles estão no plano $X = 5$ e

$25 < Y \leq 30$. Aplicando o método *K-means* nessa suposta amostra e mantendo o total de *cluster* igual a dois, o algoritmo provavelmente irá classificar os novos pontos como sendo do grupo azul, uma vez que ele é o mais próximo da nova instância. Entretanto, apesar de ser o grupo mais próximo, é evidente pelo gráfico que os novos pontos pertencem na verdade a um outro *cluster* não definido na análise, gerando um ruído no resultado final.

O cálculo da distância de *Manhattan* citado nos parágrafos anteriores também é conhecida como a geometria do táxi. Essa técnica foi abordada por Hermann Minkowski no século XIX, em que a distância entre dois pontos que usualmente era calculada pelo método de euclidiana, passava a ser obtida pela soma das diferenças absolutas de suas coordenadas. Esse cálculo é dado pelo somatório:

$$d(p_n - q_n) = \sqrt{\sum_{i=1}^n |p_i - q_i|}$$

Já a distância de Minkowski de ordem P entre dois pontos $X = (x_1, x_2, \dots, x_n)$ e $Y = (y_1, y_2, \dots, y_n)$ é dada por:

$$D(X, Y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

Moore (2001) ainda apresenta um método de distância pela correlação de *Pearson*, também conhecido como coeficiente de correlação produto-momento. Esta técnica consiste em medir o grau de correlação entre duas variáveis de escala métrica. Este coeficiente, geralmente representado pela letra ρ assume apenas valores entre -1 e 1. Para $\rho = 1$, a correlação é perfeita e positiva entre as duas variáveis. Para $\rho = -1$, a correlação é perfeita e negativa entre as duas variáveis, ou seja, se uma aumenta, a outra sempre diminui. Por fim, para $\rho = 0$, as duas variáveis não dependem linearmente uma da outra. Neste caso, deve ser investigado por outros meios, uma vez que pode existir uma dependência não linear entre elas (NOGARE, 2015).

O coeficiente de correlação de *Pearson* é calculado pela função ρ , onde os valores medidos de ambas as variáveis é dado por x_1, x_2, \dots, x_n e y_1, y_2, \dots, y_n , e as médias aritméticas de ambas as variáveis é obtidas por \bar{x} e \bar{y} , como mostra a fórmula a seguir:

$$\rho = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i \quad \text{and} \quad \bar{y} = \frac{1}{n} \cdot \sum_{i=1}^n y_i$$

2.6 O método Elbow

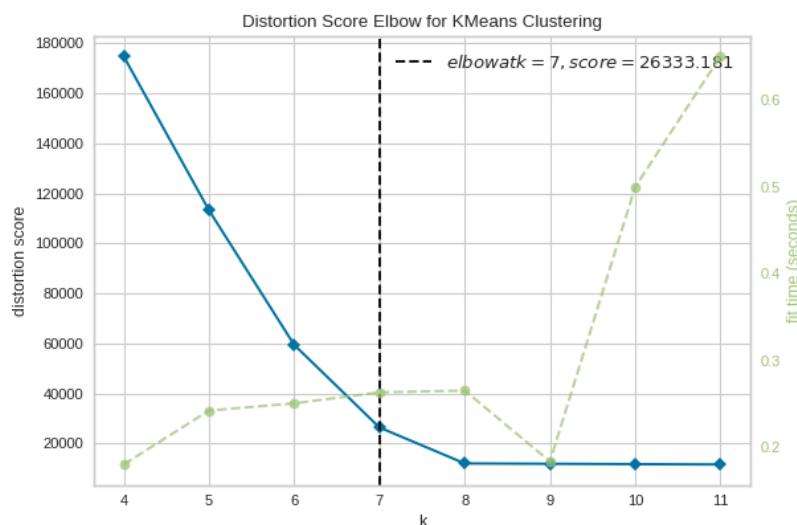
O *Clustering*, também chamado de *Elbow Method* ou Método do Cotovelo em tradução livre, é uma metodologia de interpretação e validação estabelecida para auxiliar o encontro do número apropriados em um *dataset* (THORNDIKE, 1953). O método tem esse nome por se parecer com um braço e formar um ângulo semelhante a um “cotovelo” para definir os números aceitáveis de *clusters* a serem gerados com base nos dados amostrados.

A lógica por trás da metodologia funciona para analisar a porcentagem da variação como uma função pelo número de *clusters*. Ou seja, ao se traçar a porcentagem de variação pelos *clusters* em relação ao seu número, o resultado trará uma variação muito alta em primeiro momento, mas após alguns pontos, esse número cai formando assim o perceptível ângulo de “cotovelo” (THORNDIKE, 1953).

A partir de um *cluster*, o método vai aumentando a quantidade de K's e trazendo assim melhorias a cada incremento. No momento em que a distância entre uma quantidade de *clusters* é próxima da outra, é entendido que o algoritmo torna-se relevante com aquele valor de *clusters*, devendo usar tal número para a segmentação de dados no gráfico.

A figura 12 mostra um exemplo do método sendo aplicado pelo algoritmo *K-means*. Na figura é possível ver que a configuração ideal para o exemplo dado é utilizando apenas 4 *clusters*, onde a variação é maior entre os dados, permitindo assim um melhor desempenho na recomendação.

Figura 12 – Método Elbow utilizando K-means



Fonte: Pedregosa et al. (2011)

3 Projeto

Este capítulo apresenta as atividades realizadas no desenvolvimento do trabalho. Inicialmente é introduzida uma visão geral de alto nível da arquitetura do sistema, que se inicia na base de dados e se estende até a plataforma *mobile*. Em seguida é apresentado como os dados foram obtidos e traz também uma visão das diversas possibilidades de desenvolvimento e análise dessas informações. O capítulo também explica como foi realizado o pré-processamento dos dados, desde categorizar a normalizar os valores da base. Por fim, é descrito o funcionamento do sistemas de recomendação criado, que foi baseado em uma plataforma de reserva *online* de restaurantes.

A base de dados utilizada no trabalho foi extraída do site <https://www.chefsclub.com.br/restaurantes> em outubro de 2019 e contém dados no período de 11 de julho de 2012 até o presente momento deste trabalho. Os dados foram extraídos mediante a autorização da empresa. Ao todo foram obtidos 8.052 restaurantes, 79.634 ofertas diferentes, 64 culinárias, 22 cidades e regiões e mais de 900 bairros, como pode ser visto na seção 3.2.

3.1 Visão geral

O principal objetivo deste trabalho é o desenvolvimento de um sistema de recomendação de restaurantes a partir da análise de grande volume de dados. O projeto implementado visa comprovar que por meio das técnicas oferecidas pela área Ciência dos Dados, é possível identificar padrões e similaridades entre restaurantes e preferências de clientes, sendo elas o histórico de reservas dos clientes e restaurantes favoritos, possibilitando assim recomendar novas experiências para os usuários.

O trabalho também aborda a utilização de um sistema de recomendação com modelo semi supervisionado (aprendizado por reforço), contextualizado anteriormente no capítulo 2, onde há a necessidade da ação externa do usuário para validar o que foi recomendado e a partir dessa validação, gerar uma nova base de sugestões com maior grau de similaridade.

A figura 13 apresenta uma visão geral do projeto e o fluxo de trabalho por trás do sistema desenvolvido. Ao todo existem 5 passos destacados pela cor azul, sendo eles, Base de dados, Normalização, Analisador, Resultados, Aplicativo. A imagem também traz duas etapas em destaque que é caracterizado pelas cores amarelo e vermelho, no qual se referem a ação externa entre usuário e máquina, explicado posteriormente na subseção 3.6.3. O processo é cíclico e se dá início com a base de dados e termina novamente na base com os dados atualizados, produzidos pela interação do usuário com o aplicativo.

Figura 13 – Representação cíclica do fluxo de recomendação



Fonte: Produção do próprio autor

- 1. Base de dados: inicialmente a base foi construída por meio da extração de dados de uma aplicação de *food service*, como já explicado anteriormente neste capítulo. Posteriormente, a base de dados passou a ser populada pela produção de novas informações geradas a partir do aplicativo desenvolvido neste trabalho.
- 2. Normalização: trata-se da eliminação de toda e qualquer informação incorreta ou fora do padrão da base. Vale lembrar que a base de dados foi extraída a partir de uma aplicação *mobile* de reservas em restaurantes que está operando a mais de 5 anos no mercado de *food service*, ou seja, a base de dados carrega consigo uma herança de longa data, podendo assim conter erros. Entre os principais erros encontrados, dados nulos e valores atípicos foram os que mais se repetiram.
- 3. Analisador: etapa onde os dados são analisados e recomendados. Por serem dados inicialmente não supervisionados, é utilizado a técnica de clusterização, abordado por (KATARYA, 2017) em seu sistema de recomendação. Para efetuar a análise, foi escolhido o método *K-means*, conforme citado anteriormente na subseção 2.5 e explicado em detalhes nas próximas seções deste capítulo.
- 4. Resultados: trata-se do processo onde os dados recomendados são organizados em modelos de objetos. Este processo é realizado por meio do paradigma orientado a

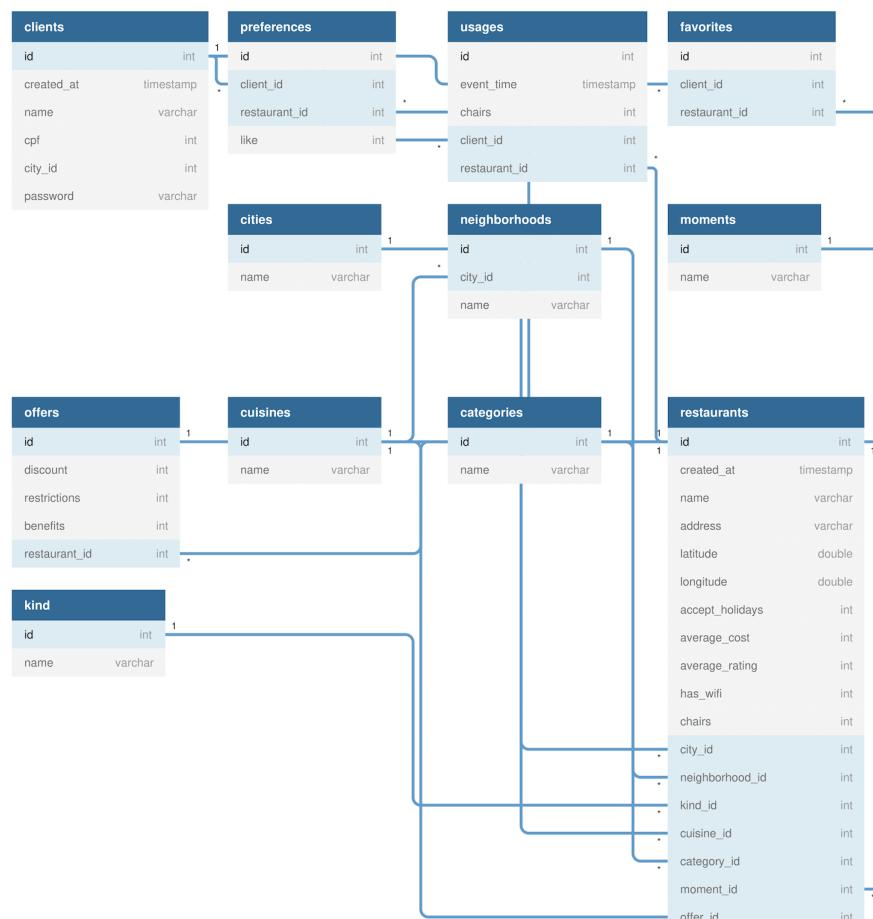
objetos¹. Os principais modelos utilizados no trabalho são: *restaurant*, *client*, *usages* e *favorites*, contextualizado melhor na seção 3.5.

- 5. Aplicativo: trata-se da experiência visível do trabalho. Utilizando a linguagem *Flutter* e *guidelines* de *Design Thinking*², a aplicação apresenta de forma clara, listas inteligentes de recomendações e demais telas.

3.2 Base de dados

Como já citado anteriormente, a base de dados do presente trabalho utiliza dados extraídos do site <https://www.chefsclub.com.br/restaurantes>. Os dados foram extraídos mediante a autorização da empresa. Ao todo foram criados 12 tabelas a partir dos dados obtidos para desenvolver o sistema de recomendação proposto. A extração dos dados foi realizada em outubro de 2019 e contém dados no período de 11 de julho de 2012, até o presente momento deste trabalho.

Figura 14 – Diagrama do banco de dados desenvolvido para o trabalho



Fonte: Produção do próprio autor

¹ <https://www.inf.pucrs.br/~gustavo/disciplinas/pli/material/paradigmas-aula12.pdf>

² <https://www.enterpriseirregulars.com/125085/what-is-design-thinking/>

Para criar o modelo *restaurant*, contextualizado na seção 3.5, foi necessário reunir o máximo de informações disponíveis, a fim de obter o maior número de características possíveis a serem utilizadas posteriormente no sistema de recomendação.

O foco do trabalho está na recomendação de restaurantes situados na cidade de São Paulo, muito por conta da quantidade de dados existentes na base para esta cidade. Ao todo existem 8.052 restaurantes cadastrados e distribuídos em 22 cidades e regiões diferentes na base. Desse total, 3.943 pertencem a cidade foco escolhida, representando 48,5% da base extraída e aproximadamente 31.5% dos restaurantes existentes em São Paulo até o ano de 2017³.

É importante ressaltar que apesar da escolha de uma cidade para fazer as análises durante o desenvolvimento, as demais cidades e regiões continuam disponíveis, sendo possível selecionar a que desejar no aplicativo, como é contextualizado na seção 3.6.

A seguir são apresentado as menores tabelas do projeto, sendo elas: *kind*, *moments* e *categories*. Os dados destas, estão respectivamente representadas nas tabelas 1, 2 e 3.

Tabela 1 – Distribuição de dados da tabela *kind*

id	name
1	booking
2	checkin

Tabela 2 – Distribuição de dados da tabela *moments*

id	name
1	Pra ir com os amigos
2	Pra ir com a família
3	Almoço de trabalho
4	Romântico

Tabela 3 – Distribuição de dados da tabela *categories*

id	name	id	name	id	name
1	À la carte	6	Executivo	11	Saudável
3	Rodízio	7	Cafeteria	12	Sobremesa
4	Comida rápida	8	Buffet	-	-
5	Food Truck	10	Bar	-	-

- *kind*: é utilizado para representar se um restaurante aceita *booking* ou *check-in*. A reserva (*booking*) é necessário fazer com antecedência, ou seja, antes de comparecer ao restaurante. Já no caso do *check-in*, assim como em um hotel, é realizado no momento em que o cliente chega ao restaurante.

³ <https://sao-paulo.estadao.com.br/blogs/edison-veiga/quantos-restaurantes-existem-em-sp>

- *moments*: é o que define o ambiente do restaurante, isto é, se é recomendado para ir com amigos ou um simples almoço de trabalho.
- *categories*: também é um dado que pertence ao modelo *restaurant*, e é utilizado para especificar o tipo de estabelecimento.

A seguir é apresentado a tabela de restaurantes, que está dividida em 3 partes pelas tabelas 4, 5 e 6. Além das características já apresentadas nas tabelas anteriores, essa traz outras informações que compõe o modelo *restaurant*. A tabela a seguir é a principal do projeto e é a mais acessada durante o processo de recomendação pois é onde está armazenado os dados de restaurante.

Tabela 4 – Distribuição de dados da tabela *restaurant*, parte 1

id	createdAt	name	address	chairs	cityId
193	2012-07-11	Shinjuku	R. Dr. Mário Ferraz	4	10
288	2012-08-16	Soul Fish	R. Manoel Guedes	8	10
356	2012-10-30	Fazenda	Avenida Morumbi	6	10
361	2012-10-30	Kiichi	Alameda Lorena	6	10

Tabela 5 – Distribuição de dados da tabela *restaurant*, parte 2

kindId	price	wifi	neighborhoodId	latitude	longitude
2	70	0	133	-23.5809784	-46.6874504
1	120	0	133	-23.5845699	-46.6791916
1	95	0	250	-23.6131992	-46.7139015
1	60	0	17	-23.5728016	-46.6563721

Tabela 6 – Distribuição de dados da tabela *restaurant*, parte 3

offerId	rating	cuisineId	holidays	momentId	categoryId
10804	5	36	1	2	6
10899	5	36	1	2	1
10968	5	4	1	2	1
10973	5	36	1	2	1

É possível notar que alguns dados na tabela apresentada não são tão intuitivos de serem compreendidos, pois estão ligados diretamente a regras de negócio da aplicação pela qual produziu os dados exibidos. São eles: *chairs*, *wifi*, *offerId*, *rating* e *holiday*.

É importante ressaltar que para a aplicação desenvolvida, algumas dessas regras de negócios foram abstraídas, uma vez que a aplicação apenas simula o sistema de reservas *online*. Um exemplo dessa abstração é a aplicada em *chairs*. Originalmente, o usuário deve selecionar o total de pessoas na mesa, entretanto, para o projeto desenvolvido, pouco importa essa informação, aplicando assim a abstração de sempre considerar o número máximo de pessoas que o restaurante suporta por reserva.

- *chairs*: é utilizado para representar o número total de cadeiras que podem ser ocupadas no restaurante pelo aplicativo.
- *wifi*: define se o ambiente do restaurante tem acesso a *wifi* gratuito aos clientes.
- *offerId*: esse dado é um identificado da tabela *offers* e é responsável em associar o restaurante a uma oferta válida. Uma oferta possui desconto, que varia de 10% até 50%, restrição, que pode impedir por exemplo que o desconto seja aplicado em alguma promoção interna do restaurante e por fim, benefício, isto é, além do desconto obtido, o cliente também pode ganhar algo extra do restaurante, como por exemplo, uma sobremesa grátis.
- *rating*: é a avaliação média do restaurante. O número é obtido a partir da média de 3 outras notas que são referentes ao ambiente, comida e atendimento. Para o trabalho desenvolvido, é usado apenas o dado de avaliação média e não é possível popular o sistema com novas avaliações, sendo assim, uma melhoria no sistema para trabalhos futuros.
- *holiday*: é utilizado para informar se o restaurante aceita a oferta associada a ele durante feriados e datas comemorativas.

Apresentado anteriormente, ao todo existem 22 cidades e regiões diferentes cadastradas na base. A tabela 7 traz a visão de todas elas ao lado do número total de restaurantes disponíveis. Como já falado, a cidade de São Paulo foi a utilizada durante o projeto para realização das recomendações e testes, uma vez que ela é a cidade com maior quantidade de restaurantes disponíveis.

Tabela 7 – Distribuição de cidades e total de restaurantes

id	name	restaurants	id	name	restaurants
3943	10	São Paulo	98	21	Balneário Camboriú
1400	16	Rio de Janeiro	89	120	Niterói
472	7	Belo Horizonte	85	97	Florianópolis
372	8	Curitiba	82	86	Goiânia
272	2	Brasília	34	6	Ribeirão Preto
270	41	Recife	29	248	Baixada Fluminense
246	15	Porto Alegre	25	107	Londrina
200	4	Salvador	22	13	Joinville
129	12	Fortaleza	20	226	Norte Fluminense
121	1	Vitória	14	225	Criciúma
115	37	Campinas	14	32	Santos

A seguir é apresentado a sexta tabela do banco de dados e última a ser detalhada no projeto. Ela é representada pela figura 15, contendo as cozinhas dos restaurantes, pelo qual também tem importante papel no processo de recomendação. Ao todo é utilizado 64 opções

de cozinhas. Repare que 2 opções nesta lista não são exatamente cozinhas, entretanto, acabam por ser utilizadas por praças de alimentação, sendo elas: Comida Rápida e Almoço Express.

Figura 15 – Distribuição de dados da tabela cuisine

id	name	id	name	id	name
1	Peruana	24	Tailandesa	49	Coreana
2	Alemã	25	Francesa	50	Havaiana
3	Árabe	26	Trattoria	51	Vegana
4	Brasileira	27	Internacional	52	Comida Rápida
5	Judaica	28	Portuguesa	53	Colombiana
6	Ibérica	29	Frutos do Mar	54	Vietnamita
7	Mineira	30	Mediterrânea	55	Pastelaria
8	Asiática	31	Cafeteria	56	Lanches
9	Contemporânea	32	Argentina	57	Padaria
10	Russa	33	Carnes	58	Confeitaria
11	Americana	34	Belga	59	Bósnia
12	Marroquina	35	Italiana	60	Grega
13	Chinesa	36	Japonesa	61	Creperia
14	Nordestina	37	Irlandesa	62	Sorveteria
15	Pizzaria	38	Espanhola	63	Almoço Express
16	Austriaca	39	Inglesa	64	Australiana
17	Saudável	40	Galeteria	65	Comida de Boteco
18	Mexicana	41	Tex-Mex	66	Fusion
19	Variada	45	Vegetariana	67	Molecular
20	Fondue	46	Andina	70	Bar
21	Hamburgueria	47	Shake	-	-
22	Indiana	48	Torta	-	-

3.3 Normalização

A normalização é importante pois ela garante que não exista valores ausentes e *outliers*, ou seja, um valor atípico, que apresenta um grande afastamento dos demais da base. Além disso, este processo ajuda na criação de escalas numéricas (AVANÇO, 2015).

Para uma boa normalização, é recomendado primeiramente que se faça uma análise de instâncias com valores nulos na base de dados a ser utilizada. Se existir poucas ocorrências, então basta remover as instâncias nulas, entretanto, caso exista diversas ocorrências, é melhor remover a coluna inteira. Por exemplo, digamos que exista uma base de dados com 3 colunas e 10 linhas, essas colunas são referentes a nome, CPF e *email*. Suponha que o sistema utilizado para obter os dados não obrigava os clientes a digitar o CPF, neste caso, é muito provável que exista mais de uma ocorrência nula para esta informação, o que prejudicaria a análise. Neste caso, o melhor a se fazer é remover esta coluna.

Finalizando a etapa da remoção dos valores nulos, é dado início a normalização de escala. Para este projeto, será utilizado a escala entre 0 e 1, isto é, dado um conjunto de dados, a maior instância passa a ser 1 e os demais elementos sofrem alteração com base neste, sendo 0 o limite mínimo do conjunto. A figura 16 apresenta uma tabela com exemplos de formulas para calcular essa normalização.

Figura 16 – Tabela de normalização por transformação linear

Fórmulas de transformação	Valores normalizados dos extremos da escala	Observações
$(z_i^k)_N = \frac{z_i^k - z_{min}^k}{z_{max}^k - z_{min}^k}$	Máximo = 1 Mínimo = 0	Atributos de maximização Ideal = 1
$(z_i^k)_N = \frac{z_{max}^k - z_i^k}{z_{max}^k - z_{min}^k}$	Máximo = 0 Mínimo = 1	Atributos de minimização Ideal = 1
$(z_i^k)_N = \frac{z_i^k}{z_{max}^k}$	Máximo = 1 Mínimo = z_{min}^k / z_{max}^k	Atributos de maximização Ideal = 1
$(z_i^k)_N = \frac{z_{max}^k}{z_i^k}$	Máximo = z_{max}^k / z_{min}^k Mínimo = 1	Atributos de minimização Transformação não-linear

Fonte: (AVANçO, 2015)

Elas referem-se a um atributo k , onde a alternativa i tem o valor z_i^k . Note que a transformação linear do atributo não ocorre na última função, por este motivo, não é completamente correto usar o termo normalização para este caso. Essa normalização em escala é feita quase que instantaneamente a aplicação do método *K-means*, ou seja, os dados salvos na base de dados estão apenas categorizados e somente na etapa de análise que é aplicado essa escala entre 0 e 1, mantendo assim a originalidade dos dados salvos.

A tabela 8 apresenta a distribuição dos dados de categorias após a aplicação do método de normalização:

Tabela 8 – Distribuição de categorias normalizadas

normalized	name	id
0.083333	À la carte	1
0.250000	Rodízio	3
0.333333	Comida rápida	4
0.416667	Food Truck	5
0.500000	Executivo	6
0.583333	Cafeteria	7
0.666667	Buffet	8
0.833333	Bar	10
0.916667	Saudável	11
1.000000	Sobremesa	12

3.4 Analisador

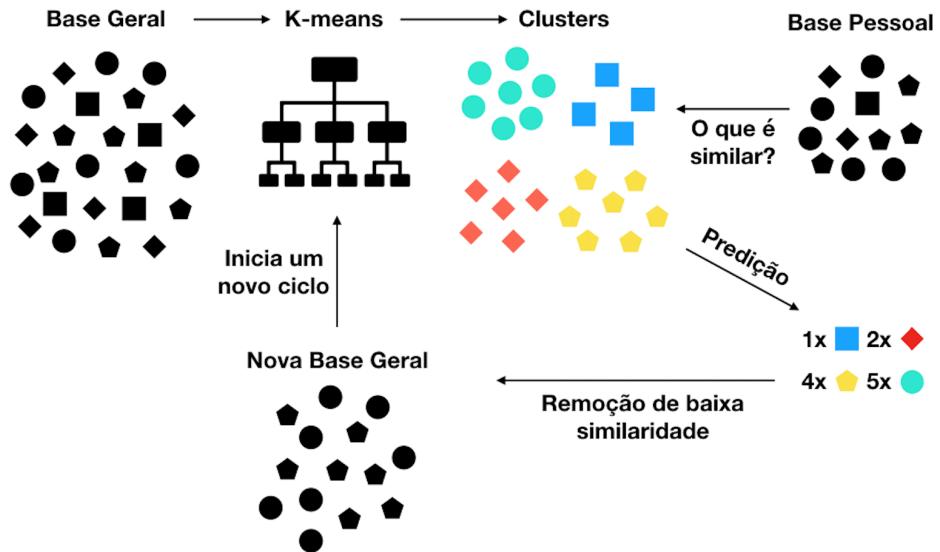
O analisador é o cérebro do projeto e por sua vez é responsável em obter informações relevantes para o estudo do caso, recomendação de restaurantes. Por se tratar de dados não supervisionados, isto é, quando não se tem uma saída esperada após a análise das informações, foi necessário escolher um método que pudesse ser eficaz para este tipo de modelo. Já citado em capítulos anteriores, o método utilizado para realizar as análises necessárias do projeto é o *K-means*. O emprego desse método pode ser visto em diferentes áreas do conhecimento, como já citado. Ele faz uso da clusterização de dados com o objetivo de dividi-los em grupos de dados semelhantes.

Para uma melhor compreensão do que está sendo apresentado, a figura 17 a seguir exemplifica como funciona essa divisão dos dados por semelhantes. No exemplo, os dados são objetos e estão diferenciados por uma única informação, a forma destes objetos.

Voltando para o cenário deste trabalho, ao invés de objetos, o autor trabalha com dados que representam restaurantes no meio digital. Para tal representação, é utilizado ao todo 16 características, elevando significativamente a complexidade do problema quando comparado a figura 17, onde só existe a forma do objeto para diferenciar os dados.

Os atributos utilizados no trabalho são: cidade, bairro, latitude, longitude, preço, avaliação, categoria, cozinha, tipo, momento, desconto, benefício, restrição, quantidades de pessoas na mesa, se é válido para feriados e se possui *wifi* gratuito.

Figura 17 – Abstração do funcionamento do algoritmo desenvolvido

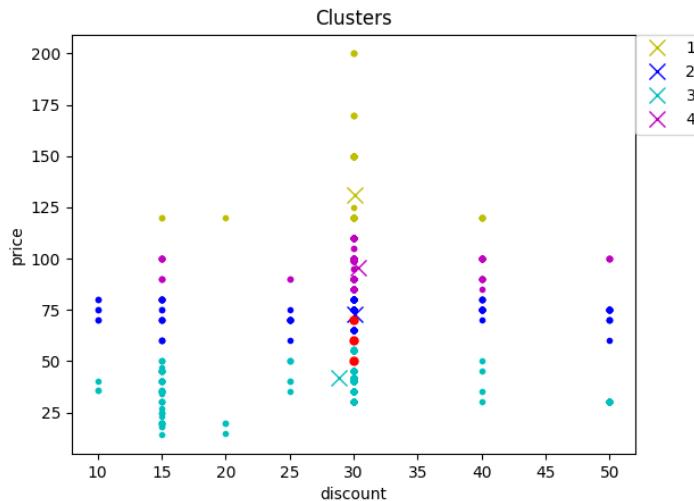


Fonte: Produção do próprio autor

Dado a complexidade do problema analisado, o autor constantemente comparou em pares, os diferentes atributos do sistema por meio de uma visualização gráfica. A análise em pares permitiu distribuir os dados em um gráfico 2D de eixos X e Y. Essa visão facilitou

o entendimento do algoritmo durante o processo de desenvolvimento, além de trazer uma visualização dos dados *clusterizados*, como mostra o exemplo dado pela figura a seguir.

Figura 18 – *Clusters* separador por *price* e *discount*



Fonte: Produção do próprio autor

No exemplo da figura 18, foi utilizado uma pequena amostra de dados para dividir os restaurantes por meio de preço e desconto em 4 grupos distintos. O X representa as centróides de cada *cluster*, já citado na seção 2.5 onde o *K-means* é introduzido. É possível perceber que além das 4 cores utilizadas para representar cada *cluster*, também é apresentado uma quinta cor, menos presente no gráfico. Esta cor representa os restaurantes de base pessoal utilizados no processo de predição, como é introduzido na figura 17 no problema com os objetos de formas diferentes.

Outro ponto de dificuldade observado durante o desenvolvimento do trabalho se dá pelo fato do modelo ser não supervisionado. Por não haver uma saída esperada que possa informar se a análise deu certo ou errada, o grau de confiabilidade das informações sugeridas aos usuários acabam sendo comprometidas. Para minimizar tal problema do modelo, o autor utiliza de técnicas de aprendizado por reforço, apresentado inicialmente por Pavlov (2013) em seu experimento conhecido por *Pavlov's Dogs*.

Tal aprendizado também é conhecido como modelo semi supervisionado e é dado pela ação externa do usuário. É uma forma inteligente de melhorar a confiabilidade das recomendações passadas, além de melhorar as métricas qualitativas do método *K-means*.

Outra maneira de mitigar problemas com recomendação mal classificadas, isto é, quando é recomendado um dado com baixa similaridade em relação as preferências do usuário, é fazer o refinamento dos dados. Para melhor compreensão, a figura 17 apresenta esse passo durante o processo de remoção de dados de baixa similaridade. Repare na figura que se trata de um processo cíclico, sendo assim a etapa responsável em refinar repetidamente os dados até que seja obtido o resultado esperado.

Para o sistema desenvolvido, o momento de parada do ciclo, isto é, o resultado esperado, se dá quando o algoritmo não conseguir mais separar os dados em diferentes *clusters*, ou seja, a similaridade entre eles é grande o suficiente, perdendo assim o sentido em continuar o ciclo. Outra verificação que ocorre em paralelo é referente ao número total de elementos na nova base geral, ou seja, após ser refinada. Se essa base for menor ou igual a 50 itens, o ciclo também é encerrado e os que restaram são recomendados. É importante ressaltar que tais decisões foram baseadas e adaptadas de aplicações semelhantes a desenvolvida neste trabalho, como é explicado nos parágrafos seguintes.

A análise cíclica desenvolvida utiliza como entrada do método de avaliação a base geral de restaurantes, sendo levado em consideração somente dados da cidade escolhida pelo cliente, e as preferências de usuário, sendo elas reservas realizadas, listagem de favoritos e avaliação de recomendações antigas.

Para a escolher as atributos a serem analisados neste trabalho, assim como a utilização de recomendação refinada, o autor partiu da linha de pensamento dos artigos publicados por Sawant (2013), que propõe uma recomendação de restaurantes utilizando a base de dados da *Yelp*, Jayasimhan; Rai; Parekh; Patwardhan (2017), que por meio dos dados obtidos da plataforma *Zomato* e novamente da *Yelp*, aplica técnicas e algoritmos de aprendizagem de máquina para recomendar restaurantes, e por fim, Manoli (2015), que utilizam dados de localização do restaurante, cozinha, ambiente e avaliações como quesitos para preparar a recomendação.

Apresentado uma visão completa de tudo o que foi dito até o momento, a figura 19, contextualiza o passo a passo do algoritmo de recomendação desenvolvido neste trabalho.

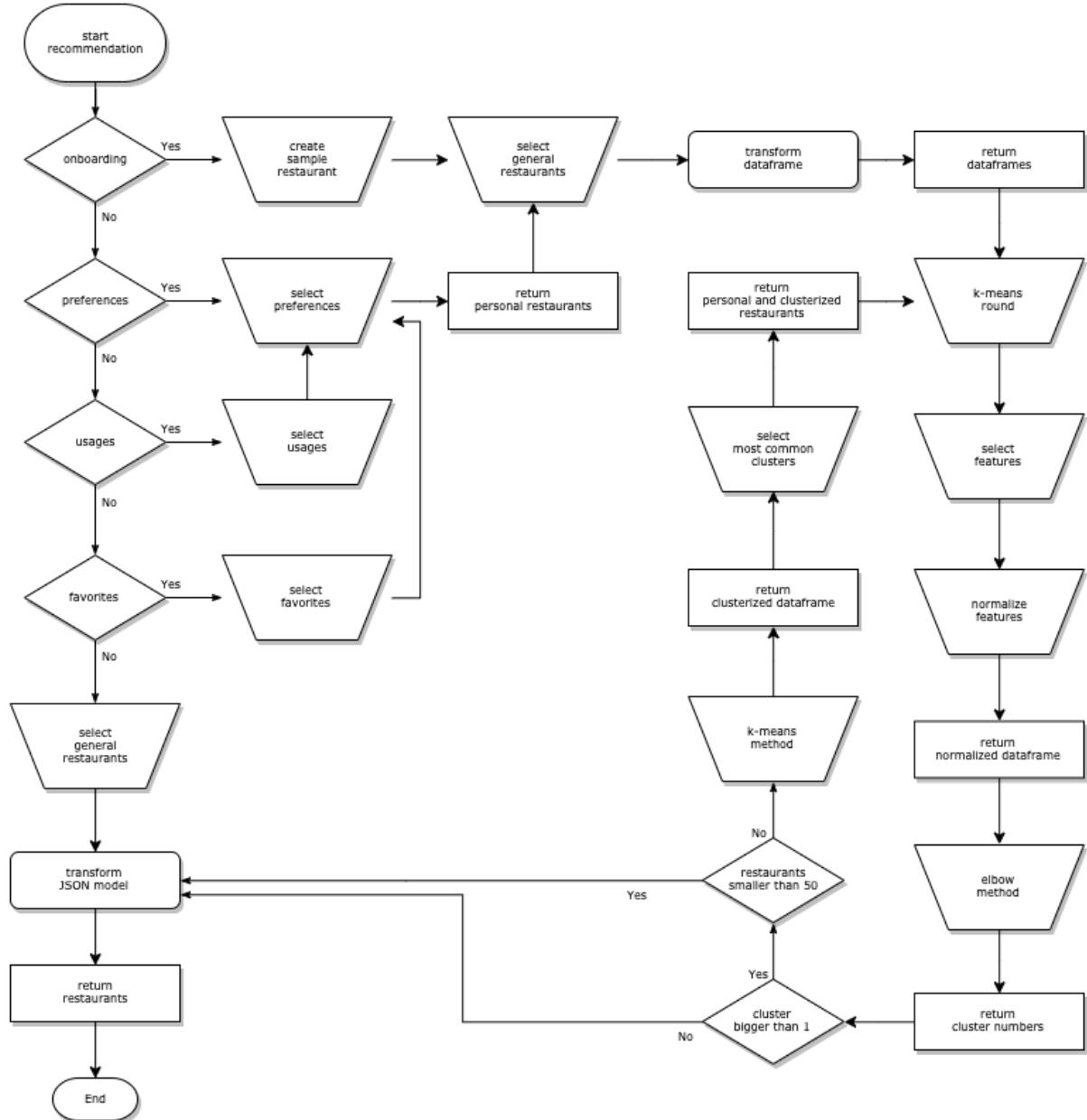
Como citado anteriormente, o sistema emprega o uso do modelo não supervisionado e por meio de avaliação externa com usuário, passa para o ambiente de aprendizado por reforço. Esse fluxo ocorre da seguinte forma: dado uma recomendação, o usuário pode informar se gostou ou não da mesma, entretanto, é preciso que haja recomendações para que seja possível avaliá-las.

Pensando no que foi apresentado, um novo usuário que acabara de criar uma conta, ainda não possui consigo, dados suficientes para que seja feita uma recomendação inicial eficiente, isto é, que seja próxima a suas preferências. Sendo assim, uma forma de minimizar o problema apresentado, é a utilização de *onboarding*, apresentado em detalhes na subseção 3.6.1. Por meio desta solução, sempre que um novo usuário entrar na base, é solicitado a este que preencha alguma de suas preferências, permitindo assim uma primeira recomendação mas eficiente, ainda que em ambiente não supervisionado.

A etapa de *onboarding* pode ser visualizada na figura 19, sendo ela a primeira lógica a ser verificada antes de dar início as recomendações. O usuário só consegue avançar para a listagem de restaurantes e visualizar os dados, após preencher os requisitos solicitados,

como é apresentado no seção 3.6.

Figura 19 – Fluxograma do algoritmo de recomendação



Fonte: Produção do próprio autor

Ainda analisando o fluxograma da figura 19, é possível observar que existem três tipos diferentes de recomendação, sendo elas: Preferências, Usos e Favoritos.

- Recomendação por preferências: é a principal recomendação do sistema desenvolvido. Ela utiliza os dados obtidos no *onboarding* e as avaliações de recomendações anteriores como base para novas recomendações. Citado na seção 3.6, sempre que uma recomendação é feita, o usuário tem a possibilidade de informar se gostou do que foi sugerido, ou simplesmente remover a recomendação de suas listas, como mostra a figura 20. Sempre que um restaurante recomendado é removida da lista, o algoritmo

passa a não considerar mais o dado como relevante nas recomendações para esse usuário. Caso a avaliação seja positiva, como mostra a figura 21, o sistema passa a ter como base as características dessas preferências para recomendar novos itens.

- Recomendação por uso: é obtida por meio da análise de restaurantes pertencentes a base de histórico de reservas (*booking*) e *check-in*. Estes usos podem ser realizados a partir do aplicativo desenvolvido, citado na seção 3.6, e tem o objetivo de simular reservas em restaurantes.
- Recomendação por favoritos: similar a recomendação por uso, a base de dados utilizada aqui é a de restaurantes favoritos. Caso a lista do usuário esteja vazia, essa recomendação não é oferecida no aplicativo.

Após obter a base de dados pessoal, seja ela criada a partir de preferências, usos ou favoritos, os dados são passados juntos a base geral de restaurantes para o método *K-means Round*, responsável por controlar cada ciclo.

Dado início ao primeiro ciclo, os dados recebidos são pré-processados, isto é, uma nova base interna ao método *K-means Round* é criada somente com as atributos a serem *clusterizadas*. Além disso, os dados também passam pelo processo de normalização entre 0 e 1, já citado na seção 3.3.

Para obter o número ideal de *clusters* pelo qual os dados devem ser divididos, é utilizado o método *Elbow*. Ele é responsável por avaliar a variação entre os dados e informar a melhor distribuição entre eles.

Conforme apresenta a figura 19, uma vez normalizado e total de *clusters* obtidos, é feito a verificação citada anteriormente para saber quando o ciclo deve ser encerrado. Na sequência o método *K-means* finalmente é chamado e os dados passam a ser *clusterizados*. Ao final do processo, é dado início a etapa de predição, isto é, dado a base pessoal do usuário e as bases separadas pelo *K-means*, é visto quais delas mais se assemelham as preferências do usuário. Os grupos que não tiverem semelhança com o ou possuírem um número de semelhantes abaixo da média, são descartados e é dado início a um novo ciclo com os dados que restaram.

Em média foram necessários 4 rodadas até conseguir uma recomendação refinada. Além disso, o autor também percebeu que o número total de *clusters* não passam de 6, sendo em média 4, o total de divisões mais comuns observadas.

3.5 Resultados

A etapa de resultados é onde os dados recomendados são organizados em modelos de objetos. Este processo é realizado por meio do paradigma orientado a objetos⁴, que facilita o desenvolvimento da aplicação para visualizar os dados, uma vez que o paradigma tem o objetivo de representar estruturas lógicas o mais próximo do mundo real.

Ao todo foram criados 4 modelos de dados, sendo eles: *user*, responsável em montar a estrutura da tabela *client* apresentada na seção 3.2; *filter*, utilizado para mapear as tabelas mais simples, como a de *cuisines* por exemplo, onde o dado é uma estrutura de *id* e *name*; *offer*, usado para representar uma oferta de restaurante; *restaurant*, modelo robusto que depende dos modelos de *filter* e *offer* para mapear os dados de restaurante.

No início do projeto, este processo de resultados também foi utilizado para gerar gráficos 2d por meio do *Matplot*, com o objetivo de analisar os resultados obtidos a partir das recomendações geradas. Essas análises são apresentadas no capítulo 4 com mais detalhes.

3.6 Aplicativo

Embora fosse possível apresentar o sistema de recomendação diretamente do console da aplicação *Python*, e ao mesmo tempo mostrar a sua eficácia por estatísticas e dados matemáticos, o autor do presente trabalho sentiu a necessidade de apresentar as recomendações de uma forma mais perceptível e amigável ao usuário e por isso, foi desenvolvida uma aplicação mobile para este fim.

Como visto no capítulo 3 seção 3.7, a aplicação mobile utilizada a linguagem *Flutter*, que permite construir um aplicativo multi plataformas, isto é, tanto para celulares que utilizam *Android* como *iOS*, utilizando o mesmo código base para ambos.

Um dos principais pontos para a decisão em fazer uma aplicação mobile veio ao trazer as listas inteligentes para telas e a importância do gerenciamento de dados das mesmas. Em celulares, as listas são renderizadas conforme o usuário faz *scroll* em tela e elimina automaticamente os dados não visíveis e sobressalentes. O mesmo não ocorre na web, o que faria com que o suposto site ficasse lento, uma vez que este projeto utiliza milhares de restaurantes em sua base.

Outro importante aspecto está no *Design Thinking*⁵. De forma simplificada, o intuito do *Design Thinking* é apresentar o material de uma forma clara, coesa e que apresente facilmente as informações ao usuário. Por isso, no desenvolvimento desta aplicação foram

⁴ <https://www.inf.pucrs.br/~gustavo/disciplinas/pli/material/paradigmas-aula12.pdf>

⁵ <https://www.enterpriseirregulars.com/125085/what-is-design-thinking/>

seguidas a Material Guideline⁶ e Cupertino Guideline⁷, representadas pela *Google* e *Apple*, respectivamente; as duas pelas quais prometem boas apresentação de dados em celulares.

O armazenamento de cache também foi levado em consideração. Para evitar com que a aplicação consulte frequentemente a base de dados, é importante que as informações fiquem armazenadas na mesma. Como a exclusão destes dados é mais difícil de ser realizada em ambientes mobile do que na web, a performance do algoritmo em celular tende a ser eficaz.

3.6.1 Onboarding

Onboarding em tecnologia, é o processo dado pela etapa de explicação de um novo produto para os usuários que estão a conhecê-lo pela primeira vez⁸. A técnica de *onboarding* já é utilizada por muitos sistemas de recomendações, como por exemplo os aplicativos *Apple Music*, *Spotify*, *Deezer* e *Netflix*. Para que o usuário tenha uma melhor primeira experiência e o algoritmo consiga definir com mais exatidão as recomendações, é fundamental obter algumas preferências previamente. No presente trabalho, as informações solicitadas no *onboarding* são:

- Cidade: no momento do cadastro o usuário precisa indicar uma cidade. Aqui são apresentadas as cidades onde possuem restaurantes cadastrados na base de dados
- Culinária: aqui são apresentadas as culinárias dos restaurantes cadastrados na base de dados. O usuário seleciona qual é a sua preferência absoluta.
- Avaliação: aqui é mostrada a avaliação do restaurante em estrelas. O usuário seleciona qual quantidade é a sua preferência.
- Valor Médio Gasto: aqui é apresentada a quantidade em reais que é gasta em média, por pessoa, em restaurantes. O usuário precisa escolher uma quantia que melhor se adapta às suas necessidades.
- Pessoas na Mesa: aqui o usuário precisa definir a quantidade de pessoas que costuma ter como companhia em suas refeições.
- Ocasião: aqui são apresentadas as principais ocasiões que podem fazer o usuário sair para comer fora. O usuário precisa escolher uma entre as seguintes opções: Para ir com a família, Para ir com Amigos, Almoço de trabalho ou Romântico.

Depois de selecionadas as informações acima, o usuário precisa validá-las pressionando o botão “Ver restaurantes”. Assim que as informações forem recebidas, o algoritmo

⁶ <https://material.io/design/>

⁷ <https://developer.apple.com/design/human-interface-guidelines/>

⁸ <http://www.elasticode.com/what-is-onboarding.php>

inicia o que seria a segunda parte do *onboarding*, e a primeira recomendação do aplicativo, neste momento não supervisionada, sugerindo alguns restaurantes para o usuário autenticar o seu interesse.

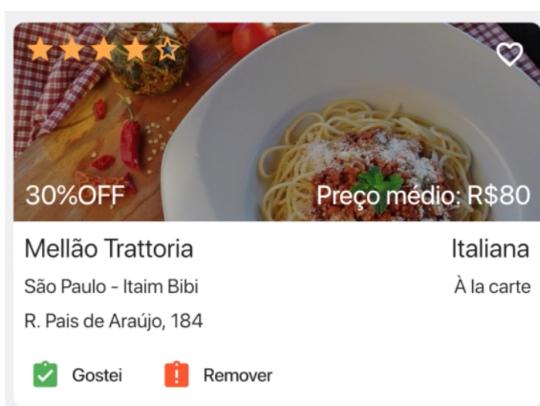
3.6.2 Primeira recomendação não supervisionado

Nesta primeira tela de recomendação do aplicativo, o algoritmo utiliza o *K-means* para gerar grupos de diferentes perfis, comparando este usuário a grupos semelhantes e eliminando os divergentes. O mesmo é feito até que existam no máximo 50 opções de recomendação, garantindo assim que não seja recomendado milhares de opções, perdendo o sentido da recomendação, ou então quando existir apenas um grupo de semelhantes.

O número de grupos é obtido pelo método *Elbow* pelo qual compara a variação entre os dados e à partir dela, é recomendado dividir os dados em N grupos. Quando esse N é igual a 1, significa que os dados são muito semelhantes, não havendo mais a necessidade de dividi-los, resultando em uma recomendação não supervisionada precisa.

O intuito nesse momento é que o usuário escolha pelo menos uma das opções mostradas e valide no botão “Gostei”. Ao selecionar todas as opções que lhe são interessantes e pressionar o botão “Continuar”, o usuário é direcionado para outra tela já com as recomendações que melhor se adaptam ao seu perfil e preferências, finalizando esse processo inicial de *onboarding* e apresentando as recomendações semi supervisionadas.

Figura 20 – Exemplo de pedido de avaliação de recomendação



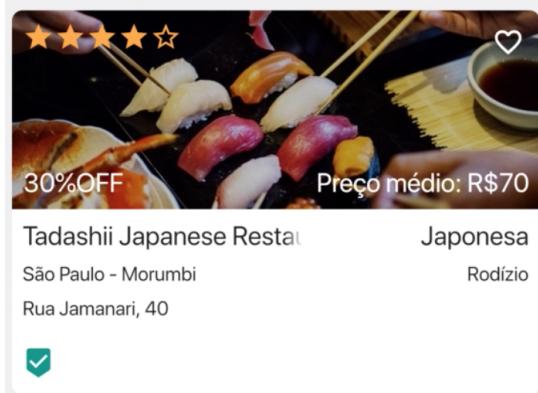
Fonte: Produção do próprio autor

3.6.3 Primeira recomendação semi supervisionado

As recomendações semi supervisionadas são feitas após uma validação do usuário ao receber uma sugestão não supervisionada. Isso é, depois de receber uma recomendação e aprová-la ou rejeitá-la, o algoritmo *K-means* faz uma nova análise baseada nessa nova informação.

Caso o usuário assinale que gosta de um restaurante, mais opções de lugares semelhantes serão expostos em seguida. Entretanto, caso o usuário selecione a remoção de uma recomendação, o algoritmo nunca mais trará esse restaurante para as suas recomendações, tampouco recomendará restaurantes parecidos.

Figura 21 – Exemplo de recomendação avaliada positivamente



Fonte: Produção do próprio autor

3.6.4 Recomendações baseadas em reservas e favoritos

As recomendações baseadas em reservas e restaurantes favoritos, são feitas por meio do histórico das mesmas. Isso é, sempre que o usuário realizar uma nova reserva ou favoritar um novo restaurante, os novos dados são passados para o algoritmo *K-means* que inicia uma nova leitura de recomendações.

Desta forma, o algoritmo consegue traçar um perfil com base nos restaurantes selecionados, possibilitando mais uma recomendação não supervisionada. O mesmo ocorre a cada nova informação recebida, comparando os grupos na base de dados e separando-os, até com que o número de grupos tenda a um.

3.7 Tecnologias utilizadas

Para o desenvolvimento do *web-service* e todo o sistema de recomendação deste projeto, foram utilizados métodos e *frameworks* da linguagem de programação *Python*, em sua versão 3.6. As principais bibliotecas que tornaram a solução possível foram: *scikit-learn*, *matplotlib*, *pandas*, *numpy*, *flask* e *sqlalchemy*.

- *Scikit-learn*: é uma *framework* escrita em *Python* com diversos métodos de aprendizado de máquina. Para o trabalho, foi utilizado o método *K-means* presente nesta biblioteca.

- *Matplotlib*: permite criar gráficos 2D de boa qualidade e diferentes formatos. O *Matplotlib* permitiu visualizar os dados e *clusters* durante o processo de ajuste do algoritmo *K-means*, sendo assim essencial para o trabalho.
- *Pandas*: oferece técnicas de manipulação e análise de dados, podendo assim trabalhar com estruturas que facilitam o processo de estudo e compreensão das informações analisadas.
- *Numpy*: possui diversas funções matemáticas de alta nível que permitem operar de forma facilitada com matrizes e dados multi-dimensionais.
- *Flask*: é uma *framework* com métodos que permitem desenvolver textitweb-services. Sua simplicidade facilita a compreensão dos desenvolvedores.
- *Sqlalchemy*: permite trabalhar com a linguagem *SQL* para gerenciar banco de dados. As leituras e gravações feitas no *database* deste projeto foram por meio desta biblioteca.

Para facilitar o desenvolvimento do projeto, foi utilizado a IDE *PyCharm* para *Python*. O uso da IDE permitiu focar no que realmente importava no trabalho, o sistema de recomendação. Problemas de sintaxe com a linguagem, importação e instalação de *frameworks*, eram resolvidos pela própria ferramenta, sendo assim muito útil.

Para o desenvolvimento da visualização de dados e produção de nova base, foi implementado um aplicativo, como já citado anteriormente, utilizado a mais recente linguagem da *Google*, o *Flutter*. Ele permite o desenvolvimento da UI criado por *Google* e *Apple*. É usado para desenvolver aplicações tanto para *Android*, *iOS*, *Windows*, *Mac*, *Linux* e *web*.

Diferente de outras linguagens e *frameworks* similares ao *Flutter*, como o *Cordova*, *Xamarin*, etc, o código gerado é nativo e compilado utilizando as ferramentas de cada plataforma, mitigando possíveis lentidões e problemas de compatibilidade, sendo muito parecido com o que é proposto pelo *React Native*, *framework* do *Facebook* com o propósito de criar aplicações em multi-plataformas.

A *API* desenvolvida em *Python* segue no padrão *RESTful* e a arquitetura da aplicação mobile tem como base a arquitetura *MVVM* que tem o objetivo de separar lógica e regra de negócio da parte visual, isto é, da *view*. Para o armazenamento do dados, foi utilizado a estrutura de banco de dados *SQLite*.

4 Resultados e Discussões

Com o objetivo de facilitar o acompanhamento de testes realizados durante o desenvolvimento do projeto, foram gravados 2 vídeos sobre o funcionamento do sistema. O primeiro¹ foi realizado pelo próprio autor, utilizando como base suas próprias preferências. O segundo² vídeo foi gravado com um usuário externo pelo qual o autor guiou o teste sem impor qualquer viés, como é explicado nas seções seguintes. É importante ressaltar que todos os usuários que foram submetidos aos testes concordaram em compartilhar as informações aqui apresentadas.

Durante os testes, o aprendizado por reforço mostrou-se capaz de melhorar a recomendação de restaurantes apresentados. A seguir é apresentado um caso onde essa melhora pode ser observada: No momento em que a recomendação não supervisionada foi sugerida, a mesma continha alguns exemplos de culinária italiana, conforme o usuário rejeitava escolhas de tal gastronomia, o algoritmo passava a mostrar com menor frequência tal cozinha. O inverso ocorreu quando o usuário fez uma reserva em um restaurante francês e favoritou restaurantes de frutos do mar, fazendo com que a recomendação semi supervisionada passe a apresentar dados mais relevantes e com opções mais semelhantes a tais escolhas. O caso apresentado pode ser visto nos vídeos que foram gravados durante os testes.

Outro ponto interessante é a melhora do algoritmo diante ao processo de *onboarding*. Com isso, é possível conhecer um usuário novo que ainda não possui reservas ou restaurantes favoritos e facilita que ele entre rapidamente em um ambiente semi supervisionado. Nas versões anteriores do trabalho, isto é, durante o processo de desenvolvimento onde tal etapa ainda não existia, era inviáveis recomendar restaurantes para novos usuários sem considerar premissas como melhores avaliados ou os mais reservados.

Um dos principais desafios deste trabalho está relacionada à normalização de dados. Hoje, o algoritmo não leva em consideração a distância entre os elementos do mundo real, como é o caso de cidades e bairros. Uma normalização mais eficaz seria capaz de compreender essa distância e enumerá-las pelo seu afastamento físico. No entanto, o problema foi mitigado no momento que as recomendações passaram a mostrar restaurantes que estavam localizando somente na cidade que o usuário elegeu no momento do cadastro.

¹ <https://drive.google.com/open?id=1f9bPXx8ldaWrOWNMD0vBmvapRpdhtA4X>

² https://drive.google.com/open?id=15hdxpT7dhD3U_x3Ok_9ahWUKiOyl02pP

4.1 Método de avaliação

Com o objetivo de avaliar a eficácia do sistema de recomendação desenvolvido neste trabalho, foi necessário escolher uma metodologia de avaliação. Por se tratar de um sistema que trabalha com dados não supervisionados e que utiliza do aprendizado por reforço para evoluir a qualidade das recomendação, a escolha do método acabou por ser a pesquisa em campo com usuários.

Essa pesquisa envolveu um cenário pelo qual eles pudessem criar uma conta no aplicativo desenvolvido, efetuar o passo de *onboarding*, e a partir deste ponto, ter livre acesso as listagens de restaurantes, sendo elas, gerais e recomendados. Afim de não enviesar os testes, foi informado para o usuário apenas as funcionalidades de reserva, favoritos e avaliação de recomendação. A partir dessa explicação inicial, os usuários foram instruídos a navegarem livremente pelo sistema, pedindo apenas que fizessem ao menos 5 avaliações das recomendações fornecidas.

4.2 Teste com usuários

Ao todo foram realizados 83 testes, sendo 32 deles presenciais e 51 via TeamViewer³, plataforma para efetuar chamadas de vídeo, que também permite dar acesso ao controlamento remoto da maquina. Os testes locais foram realizados direto pelo dispositivo mobile e os demais por meio de simulador. Em ambos os testes a cidade escolhida foi São Paulo, sendo a mais numerosa da base, como já citado em capítulos anteriores.

Os testes foram realizados em novembro de 2019 com usuários de diferentes classes sociais, sendo elas, B, C e D⁴. Além disso, também contou com diferentes faixas etárias de idade, sendo essas, grupos de pessoas entre 20 e 26 anos, 35 e 40 anos e entre 55 e 60 anos. Excluindo a última faixa etária de idade, todos os demais usuários já haviam realizado ao menos uma reserva online de restaurante por meio de aplicativo *mobile*.

Após os 83 usuários serem concluídos, foram recolhidas questões interessantes em relação ao aplicativo e ao algoritmo de recomendação de restaurantes. Esses tópicos foram divididos em pontos principais, explicados a seguir.

Uma das questões abordadas pelos usuários de teste foi em relação a demora das recomendações aparecerem na tela do aplicativo. Depois de uma breve análise, foi constatado que o atraso seria decorrente dos vários elementos verificados em tempo real e da forma que o projeto foi implementado, as recomendações são elaboradas ao abrir a tela de restaurantes. Para resolver esse problema, seria interessante realizar uma recomendação em background, por exemplo.

³ <https://www.teamviewer.com/en/>

⁴ <https://cps.fgv.br/qual-faixa-de-renda-familiar-das-classes>

Outra melhoria apresentada foi a semelhança das listas de recomendação depois de algum tempo de uso. Segundo alguns usuários testes, as listas ficam iguais ou muito semelhantes e isso acabou por confundi-los. Conforme os dados constataram tal afirmação, talvez não faça sentido usar três listas diferentes. Embora elas auxiliam no começo quando existem poucas reservas e restaurantes favoritos, depois de um tempo elas se tornam dispensáveis.

Segundo alguns usuários, a presença de recomendações com culinária italiana foi frequente mesmo quando o mesmo não selecionou tal cozinha. Como foi explicado acima, o algoritmo não se baseia exclusivamente nas escolhas específicas realizadas no *onboarding* e ao analisar os testes em questão, foi observado um padrão de comportamento: valores mais altos e ambientes românticos. Tais fatores foram frequentemente encontrados em restaurantes italianos da base aqui utilizada. No entanto, ao solicitarem a remoção de restaurantes italianos da lista, as recomendações de tal cozinha foram cada vez menos frequentes.

Em alguns casos, os usuários não gostaram do que foi recomendado no momento do *onboarding* e mesmo assim foram obrigados pela aplicação a escolher um opção. Uma solução futura para o problema seria considerar as opções como “não gostei” e implementar um botão para mostrar mais recomendações.

Outro ponto analisado veio de usuários que não conheciam os restaurantes. Segundo eles, uma galeria de fotos dos locais e dos pratos ajudaria mais na escolha do que as fotos estáticas de acordo com a culinária do restaurante que são utilizadas hoje.

O *onboarding* foi tema de melhorias principalmente por usuários cuja a prioridade principal é o desconto. De acordo com eles, carecia de informações dos benefícios e por isso, ficava difícil escolher um restaurante dentre os recomendados. No entanto, não houve indagações do porquê o *onboarding* era necessário, talvez por já ser muito utilizado em outras aplicações mais conhecidas.

De forma geral, os usuários pareceram satisfeitos com as recomendações. Em 70% dos casos o *feedback* foi positivo. Dentre os pontos favoritos estão o respeito pelo valor pré estabelecido, variedade de bairros, qualidade dos restaurantes e semelhança de opções com que as quais o usuário já costuma ir.

Ao todo foram recomendados 2483 restaurantes, como mostra a tabela 9, sendo em média um total de 30 resultados por usuário. Esse número varia de acordo com as preferências escolhidas durante o processo de *onboarding* e o número de rejeições de recomendações realizadas. É importante lembrar que uma das decisões tomadas durante o desenvolvimento foi exibir no máximo 50 recomendações, o que também causou impacto nestes resultados exibidos.

Tabela 9 – Distribuição de recomendações por usuários

clientId	results	clientId	results	clientId	results	clientId	results
1	11	22	35	43	18	64	14
2	22	23	25	44	50	65	12
3	44	24	27	45	48	66	13
4	48	25	19	46	31	67	10
5	31	26	20	47	26	68	22
6	26	27	34	48	37	69	32
7	24	28	40	49	25	70	44
8	49	29	32	50	35	71	33
9	37	30	30	51	23	72	47
10	45	31	17	52	16	73	29
11	29	32	43	53	11	74	49
12	50	33	10	54	30	75	41
13	46	34	12	55	28	76	27
14	47	35	33	56	38	77	20
15	38	36	36	57	39	78	24
16	13	37	39	58	21	79	34
17	28	38	14	59	17	80	19
18	16	39	21	60	42	81	40
19	42	40	18	61	45	82	46
20	23	41	41	62	15	83	23
21	15	42	43	63	36	-	-
Total	684	Total	589	Total	631	Total	579

Uma possível melhoria no algoritmo desenvolvido seria testar outras formas de parada do *K-means Round* implementado. Isto por que na atual abordagem, foi percebida durante os testes que na rodada N-1 onde N é o total de rodadas, caso as recomendações sejam muito próximas de 50, entretanto, maior que 50, outra rodada era iniciada e com isso, os resultados acabavam sendo em torno de 15 restaurantes.

Como dito anteriormente neste capítulo, para conseguir avaliar o sistema de recomendação com um quantidade de dados relevantes, foi solicitado aos usuários a avaliação de ao menos 5 recomendações diferentes. Esse direcionamento foi positivo pois após os testes, foram obtidos ao todo, 408 resultados, como mostram as tabelas 10 e 11 a seguir.

Observe que a tabela 10 representa o maior número de avaliações, sendo essas, críticas positivas. Ao todo houve 284 respostas em relação aos restaurantes recomendados, o que representa 70% da base de avaliações. Além disso, se comparar a tabela 10 com a 11, existem muito mais usuários que deram 5 avaliações positivas do que aqueles que deram negativas, chegando as negativas em no máximo 4 vezes e tendo apenas 3 ocorrências. Já com as pessoas que avaliaram 5 vezes de forma positiva, tivemos 26 casos.

Tabela 10 – Distribuição de avaliações positivas (*like*) por usuário

id	like										
1	3	15	5	29	5	43	5	58	3	73	3
2	4	16	5	30	5	44	2	59	2	74	3
3	2	17	5	31	5	46	2	60	2	75	4
4	5	18	5	32	5	47	3	61	4	76	4
5	4	19	5	33	2	48	2	62	4	77	3
6	5	20	5	34	3	49	2	63	2	78	3
7	4	21	5	35	2	50	4	64	4	79	2
8	3	22	5	36	1	51	5	65	3	80	4
9	2	23	5	37	4	52	2	66	3	81	1
10	5	24	5	38	2	53	4	68	3	82	5
11	2	25	5	39	3	54	3	69	2	83	3
12	3	26	5	40	2	55	3	70	3	-	-
13	5	27	5	41	2	56	2	71	3	-	-
14	5	28	5	42	4	57	2	72	3	-	-
Total	52	Total	70	Total	45	Total	41	Total	41	Total	35

A seguir é apresentado a distribuição das avaliações negativas. Como pode ser observado, ao todo ocorreram 124 pedidos de remoção de restaurantes recomendados, representando 30% da base de avaliações.

Tabela 11 – Distribuição de avaliações negativas (*unlike*) por usuário

id	unlike								
1	2	36	4	49	3	62	1	74	2
2	1	37	1	50	1	63	3	75	1
3	3	38	3	52	3	64	1	76	1
5	1	39	2	53	1	65	2	77	2
7	1	40	3	54	2	66	2	78	2
8	2	41	3	55	2	67	2	79	3
9	3	42	1	56	3	68	2	80	1
11	3	44	3	57	3	69	3	81	4
12	2	45	1	58	2	70	2	83	2
33	3	46	3	59	3	71	2	-	-
34	2	47	2	60	3	72	2	-	-
35	3	48	3	61	1	73	2	-	-
Total	26	Total	29	Total	27	Total	24	Total	18

5 Conclusão

Com este trabalho, apresentou-se uma proposta para recomendação de restaurantes a partir de suas características, e comprovar que por meio de técnicas de clusterização de dados partindo do método *K-means* e aprendizagem por reforço por meio da ação externa do usuário, é possível obter resultados satisfatórios. Como foi apresentado no capítulo 4, os resultados por meio de testes com usuários apontaram que 70% da base de avaliações foram positivas. Se considerarmos o princípio de Pareto, também conhecido como regra do 80/20, é possível concluir não somente que o resultado obtido no sistema implementado foi positivo, mas que também é viável melhorar os resultados obtidos trabalhando nas principais melhorias apontadas durante os testes com usuários.

O processo de *onboarding* proposto no trabalho para conhecer as preferências de novos usuários mostrou-se muito eficiente. Mesmo que durante os testes o processo tenha sido alvo de algumas críticas negativas, vale ressaltar que foi muito bem recebido por boa parte dos usuários e elogiado por trazer dados que respeitavam as escolhas previamente feitas. Além disso, já citado anteriormente, o processo de *onboarding* pode ser visto em diferentes sistemas de recomendação, como é o caso da *Netflix*, que solicita a escolha de gêneros e filmes para conhecer novos usuários. Como tudo o que foi dito, pode-se concluir que tal implementação tem valor significativo para recomendar com qualidade.

O entendimento das diferentes técnicas existentes na disciplina de ciência de dados, assim como sua ramificação¹ para outras frentes de estudo, como por exemplo o aprendizado de máquina e a aplicação em sistemas de recomendação, foi de suma importância para concluir desenvolvimento deste trabalho.

O desenvolvimento do projeto também mostrou que encontrar padrões entre dados não é uma tarefa simples e depende de um grande volume de dados e diferentes características para conseguir obter uma análise eficaz. Vale lembrar que a base de dados utilizada neste trabalho contou com 8052 restaurantes distintos e para a análise foi utilizado 16 características, e ainda assim, o resultado obtido foi de 70% de aprovação. Com as informações que foram apresentadas, não se pode saber se com o aumento ou diminuição dos dados, esse resultado seria diferente. Entretanto, olhando somente para os apontamentos levantando durante a fase de teste, a dificuldade em avaliar uma recomendação foi maior exatamente onde a análise feita utilizava menos características, isto é, no processo de *onboarding*. Portanto, se olharmos somente para este cenário analisado, pode-se então concluir que o número total de características utilizadas no método de recomendação pode sim influenciar em um resultado melhor ou pior.

¹ <https://coggle.it/diagram/WypjBWlbpEHkyhcg/t/data-science-for-managers/c0083ef2d003b849535618df0b84e6dad73e301910c45e7eab5b72ff12f016e0>

É importante ressaltar que a compreensão dos diferentes modelos de aprendizado de máquina aqui apresentados foi essencial para o desenvolvimento do trabalho. Foi por meio deste entendimento que o autor decidiu utilizar o aprendizado semi supervisionado por meio da ação externa do usuário e agrupar os dados utilizando o método *K-means*, sendo ele uma solução possível para o problema proposto inicialmente neste trabalho.

O uso do método *Elbow* para definir o número total de *clusters* também trouxe um ganho para o sistema, isto por que ele permitiu que os dados fossem agrupados no número ideal de *clusters* e não em uma divisão estática, sendo esse valor ideal obtido por meio de cálculos da variação entre os dados.

Podemos concluir também que o método de aprendizado semi supervisionado trouxe um ganho positivo para o trabalho, uma vez que por meio deste método e de teste realizados com usuários, foi possível avaliar o sistema de recomendação desenvolvido.

O modelo de recomendação proposto pode ser implementado e aplicado em plataformas de reservas *online* de restaurantes, permitindo que os usuários dessas soluções recebam indicações de onde comer fora de casa.

5.1 Sugestões para trabalhos futuros

Brevemente citado no capítulo 4, algumas possíveis melhorias e abordagens que podem ser realizadas como trabalhos futuros são:

- *GPS*: um outro método para descobrir a localização do usuário é por meio do *GPS* do cliente em seu *mobile*. Neste projeto, foi utilizada uma pergunta questionando a localização do mesmo. Implementando o *GPS*, seria possível diminuir um passo do *onboarding*.
- Redes neurais: para a recomendação deste projeto foi utilizado o método *K-means*, que é apropriado para dados não supervisionados, como já visto anteriormente no capítulo 2. Entretanto, alguns autores utilizam a linha de pensamento voltada para Redes Neurais. Uma comparação entre os dois métodos, isto é, qual recomenda melhor, pode ser uma interessante análise futura.
- Busca do usuário: armazenar e considerar o histórico de busca dos usuários para recomendar restaurantes também pode ser um caminho a ser estudado no futuro.
- Análise textual: além das implementações citadas anteriormente, pode ser interessante a utilização de análises textuais para analisar não somente os dados discretizados, mas também informações geradas por meio de avaliações. Identificar se o comentário é positivo, negativo ou neutro e utilizar desses indicadores como elementos de recomendação pode ser uma boa abordagem.

- Filtros: a filtragem de restaurantes para melhorar a visualização dos dados é outro tema que pode ser trabalho futuramente. Desta forma, o usuário poderia procurar um restaurante que está presente na base de dados e favoritá-lo ou realizar uma reserva, possibilitando novas recomendações e melhorando o aprendizado do algoritmo resultante dessas ações.
- Cruzamento de dados: O cruzamento de dados entre usuários também pode ser vantajoso, deste modo, seria possível traçar o perfil de pessoas parecidas e recomendar baseado nessa semelhança.

Como pode ser visto, muitas são as melhorias que podem ser realizadas para sistemas de recomendação de restaurante. Além disso, como foi dito, existem diferentes abordagens para complementar a análise aqui feita, sendo o cruzamento de dados entre usuários e a análise textual, as abordagens favoritas do autor para estudar em um segundo momento.

Referências

- AMARAL, F. *Introdução a Ciência de Dados - Mineração de Dados e Big Data*. [S.l.]: Alta Books, 2016. v. 1. Citado 5 vezes nas páginas 10, 14, 16, 18 e 19.
- AVANÇO, L. V. *Sobre normalização e classificação de polaridade de textos opinativos na web*. [S.l.]: Universidade de São Paulo, 2015. Citado 2 vezes nas páginas 39 e 40.
- BAUDISCH, A. R. Ciência de dados é explorar big data para fazer perguntas para prever o futuro. *Medium*, 2016. Acesso em: 10 nov. 2018. Disponível em: <<https://medium.com/@AlfredBaudisch/o-que-%C3%A9-ci%C3%Aancia-de-dados-data-science-7af5bdac101a>>. Citado na página 17.
- BERKOVSKY, S.; CANTADOR, I.; TIKK, D. *Collaborative Recommendations: Algorithms, Practical Challenges and Applications*. [S.l.]: World Scientific, 2018. v. 1. Citado na página 27.
- CAVIQUE, L. Big data e data science. *Associação Portuguesa de Investigação Operacional*, 2014. Acesso em: 10 fev. 2019. Disponível em: <<http://hdl.handle.net/10400.2/3918>>. Citado na página 19.
- CORTES, S. da C. Mineração de dados – funcionalidades, técnicas e abordagens. *Pontifícia Universidade Católica*, 2002. Citado na página 14.
- DOMO. *Data Never Sleeps 6.0*. 2017. Acesso em: 23 nov. 2019. Disponível em: <<https://www.domo.com/learn/data-never-sleeps-6>>. Citado na página 15.
- ECKERSON, W. Smart companies in the 21st century: the secrets of creating successful business intelligent solutions. *TDWI - The Data Warehousing Institute*, 2013. Acesso em: 24 nov. 2019. Disponível em: <http://download.101com.com/tdwi/research_report/2003BIReport_v7.pdf>. Citado na página 16.
- FABRIS, F. A evolução de business intelligence para data science. 2017. Citado na página 19.
- FAGHMOUS, J. H.; KUMAR, V. A big data guide to understanding climate change: The case for theory-guided data science. *Big Data*, v. 2, n. 3, p. 155–163, 2014. Acesso em: 14 nov. 2019. Disponível em: <<https://doi.org/10.1089/big.2014.0026>>. Citado 2 vezes nas páginas 10 e 17.
- FINZER, W. The data science education dilemma. *Technology Innovations in Statistics Education*, v. 7, n. 2, 2013. Acesso em: 14 nov. 2019. Disponível em: <<https://escholarship.org/uc/item/7gv0q9dc>>. Citado 2 vezes nas páginas 10 e 17.
- HONDA, H.; FACURE, M.; YAOHAO, P. Os três tipos de aprendizado de máquina. *LAMFO - Laboratório de Aprendizado de Máquina em Finanças e Organizações*, 2017. Acesso em: 23 out. 2019. Disponível em: <<https://lamfo-unb.github.io/2017/07/27/tres-tipos-am/>>. Citado 8 vezes nas páginas 10, 11, 20, 21, 22, 23, 24 e 25.

- JAYASIMHAN, A. et al. Recommendation system for restaurants. *International Journal of Computer Applications*, Foundation of Computer Science (FCS), NY, USA, New York, USA, v. 167, n. 6, p. 23–25, Jun 2017. ISSN 0975-8887. Disponível em: <<http://www.ijcaonline.org/archives/volume167/number6/27776-2017914312>>. Citado 2 vezes nas páginas 11 e 43.
- KATARYA, R. An effective collaborative movie recommender system with cuckoo search. *Egyptian Informatics Journal*, p. 105–112, 2017. Citado na página 34.
- KRUMHOLZ, H. M. Big data and new knowledge in medicine: The thinking, training, and tools needed for a learning health system. *Health Affairs*, v. 33, n. 7, p. 1163–1170, 2014. Acesso em: 14 nov. 2019. Disponível em: <<https://doi.org/10.1377/hlthaff.2014.0053>>. Citado 2 vezes nas páginas 10 e 17.
- MANOLI, D. C. Foodr recommender system for restaurants. *University of Manchester*, 2015. Citado 4 vezes nas páginas 11, 19, 27 e 43.
- MELVILLE, P. Recommender systems. *Encyclopedia of Machine Learning*, 2010. Acesso em: 15 abr. 2019. Disponível em: <<https://www.prem-melville.com/publications/recommender-systems-eml2010.pdf>>. Citado na página 26.
- MILLER, R. *Cyclopædia of commercial and business anecdotes*. Appleton and company, 1865. Acesso em: 24 nov. 2019. Disponível em: <<https://archive.org/details/cyclopaediacomm00devegoog>>. Citado na página 16.
- MOONEY, R. J. Content-based book recommendation using learning for text categorization. *Workshop on Recommender Systems: Algorithms and Evaluation*, 1999. Acesso em: 15 abr. 2019. Disponível em: <<http://www.cs.utexas.edu/users/ml/papers/libra-sigir-wkshp-99.pdf>>. Citado na página 26.
- MOORE, A. W. *K-means and Hierarchical Clustering*. [S.l.]: Carnegie Mellon University, 2001. Citado 2 vezes nas páginas 28 e 31.
- NAUR, P. *Concise survey of computer methods*. [S.l.]: Petrocelli Books, 1974. v. 1. Citado 2 vezes nas páginas 10 e 16.
- NOGARE, D. *Entendendo como funciona o algoritmo de Cluster K-Means*. 2015. Acesso em: 11 jun. 2018. Disponível em: <<http://www.diegonogare.net/2015/08/entendendo-como-funciona-o-algoritmo-de-cluster-k-means/>>. Citado 3 vezes nas páginas 29, 30 e 31.
- PAVLOV, I. P. *Conditioned Reflexes and Psychiatry - Lectures on Conditioned Reflexes*. Read Books Limited, 2013. v. 2. Acesso em: 24 nov. 2019. ISBN 9781473387751. Disponível em: <<https://books.google.pt/books?id=qxh-CgAAQBAJ>>. Citado 4 vezes nas páginas 10, 23, 24 e 42.
- PEDREGOSA, F. et al. *Scikit-learn: Machine Learning in Python*. [S.l.]: Journal of Machine Learning Research, 2011. Citado na página 32.
- POSTMUS, S. Recommender system techniques applied to netflix movie data. *Vrije Universiteit Amsterdam*, 2018. Acesso em: 10 jan. 2019. Disponível em: <https://beta.vu.nl/nl/Images/werkstuk-postmus_tcm235-877824.pdf>. Citado na página 19.

- PROVOST, F. *Big Data*. [S.l.]: Mary Ann Liebert, Inc, 2013. v. 1. Citado na página 19.
- RICCI, F. *Recommender Systems Handbook*. [S.l.]: Springer, 2011. v. 1. 1-35 p. Citado na página 25.
- RUSSELL, S.; NORVIG, P. *Inteligência artificial*. CAMPUS - RJ, 2004. ISBN 9788535211771. Disponível em: <<https://books.google.pt/books?id=wBMvAAAACAAJ>>. Citado na página 20.
- SAWANT, S. Yelp food recommendation system. *Stanford University*, 2013. Acesso em: 01 maio 2019. Disponível em: <<http://cs229.stanford.edu/proj2013/SawantPai-YelpFoodRecommendationSystem.pdf>>. Citado 4 vezes nas páginas 11, 19, 27 e 43.
- SCHWARTZ, B. *O Paradoxo da Escolha: por que mais é menos*. [S.l.]: Editora Girafa, 2007. v. 1. Citado na página 12.
- STATISTA. *Do you trust online customer reviews as much as personal recommendations?* 2014. Acesso em: 11 abr. 2019. Disponível em: <<https://www.statista.com/statistics/315755/online-custmer-review-trust/>>. Citado na página 26.
- TAKAHASHI, M. M. Estudo comparativo de algoritmos de recomendação. *Instituto de Matemática e Estatística – Universidade de São Paulo*, 2015. Acesso em: 01 maio 2019. Disponível em: <https://bcc.ime.usp.br/tccs/2014/marcost/monografia_final.pdf>. Citado na página 10.
- THORNDIKE, R. L. Who belong in the family? *Psychometrika*, v. 18, p. 267–276, 02 1953. Citado na página 32.
- TUKEY, J. W. The future of data analysis. *The Annals of Mathematical Statistics*, p. 1–67, 1962. Citado 2 vezes nas páginas 10 e 16.
- TUKEY, J. W. *Exploratory Data Analysis*. [S.l.]: Pearson, 1977. v. 1. Citado na página 10.
- TURBAN, E. et al. *Business Intelligence: Um enfoque gerencial para a inteligência do negócio*. [S.l.]: Bookman, 2009. v. 1. ISBN 9788577803347. Citado na página 16.
- ZIVIANI, A. Ciência de dados: desafio para a ciência, indústria e governo. *Revista Eletrônica De Jornalismo Científico*, 2015. Acesso em: 15 fev. 2019. Disponível em: <https://www.researchgate.net/publication/279964734_Ciencia_de_dados_desafio_para_a_ciencia_industria_e_governo>. Citado na página 17.

A APÊNDICE: Carta de solicitação de autorização para uso de dados



São Paulo, dezembro de 2019.

Prezados Senhores,

Gostaria de convidá-los a participar de um estudo acadêmico da área de Ciência da Computação, cujo resultado será representado na forma de um Trabalho de Conclusão de Curso (TCC) do curso de Bacharelado em Ciência da Computação do Centro Universitário Senac.

O objetivo deste trabalho é desenvolver um sistema de recomendação de restaurantes utilizando técnicas de aprendizagem de máquina para modelos não supervisionados.

O trabalho está sendo realizado pelo aluno Paulo Henrique Fernandes Leite, regularmente matriculado sob o nº 1140073234, no curso de Bacharelado em Ciência da Computação desta Instituição de Ensino Superior e meu orientado neste TCC.

Solicitamos autorização para acesso e utilização dos dados de seus sistemas, considerando que todas as informações obtidas serão utilizadas exclusivamente para fins desta pesquisa e serão mantidas sob sigilo e confidencialidade, não sendo utilizada para quaisquer outras finalidades.

Informamos ainda que não serão apresentados dados ou informações que possam vir a caracterizar ou identificar os clientes ou pessoas da base de dados da instituição pesquisada.

O foco do trabalho é desenvolver um sistema de recomendação de restaurantes por meio do histórico de reservas, lista de restaurantes favoritos e preferências dos usuários. Também é esperado por meio de testes com usuários, validar o funcionamento e aceitação do sistema. Estes resultados serão compartilhados ao final da pesquisa para sua organização.

No aguardo de um pronunciamento por parte dos Senhores, colocamo-nos à disposição para esclarecimentos que se fizerem necessário.

Em nome do Centro Universitário Senac, na figura de orientadora deste trabalho, meus sinceros votos de agradecimento, estima e consideração.

Prof. Dr. Alexandre dos Santos Mignon

alexandre.smignon@sp.senac.br

Centro Universitário Senac

Av. Eng. Eusébio Stevaux, 823, São Paulo – SP

(11) 5682-7545

B APÊNDICE: Carta de solicitação de autorização para teste com usuário



São Paulo, dezembro de 2019.

Prezados Senhores,

Gostaria de convidá-los a participar de um estudo acadêmico da área de Ciência da Computação, cujo resultado será representado na forma de um Trabalho de Conclusão de Curso (TCC) do curso de Bacharelado em Ciência da Computação do Centro Universitário Senac.

O objetivo deste trabalho é desenvolver um sistema de recomendação de restaurantes utilizando técnicas de aprendizagem de máquina para modelos não supervisionados.

O trabalho está sendo realizado pelo aluno Paulo Henrique Fernandes Leite, regularmente matriculado sob o nº 1140073234, no curso de Bacharelado em Ciência da Computação desta Instituição de Ensino Superior e meu orientado neste TCC.

Solicitamos autorização para a utilização dos dados obtidos durante a realização dos testes do sistema desenvolvido em novembro de 2019, considerando que todas as informações obtidas serão utilizadas exclusivamente para fins desta pesquisa e serão mantidas sob sigilo e confidencialidade, não sendo utilizada para quaisquer outras finalidades.

Informamos ainda que não serão apresentados dados ou informações que possam vir a caracterizar ou identificar as pessoas que participaram dos testes do sistema desenvolvido.

O foco do trabalho é desenvolver um sistema de recomendação de restaurantes por meio do histórico de reservas, lista de restaurantes favoritos e preferências dos usuários. Também é esperado por meio de testes com usuários, validar o funcionamento e aceitação do sistema. Estes resultados serão compartilhados ao final da pesquisa para sua organização.

No aguardo de um pronunciamento por parte dos Senhores, colocamo-nos à disposição para esclarecimentos que se fizerem necessário.

Em nome do Centro Universitário Senac, na figura de orientadora deste trabalho, meus sinceros votos de agradecimento, estima e consideração.

Prof. Dr. Alexandre dos Santos Mignon

alexandre.smignon@sp.senac.br

Centro Universitário Senac

Av. Eng. Eusébio Stevaux, 823, São Paulo – SP

(11) 5682-7545