



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Instituto de Matemática e Estatística

Evellyn de Assis Pinto

**Sistema de recomendação inteligente para nutrição usando
aprendizado de máquina.**

Rio de Janeiro

2020

Evellyn de Assis Pinto

Sistema de recomendação inteligente para nutrição usando aprendizado de máquina.



Dissertação apresentada como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro.

Orientadores: Prof.^a Dra. Rosa Maria Esteves Moreira da Costa
Prof.^a Dra. Nayat Sanchez Pi

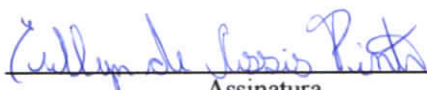
Rio de Janeiro
2020

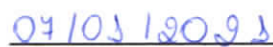
CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC-A

| | |
|------|---|
| P659 | <p>Pinto, Evellyn de Assis. Sistema de recomendação inteligente para nutrição usando aprendizado de máquina / Evellyn de Assis Pinto. – 2020. 80 f. : il.</p> <p>Orientadores: Rosa Maria Esteves Moreira da Costa, Nayat Sanchez-Pi. Dissertação (Mestrado em Ciências Computacionais) - Universidade do Estado do Rio de Janeiro.</p> <p>1. Software – Desenvolvimento – Teses. 2. Hábitos alimentares – Teses. 3. Rótulos – Aspectos nutricionais. 4. Aplicativos móveis – Teses. I. Costa, Rosa Maria Esteves Moreira da. II. Sanchez-Pi, Nayat. III. Universidade do Estado do Rio de Janeiro. Instituto de Matemática e Estatística. IV. Título.</p> <p>CDU 004.415.2</p> |
|------|---|

Patricia Bello Meijinhos CRB-7/ 5217- Bibliotecária responsável pela elaboração da ficha catalográfica

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta
dissertação, desde que citada a fonte.


Assinatura


Data

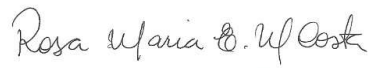
Evellyn de Assis Pinto

Sistema de recomendação inteligente para nutrição usando aprendizado de máquina.

Dissertação apresentada como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro.

Aprovada em 19 de março de 2020.

Banca Examinadora:



Prof^a. Dra. Rosa Maria Esteves Moreira da Costa (Orientador)
Instituto de Matemática e Estatística - UERJ



Prof. Dr. Luis Alfredo Vidal de Carvalho
Universidade Federal do Rio de Janeiro - UFRJ



Prof^a. Dra. Vera Maria Benjamin Werneck
Instituto de Matemática e Estatística - UERJ

Rio de Janeiro

2020

AGRADECIMENTOS

Agradeço primeiramente à Deus por conseguir realizar mais um sonho de concretizar esta pesquisa. Aos meus pais, Genésio Paulino Pinto e Oreni Nicolina Pinto, que sempre me apoiaram em todos os momentos de estudos e nas dificuldades. Ao meu irmão, Giordano, que sempre me incentivou a galgar cada degrau. As minhas avós Julieta Pinto e Rosa Maria de Assis (in memoriam). As minhas tias Arlete Júlia Pinto, Idvirgens Nicolina e Maria Júlia Pinto que me apoiaram durante o período de escola. Aos meus professores do ensino médio. Aos professores da Universidade Estácio de Sá do campus de Niterói que foi o primeiro passo para essa grande jornada. Aos meus colegas de trabalho do Telecentro de Niterói, que foi o meu primeiro estágio da área de informática. Aos meus colegas de trabalho do INT e em especial aos meus orientadores Janete Cícero e Saul Mizrahi que foram grandes apoiadores durante todo o período em que fui bolsista. Ao professor José Otávio Motta Pompeu e Silva por sua grande contribuição de conteúdo sobre assuntos relacionados com o tema. Ao professor Rubens Melo e os meus colegas da pós-graduação na PUC-RJ. Aos meus colegas de trabalho da Tecgraf que me ajudaram e apoiaram sempre com boas novidades tecnológicas, em especial ao Eduardo Gaspar, que incentivou no início da pesquisa quando realizei a palestra sobre este assunto, ao Guilherme Szundy, Leandro Nazareth e a Roberta Netto que sempre me apoiaram durante esse período de trabalho na Tecgraf. Aos professores da UERJ, o Fabiano Oliveira pelas aulas inesquecíveis de algoritmo, a Cristiane Faria e a Zochil Arenas pelas incríveis aulas de matemática e pelo apoio além da sala de aula, a professora Vera Maria Benjamin Werneck, pela colaboração e também aos colegas de classe. E em especial para as orientadoras Nayat Sanchez Pi e Rosa Maria Esteves Moreira da Costa, e ao professor Luis Martí por toda ajuda e colaboração na confecção deste projeto.

RESUMO

PINTO, Evellyn de Assis. *Sistema de recomendação inteligente para nutrição usando aprendizado de máquina*. 2020. 83 f. Dissertação (Mestrado em Ciências Computacionais) - Instituto de Matemática, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2020.

Esta pesquisa usa machine learning para criar um sistema de recomendação contendo itens mais saudáveis que os itens de uma lista base de compras de supermercado feita por um usuário. Para ajudar os consumidores na escolha de produtos melhores ao criar uma lista de compras, e levando em consideração a variedade de produtos e marcas que são comercializados, apresentamos um estudo sobre as técnicas de recomendações extraídas de conjuntos de dados de Open Food Facts usando rótulos de produtos. A metodologia utilizada foi uma pesquisa experimental, e criado um sistema para um aplicativo móvel que usa a recomendação baseado em conteúdo (RICCI; ROKACH; SHAPIRA, 2015) para testar e treinar esse conjunto de dados.

Palavras-chave: Machine Learning. Sistema de Recomendação. Similaridade. CountVectorizer. TF-IDFVectorizer. Lista de Supermercado.

ABSTRACT

PINTO, Evellyn de Assis. *Intelligent recommendation system for nutrition using machine learning*. 2020. 83 f. Dissertação (Mestrado em Ciências Computacionais) - Instituto de Matemática, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2020.

This research uses machine learning to create a recommendation system containing healthier items to user include in grocery shopping list. To assist consumers in choosing the best products by creating a shopping list, and taking into account the variety of products and brands that are marketed, we present a study on the recommendation techniques extracted from Open Food Facts datasets using consumer labels products. The methodology used was an experimental research, and a system was created for a mobile application that uses content-based recommendation (RICCI; ROKACH; SHAPIRA, 2015) to test and train this dataset.

Keywords: Machine Learning. Recommendation System. Similarity. CountVec-torizer. TF-IDFVectorizer. Grocery list

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 – Fluxo do trabalho | 14 |
| Figura 2 – Representações gráficas de duas categorias de dados | 18 |
| Figura 3 – Probabilidades previstas de um valor default usando a regressão logística | 21 |
| Figura 4 – KNN | 23 |
| Figura 5 – Matrix factorization | 24 |
| Figura 6 – Arquitetura funcional dos Sistemas de Recomendação | 31 |
| Figura 7 – Recomendação baseada em filtro colaborativo | 32 |
| Figura 8 – Recomendação baseada em conteúdo | 34 |
| Figura 9 – Medida de similaridade entre dois vetores. | 38 |
| Figura 10 – Diagrama de classes do Lista Saudável | 54 |
| Figura 11 – Tela de login do aplicativo Lista Saudável | 55 |
| Figura 12 – Representação do projeto do Sistema de recomendação do Lista Saudável | 56 |
| Figura 13 – Tela de criação de lista de produtos do aplicativo Lista Saudável | 57 |
| Figura 14 – Tela de criação de lista de produtos do aplicativo Lista Saudável | 58 |
| Figura 15 – Tela da lista recomendada pelo aplicativo Lista Saudável | 59 |
| Figura 16 – Quantidade de produtos da França por nutriscore | 61 |
| Figura 17 – Quantidade de produtos da França por categoria | 62 |
| Figura 18 – Quantidade de produtos do Brasil por categoria | 63 |
| Figura 19 – Melhor valor de k para o algoritmo - KNN | 65 |
| Figura 20 – Produto cadastrado no Open Food Facts - Feijão fradinho pote | 66 |
| Figura 21 – Produto cadastrado no Open Food Facts - Feijão manteiga cozido | 67 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – Aplicativos móveis para compra de produtos de supermercado | 41 |
| Tabela 2 – Comparativo do uso de tecnologia dos aplicativos móvel para nutrição. . | 48 |
| Tabela 3 – Resultados com 75% do <i>Dataset</i> para treino e 25% para teste. | 63 |
| Tabela 4 – Resultados com 70% para treino e 30% para teste. | 64 |
| Tabela 5 – Resultados com 60% para treino e 40% para teste. | 64 |
| Tabela 6 – Lista de produtos escolhidos pelo usuário | 68 |
| Tabela 7 – Lista de produtos recomendados pelo Lista Saudável | 69 |

LISTA DE ALGORITMOS

| | |
|--|----|
| Algoritmo 1 – Similaridade de produto baseado em Nutriscore. | 80 |
|--|----|

SUMÁRIO

| | | |
|---------|--|----|
| | INTRODUÇÃO | 10 |
| 1 | FUNDAMENTAÇÃO TEÓRICA | 15 |
| 1.1 | Inteligência Artificial | 15 |
| 1.1.1 | <u>Máquinas Inteligentes</u> | 15 |
| 1.1.2 | <u>Aprendizagem de máquina</u> | 18 |
| 1.1.2.1 | Regressão logística | 20 |
| 1.1.2.2 | ANN | 21 |
| 1.1.2.3 | KNN | 22 |
| 1.1.2.4 | Matrix factorization | 23 |
| 1.1.2.5 | Association rules | 25 |
| 1.1.2.6 | Gradient boosting | 25 |
| 1.1.2.7 | Métricas de avaliação da aprendizagem de máquina | 26 |
| 1.2 | Sistemas de Recomendação | 28 |
| 1.2.1 | <u>Técnicas de recomendação</u> | 31 |
| 1.2.1.1 | Filtragem colaborativa | 31 |
| 1.2.1.2 | Sistemas baseados em conteúdo | 33 |
| 1.2.1.3 | Sistemas híbridos | 33 |
| 1.3 | Técnicas de identificação de Similaridades entre textos | 34 |
| 1.3.1 | <u>TF-IDFVectorizer</u> | 35 |
| 1.3.2 | <u>CountVectorizer</u> | 36 |
| 1.3.3 | <u>Cosine similarity</u> | 38 |
| 1.4 | Comentários Finais | 39 |
| 2 | TRABALHOS CORRELATOS | 40 |
| 2.1 | Aplicativos | 40 |
| 2.2 | Artigos | 44 |
| 2.3 | Considerações sobre o capítulo | 48 |

| | | |
|-------|---|----|
| 3 | IMPLEMENTAÇÃO E AVALIAÇÃO DE SISTEMA DE RECOMENDAÇÃO PARA NUTRIÇÃO | 50 |
| 3.1 | Recomendação baseada no Nutriscore | 50 |
| 3.1.1 | <u>Implementação de Sistema de recomendação para nutrição</u> | 52 |
| 3.2 | Metodologia | 53 |
| 3.3 | Recomendação de lista de compra utilizando o Lista Saudável . . . | 55 |
| 3.4 | CONSIDERAÇÕES FINAIS | 60 |
| 4 | TESTES | 61 |
| 4.1 | Criação de um dataset para os testes | 61 |
| 4.2 | Ajuste dos parâmetros | 64 |
| 4.3 | Avaliação do sistema de recomendação | 65 |
| 4.4 | Considerações sobre o Experimento | 70 |
| | CONCLUSÕES E TRABALHOS FUTUROS | 71 |
| | REFERÊNCIAS | 73 |

INTRODUÇÃO

Segundo a OMS (Organização Mundial da Saúde) (OMS, 2019), a doença isquêmica do coração é a principal causa de morte no mundo. Estima-se que, até 2016, as patologias como a cardiopatia isquêmica e o acidente vascular cerebral foram responsáveis por 15,2 milhões de óbitos. Sendo que, essas doenças têm sido as principais causas de morte no mundo nos últimos 15 anos.

No contexto do Brasil, Cantieri, Bueno e Martinez-Ávila (2018) afirmam que, em média, 70% da mortalidade no país ocorre por conta de Doenças Crônicas Não Transmissíveis (DCNT).

Os fatores de risco mais importantes para a mortalidade relacionada às DCNT são: hipertensão arterial sistêmica, hipercolesterolemia, sobrepeso ou obesidade, inatividade física e tabagismo. A predisposição genética, a alimentação inadequada e a inatividade física estão entre os principais fatores que contribuem para o surgimento da Síndrome Metabólica (SM), que é um transtorno complexo representado por um conjunto de fatores de risco cardiovascular, usualmente relacionados à deposição central de gordura e à resistência à insulina, cuja prevenção primária é um desafio mundial contemporâneo, com importante repercussão para a saúde. O aumento da prevalência da obesidade no Brasil é uma tendência especialmente preocupante dessa síndrome, que atinge crianças em idade escolar, adolescentes e pessoas que possuem baixa renda. A adoção precoce por toda a população de estilos de vida relacionados à manutenção da saúde, como dieta adequada e prática regular de atividade física, preferencialmente desde a infância, é componente básico da prevenção da SM (TRATAMENTO, 2005).

De acordo com as pesquisas do Heart, Institute et al. (2006) o corpo produz o colesterol sérico LDL (low density lipoprotein), esse colesterol carrega as partículas de colesterol do fígado e de outros locais para as artérias. O colesterol viaja em pacotes chamados lipoproteínas, que têm gordura (lipídios) no interior e proteínas no exterior. Dois tipos principais de lipoproteínas carregam colesterol no sangue: a lipoproteína de baixa densidade, ou LDL, que também é chamada de colesterol "ruim" porque transporta colesterol para os tecidos, incluindo as artérias. A maior parte do colesterol no sangue está na

forma de LDL. Quanto maior o nível de colesterol LDL no sangue, maior o risco de doença cardíaca. Já a lipoproteína de alta densidade, ou HDL (high density lipoprotein), que também é chamado de colesterol "bom", transporta colesterol dos tecidos para o fígado, que o remove do corpo. Um baixo nível de colesterol HDL aumenta o risco de doença cardíaca. Para a Sociedade Brasileira de Cardiologia (SBC, 2019), a realização de um plano alimentar para a redução de peso, associado a exercício físico são considerados terapias de primeira escolha para o tratamento de pacientes com síndrome metabólica e está comprovado que esta associação provoca a redução expressiva da circunferência abdominal e a gordura visceral, melhora significativamente a sensibilidade à insulina, diminui os níveis plasmáticos de glicose, podendo prevenir e retardar o aparecimento de diabetes tipo 2. Há ainda, com essas duas intervenções, uma redução expressiva da pressão arterial e nos níveis de triglicérides, com aumento do HDL-colesterol.

A alimentação adequada deve: permitir a manutenção do balanço energético e do peso saudável; reduzir a ingestão de calorias sob a forma de gorduras, mudar o consumo de gorduras saturadas para gorduras insaturadas, reduzir o consumo de gorduras trans (hidrogenada); aumentar a ingestão de frutas, hortaliças, leguminosas e cereais integrais; reduzir a ingestão de açúcar livre; reduzir a ingestão de sal (sódio) sob todas as formas.

Então, um dos desafios para alcançar a dieta adequada é preparar uma lista de compras com produtos menos calóricos. Considerando a variedade de produtos e marcas que são comercializados em um supermercado, torna-se difícil avaliar quais são os alimentos mais saudáveis dentre eles.

Visando amenizar esta problemática, este trabalho propõe um sistema de recomendação de produtos alimentícios mais saudáveis.

Objetivo

O objetivo deste trabalho é desenvolver um sistema de recomendação para um dispositivo móvel, visando ajudar o consumidor na escolha de alimentos, propondo opções que contenham, por exemplo, menor teor de açúcar e sal ou menor quantidade de calorias, para, então, incluir em sua lista de compras.

Para alcançar esse objetivo, foi realizada uma revisão de literatura sobre os Sistemas de Recomendação utilizados em dispositivos móveis com o objetivo de identificar as técnicas e estratégias que são utilizadas por esses sistemas. Ao final dessa revisão, foi

constatado que os trabalhos estudados, não exploram o apoio a realização de compras em um supermercado ((HAJJDIAB et al., 2018); (ANTON et al., 2018) e (LEIPOLD et al., 2018)). Observou-se que a pesquisa de Mackenzie (MACKENZIE, 2018) abordou o tema e desenvolveu um aplicativo que automatiza todo o processo de criação de uma lista de compras, visando oferecer o mínimo de esforço e tempo gasto pelo usuário, mas não se preocupa em oferecer uma lista de compras contendo itens mais saudáveis.

Assim sendo, a principal motivação para este trabalho é demonstrar que por meio de um aplicativo para dispositivos móveis, ou celular, os usuários com características de perfis específicos podem criar listas de supermercado contendo itens mais saudáveis.

Para criar a base de dados com a classificação dos produtos brasileiros, partiu-se da classificação da qualidade dos componentes de produtos alimentícios europeus. A estratégia de aprendizado de máquina utilizou o padrão europeu para classificar os produtos brasileiros. Ou seja, o padrão europeu foi utilizado para treinamento e em seguida, os produtos brasileiros foram classificados. A técnica de Algoritmos de recomendação foi selecionada para a implementação do sistema final por ser umas das técnicas de Inteligência Artificial mais utilizadas para auxiliar usuários no processo de tomada de decisão.

Motivação

Assistir pessoas na tarefa de criar uma lista de compras de supermercados personalizada, ou seja, contendo itens selecionados a partir de uma prévia consulta em sua tabela nutricional. Levando em consideração o número de variedade dos produtos comercializados, essa simples tarefa demandaria muito tempo para determinar uma escolha definitiva, e dessa forma, selecionar um produto para compor o seu carrinho de compras.

Baseado no Programa Nacional de Nutrição e Saúde da França (PNNS) (PNNS, 2019), que foi lançado 2001, o PNNS é um plano público destinado a melhorar o estado de saúde da população agravada por um dos principais fatores: a nutrição.

Por conseguinte, para auxiliar o controle da seleção dos produtos destinados para um consumo, o sistema proposto no objetivo deste trabalho, irá oferecer ao usuário a criação de uma lista de compras com produtos mais saudáveis, sendo embasado pela seleção anterior realizada pelo próprio consumidor.

Fundamentado por uma revisão de literatura dos aplicativos móveis relacionados com supermercados, constatamos que o aplicativo Shopwell (SHOPWELL, 2019) é gratuito,

simplifica os rótulos nutricionais e ajuda a descobrir novos alimentos adequados ao estilo de vida de cada usuário, porém não oferece o suporte para criação de uma lista de compras personalizada de itens de supermercados.

Portanto, este trabalho também irá apoiar pesquisas relacionadas com assuntos sobre aplicativos de compras de supermercado, visto que para Aiolfi (2019), ainda são poucos estudos relacionados com este assunto.

Estrutura

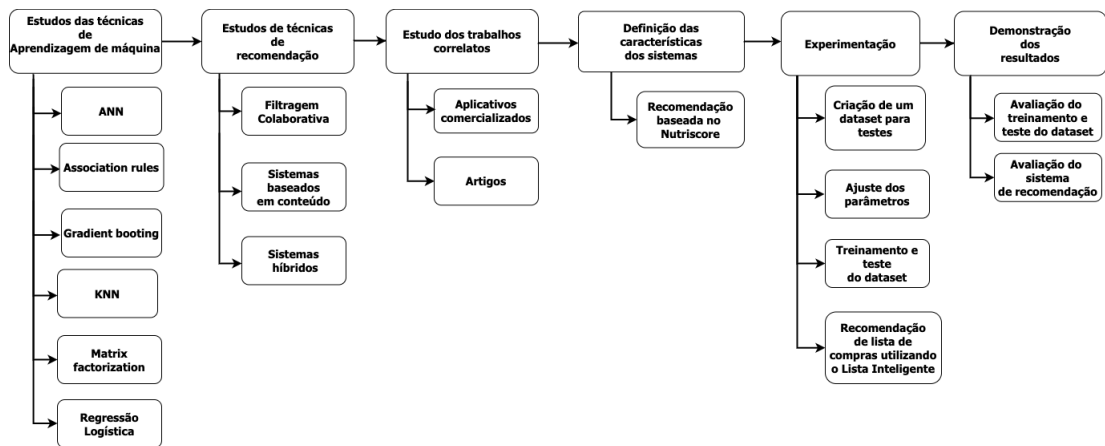
O trabalho está organizado da seguinte forma:

- Capítulo 1: apresenta definições e nomenclaturas necessárias para a compreensão de aspectos fundamentais das áreas de Inteligência Artificial. Este capítulo detalha e ilustra estudos de *machine learning*, e caracterizando os sistemas de recomendação existentes: baseado em filtragem colaborativa, baseados em conteúdo e sistemas híbridos, além da apresentação da construção do conjunto de produtos similares usando a técnica de vetorização. Além disso, descreve a aplicabilidade de aprendizado de máquina em sistemas de recomendação. E também, são apresentadas definições dos algoritmos para esses sistemas: *Regressão logística*, *ANN*, *KNN*, *Matrix factorization*, *Association rules*, *Gradiente booting*. Em seguida, são descritas as métricas para avaliar o resultado do treinamento do *dataset* utilizado nesses sistemas.
- Capítulo 2: discute os problemas relacionados a criação de uma lista de compras contendo produtos mais saudáveis, ou seja, menos calóricos, com menor teor de sal, açúcar, adoçantes, corantes, e etc. Além disso, descreve os trabalhos correlatos, destacando suas funcionalidades e tecnologias adotadas.
- Capítulo 3: relata a implementação do algoritmo de recomendação baseado em conteúdo, utilizando o conceito de similaridade, sendo assim, descreve como recomendar produtos baseado em seu nutriscore. Em seguida, são apresentadas as metodologias utilizadas para o desenvolvimento do sistema, destacando a utilização de *TF-IDF* e *CountVectorizer*, para criar um vetor de palavras utilizando como entrada o nome do produto.

- Capítulo 4 apresenta os resultados preliminares com comparativos sobre os testes realizados com os dados selecionados.
- Por fim, apresentamos as conclusões da dissertação em conjunto com as propostas de trabalhos futuros seguidos pela bibliografia e Apêndice 1, com o algoritmo desenvolvido.

A figura 1 apresenta em detalhamento os passos para a realização do trabalho.

Figura 1 – Fluxo do trabalho



1 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados alguns conceitos de Inteligência Artificial, destacando suas bases teóricas.

1.1 Inteligência Artificial

O conceito de Inteligência Artificial (IA) foi proposta pela primeira vez por John McCarthy em 1956 em uma conferência sobre o assunto. A ideia de máquinas operando como seres humanos começou a ser o assunto central da pesquisa do cientista, visando fazer máquinas que tivessem capacidades de aprender por si só.

Desde então, a área vem recebendo atenção de pesquisadores com diferentes perfis, na busca de identificar técnicas capazes de apoiar processos computacionais mais inteligentes. Nesse sentido, a Inteligência Artificial é definida como o campo de estudo que explora estratégias específicas para tornar as decisões da máquina mais próximas àquelas realizadas pelos seres humanos (ALSEDRAH, 2017).

Dentre as várias técnicas exploradas neste domínio, a Aprendizagem de Máquina vem se destacando, por utilizar base de dados em processos iterativos e automatizados, para generalizar a partir da experiência prévia sobre alguma área ou problema (LUGER, 2014).

1.1.1 Máquinas Inteligentes

Quando os computadores programáveis foram concebidos pela primeira vez, as pessoas se perguntavam se essas máquinas poderiam se tornar inteligentes, mais de cem anos antes de uma ser construída (Lovelace 1842)(GOODFELLOW; BENGIO; COURVILLE, 2016). Hoje, a inteligência artificial (IA) é um campo próspero, com muitas aplicações práticas e tópicos de pesquisa ativos. Buscamos software inteligente para automatizar o trabalho de rotina, entender fala ou imagens, fazer diagnósticos em medicina e apoiar pesquisas científicas básicas.

Nos primeiros desafios da inteligência artificial foram resolvidos problemas intelectualmente difíceis para os seres humanos, mas relativamente diretos para os computadores - problemas que podem ser descritos por uma lista de regras formais e matemáticas. Em

geral, a IA resolvia problemas fáceis de executar, mas difíceis de descrever formalmente - problemas que resolvemos intuitivamente, que parecem automáticos, como reconhecer palavras, ou rostos em imagens. As primeiras soluções permitiam que os computadores aprendessem com a experiência e compreendessem o mundo em termos de uma hierarquia de conceitos, com cada conceito definido em termos de sua relação com conceitos mais simples. Ao reunir conhecimentos a partir da experiência, essa abordagem evita a necessidade de operadores humanos especificarem formalmente todo o conhecimento necessário ao computador. Ou seja, a hierarquia de conceitos permite que o computador aprenda conceitos complicados construindo-os a partir de conceitos mais simples. Se desenharmos um gráfico mostrando como esses conceitos são construídos, o gráfico é profundo, com muitas camadas. Por esse motivo, chamamos essa abordagem de IA de Aprendizagem profunda ou ainda *Deep learning*. Muitos dos sucessos iniciais da IA ocorreram em ambientes relativamente estéreis e formais e não exigiam que os computadores tivessem muito conhecimento sobre o mundo. Por exemplo, o sistema de xadrez Deep Blue da IBM derrotou o campeão mundial Garry Kasparov em 1997 (*Hsu 2002*). O xadrez é muito simples, contendo apenas sessenta e quatro locais e trinta e duas peças que só podem se mover de maneira rigidamente circunscrita. Criar uma estratégia de xadrez bem-sucedida é uma tremenda conquista, mas o desafio não se deve à dificuldade de descrever o conjunto de peças de xadrez e os movimentos permitidos para o computador. O xadrez pode ser completamente descrito por uma lista muito breve de regras completamente formais, facilmente fornecidas com antecedência pelo programador. Ironicamente, tarefas abstratas e formais que estão entre os empreendimentos mentais mais difíceis para um ser humano estão entre as mais fáceis para um computador. Os computadores há muito conseguem derrotar até o melhor jogador de xadrez humano, mas apenas recentemente, estão considerando algumas das habilidades dos seres humanos comuns para reconhecer objetos, ou a fala.

A vida cotidiana de uma pessoa requer uma imensa quantidade de conhecimento sobre o mundo. Grande parte desse conhecimento é subjetivo e intuitivo e, portanto, difícil de articular de maneira formal. Os computadores precisam capturar esse mesmo conhecimento para se comportarem de maneira inteligente.

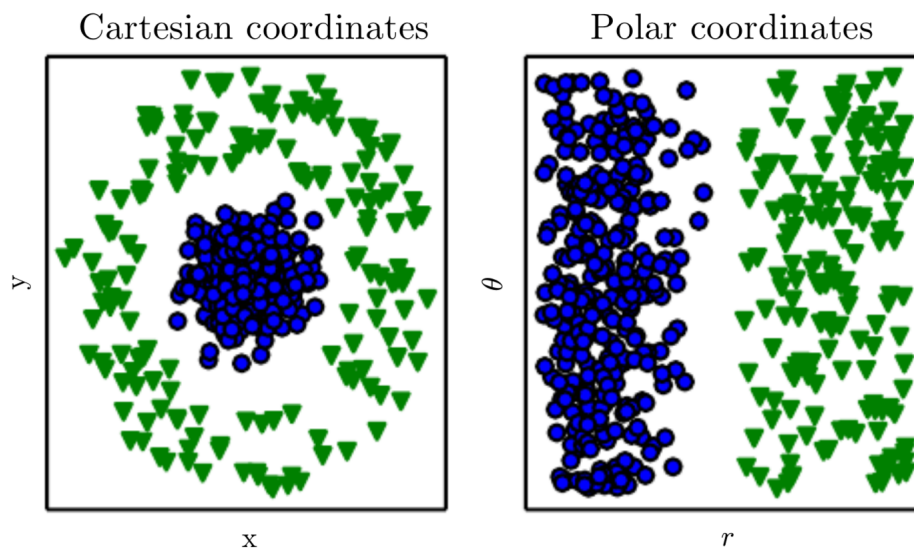
Um dos principais desafios da inteligência artificial é como colocar esse conhecimento informal em um computador. Vários projetos de inteligência artificial procuraram co-

dificar o conhecimento sobre o mundo em linguagens formais. Um computador pode raciocinar sobre declarações nessas linguagens formais automaticamente usando regras de inferência lógica. Isso é conhecido como a abordagem da base de conhecimento da inteligência artificial. Nenhum desses projetos levou a um grande sucesso. Um dos projetos mais famosos é Cyc (Lenat e Guha 1989). Cyc é um mecanismo de inferência e um banco de dados de instruções em uma linguagem chamada *CycL*. Essas declarações são inseridas por uma equipe de supervisores humanos. É um processo pesado. As pessoas lutam para criar regras formais com complexidade suficiente para descrever com precisão o mundo. Por exemplo, Cyc não conseguiu entender uma história sobre uma pessoa chamada Fred se barbear pela manhã. Sua inferência, o mecanismo *Linde 1992* detectou uma inconsistência na história: sabia que as pessoas não têm partes elétricas, mas como Fred estava com um barbeador elétrico, acreditava que a entidade “FredWhileShaving” continha partes elétricas. Por isso, perguntou se Fred ainda era uma pessoa enquanto fazia a barba. As dificuldades enfrentadas pelos sistemas que dependem de conhecimento codificado sugerem que os sistemas de inteligência artificial precisam da capacidade de adquirir seu próprio conhecimento, extraindo padrões de dados brutos. Esse recurso é conhecido como aprendizado de máquina. A introdução do aprendizado de máquina permitiu que os computadores resolvessem problemas, que envolviam o conhecimento do mundo real e tomassem decisões, que parecessem subjetivas. Um algoritmo simples de aprendizado de máquina chamado regressão logística pode determinar se é recomendado o parto cesáreo (Mor-Yosef 1990). Um algoritmo simples de aprendizado de máquina chamado Bayes ingênuo pode separar e-mail legítimo de e-mail *spam*. O desempenho desses algoritmos simples de aprendizado de máquina depende muito da representação dos dados fornecidos. Por exemplo, quando a regressão logística é usada para recomendar a cesariana, o sistema de IA não examina o paciente diretamente. Em vez disso, o médico fornece ao sistema várias informações relevantes, como a presença ou ausência de uma cicatriz uterina. Cada informação incluída na representação do paciente é conhecida como um recurso. A regressão logística aprende como cada um desses recursos do paciente se correlaciona com vários resultados. No entanto, não pode influenciar a maneira como os recursos são definidos de forma alguma. Se a regressão logística fosse submetida a uma ressonância magnética do paciente, em vez do relatório formalizado pelo médico, ele não seria capaz de fazer previsões úteis. *Pixels* individuais em uma ressonância magnética têm correlação insignificante

com quaisquer complicações que possam ocorrer durante o parto. Essa dependência de representações é um fenômeno geral que aparece na ciência da computação e até na vida cotidiana. Na ciência da computação, operações, como pesquisar uma coleção de dados, podem prosseguir exponencialmente mais rápido se a coleção for estruturada e indexada de maneira inteligente. As pessoas podem executar aritmética facilmente com algarismos arábicos, mas realizar cálculos com uma representação numérica em algarismos romanos seria mais demorada. Não é de surpreender que a escolha da representação tenha um efeito enorme no desempenho dos algoritmos de aprendizado de máquina.

Na figura 2 temos um exemplo de diferentes representações: suponha que desejamos separar duas categorias de dados, desenhando uma linha reta entre elas em um gráfico de dispersão. No gráfico à esquerda, representamos alguns dados usando coordenadas cartesianas, e a tarefa é impossível. No gráfico à direita, representamos os dados com coordenadas polares e a tarefa se torna simples de resolver com uma linha vertical.

Figura 2 – Representações gráficas de duas categorias de dados



Fonte:(GOODFELLOW; BENGIO; COURVILLE, 2016)

1.1.2 Aprendizagem de máquina

O aprendizado de máquina ou *Machine Learning* (ML), tornou-se um termo popular nas últimas década (KÜHL et al., 2019). São frequentemente usados em ciência e mídia, devido a sua capacidade de escreve um conjunto de técnicas que são comumente usados para resolver uma variedade de problemas do mundo real com a ajuda de sistemas de com-

putador que podem aprender a resolver um problema específico, por meio do treinamento de um conjunto de dados.

Esse campo de estudo deve sua adoção crescente a sua capacidade de caracterizar relacionamentos subjacentes dentro de grandes matrizes de dados, resolvendo problemas de análise em grandes bancos de dados (*big data*), reconhecimento de padrões comportamentais e evolução da informação (SU et al., 2012). Em geral, os sistemas que utilizam aprendizado de máquina podem ser treinados para categorizar as condições de processo, de forma a modelar variações no comportamento operacional. Como corpos de conhecimento evoluem sob a influência de novas ideias e tecnologias, esses sistemas podem identificar rupturas nos modelos existentes e redesenhar e reciclar-se para adaptar-se e co-evoluir com o novo conhecimento.

A sua característica computacional é generalizar a experiência de treinamento por meio de exemplos. O atributo de generalização do aprendizado de máquina permite que o sistema execute bem em instâncias de dados invisíveis, prevendo com precisão os dados futuros. O processo de generalização requer classificadores, que insiram recursos discretos, ou vetores de recursos contínuos e saída de uma classe. Ao contrário de outros problemas de otimização, esses sistemas não possuem uma função bem definida, que possa ser otimizada. Em vez disso, erros de treinamento servem como um catalisador para testar erros de aprendizagem.

Em 1959 Arthur Samuel descreveu o ML como o "campo de estudo que dá aos computadores a capacidade de aprender sem ser explicitamente programado". Ele concluiu que programar computadores para aprender com a experiência, deve eventualmente, eliminar a necessidade de muito desse esforço detalhado de programação [(AWAD; KHANNA, 2015) , (MURPHY, 2012)].

O Aprendizado de máquina possui três tipos de abordagem: aprendizado supervisionado, aprendizado não supervisionado e aprendizagem por reforço (LAMFO, 2019).

Aprendizado Supervisionado

O tipo de aprendizado supervisionado é muito empregado quando tentamos prever uma variável dependente (anos de carreira, formação, idade) a partir de uma lista de variáveis independentes (salário, férias).

A característica básica de sistemas de aprendizado supervisionado é que os dados que utilizamos para treiná-los contém a resposta desejada, isto é, contém a variável dependente

resultante das variáveis independentes observadas. Nesse caso, dizemos que os dados são anotados com as respostas, ou classes a serem previstas.

Para resolver problemas de aprendizado supervisionado dentre as técnicas mais conhecidas, podemos destacar: regressão linear, regressão logística, redes neurais artificiais, máquina de suporte vetorial (ou máquinas kernel), árvores de decisão, k-vizinhos mais próximos e Bayes ingênuo (Hugo Honda, Matheus Facure, Peng Yaohao, 2017).

Aprendizado não supervisionado

O aprendizado descritivo, ou não supervisionado, tem o objetivo de achar padrões nos dados. Isso é chamado de descoberta de conhecimento. Esse é um problema muito menos definido, pois não nos dizem que tipos de padrões procurar e não há métrica de erro óbvia a ser usada (ao contrário do aprendizado supervisionado, onde podemos comparar nossa previsão de y para um dado x , ao valor observado).

Aplicações de aprendizado não supervisionados são sistemas de recomendação de filmes ou músicas, detecção de anomalias, visualização de dados e etc. Dentre as técnicas mais conhecidas para resolver problemas de aprendizado não supervisionado estão redes neurais artificiais, expectativa-maximização, clusterização-médias, máquina de suporte vetorial (ou máquinas kernel), clusterização hierárquica, análise de componentes principais, florestas isoladoras, mapas auto-organizados, máquinas de Boltzmann restritas, eclat, apriori, t-SNE (Hugo Honda, Matheus Facure, Peng Yaohao, 2017).

Aprendizado por reforço

O aprendizado por reforço, é menos usado. Neste caso, a máquina tenta aprender qual é a melhor ação a ser tomada, dependendo das circunstâncias na qual essa ação será executada (Hugo Honda, Matheus Facure, Peng Yaohao, 2017).

Essas abordagens envolvem uma variedade de técnicas, sendo que, algumas delas são brevemente descritas a seguir.

1.1.2.1 Regressão logística

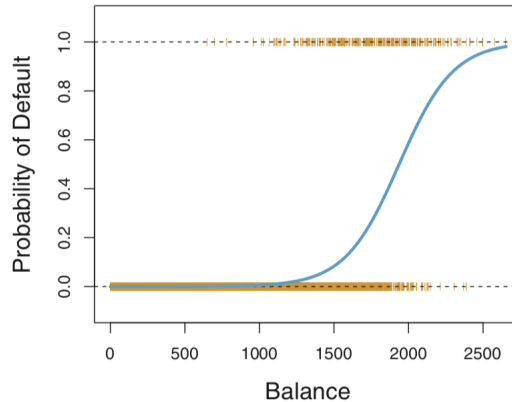
A regressão logística tem como objetivo produzir um modelo de predição associado à ocorrência de um evento em relação a um conjunto de variáveis independentes, em determinado domínio), em relação às variáveis independentes. Desta forma, a regressão

logística estima a probabilidade de certo evento ocorrer (DANGETI, 2017).

Existem dois modelos de regressão logística: regressão logística binária e regressão logística multinomial (OLKIN, 2002). A regressão logística binária é tipicamente usada quando a variável dependente é dicotômica e as variáveis independentes são contínuas, ou categóricas. A regressão logística multinomial é usada para prever a colocação categórica na probabilidade de ser membro da categoria em uma variável dependente, baseada em múltiplas variáveis. As variáveis independentes podem ser dicotômicas (ou seja, binárias), ou contínuas (ou seja, intervalo ou razão em escala, como ilustra a figura 3). A regressão logística multinomial é uma simples extensão da logística regressão binária, que permite mais de duas categorias da variável dependente, ou de resultado (SCHWAB, 2002).

Considerando um conjunto de dados, em que o padrão de resposta cai em uma das duas categorias, Sim ou Não, em vez de modelar essa determinada resposta diretamente, a regressão logística modela a probabilidade que uma resposta pertença a uma categoria específica.

Figura 3 – Probabilidades previstas de um valor default usando a regressão logística



Fonte:(OLKIN, 2002)

1.1.2.2 ANN

Segundo Andoni et al. (ANDONI; INDYK; RAZENSHTEYN, 2018), o problema do Approximate Nearest Neighbor (ANN) ou Vizinho mais Próximo Aproximado é definido da seguinte maneira: Dado um conjunto P de n pontos em algum espaço métrico (X, D) , crie uma estrutura de dados que, dado qualquer ponto q , retorne um ponto em P mais próximo de q (seu "vizinho mais próximo" em P). A estrutura de dados armazena informações adicionais sobre o conjunto P , que é usado para encontrar o vizinho mais próximo

sem calcular todas as distâncias entre q e P . O problema tem uma ampla variedade de aplicações em aprendizado de máquina, visão computacional, bancos de dados e outros campos. Para reduzir o tempo necessário para encontrar vizinhos mais próximos e a quantidade de memória usada pela estrutura de dados, pode-se formular o problema, onde o objetivo é retornar qualquer ponto $p' \in P$ de modo que a distância de q a p' é no máximo $c \cdot \min_{p \in P} D(q, p)$, para alguns $c \geq 1$.

1.1.2.3 KNN

O algoritmo de aprendizagem de máquina k-nearest neighbors algorithm (KNN) é um método classificador que utiliza técnicas de agrupamento. É altamente dependente da definição de uma medida de similaridade ou distância apropriada, considerando itens como vetores de documentos de espaço n-dimensional. Uma abordagem muito comum para medir a semelhança entre vetores, é calcular a distância *Euclidiana* (eq. 1.1), a outra maneira é calcular o ângulo do *coseno* entre eles. Esse classificador memoriza todo o conjunto de treinamento e classifica apenas se os atributos do novo registro correspondente exatamente a um dos exemplos de treinamento.

Distância Euclidiana:

A fórmula para medir a distância Euclidiana é dado por:

$$D(x,y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2} \quad (1.1)$$

onde, n é o número de dimensões (atributos) e x_k e y_k são os k 's atributos (componentes) dos objetos de dados x e y , respectivamente.

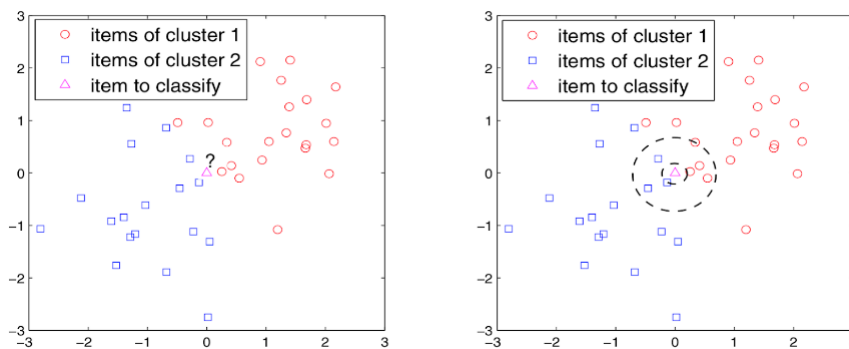
Dado um ponto a ser classificado, o classificador *KNN* encontra os k pontos mais próximos (vizinhos mais próximos) dos registros de treinamento. Em seguida, ele atribui o rótulo de classe de acordo com os rótulos de classe de seus vizinhos mais próximos. A ideia subjacente é que, se um registro cai em uma vizinhança em particular, onde um rótulo de classe é predominante, é porque o registro provavelmente pertence a essa mesma classe. Como o *KNN* não constrói modelos explicitamente, ele é considerado um aprendiz preguiçoso. Ao contrário de árvores de decisão ou sistemas baseados em

regras, que deixam muitas decisões para a etapa de classificação. Portanto, classificar registros desconhecidos é relativamente caro, pois não é necessário aprender e manter um determinado modelo, como demonstrado na figura 4. Mas, será necessário recalculer o posicionamento dos pontos à medida que ocorrem mudanças na matriz de similaridade.

O *KNN* é uma das abordagens mais comuns para para projetar um sistema de recomendação do tipo filtragem colaborativa. Talvez a questão mais desafiadora dessa abordagem seja como escolher o valor de k . E se k é muito pequeno, o classificador será sensível aos pontos de ruído. Mas se k é muito grande, a vizinhança pode incluir muitos pontos de outras classes (RICCI; ROKACH; SHAPIRA, 2015).

A figura 4 ilustra a classificação com *KNN* dos itens nos pontos de treinamento com dois rótulos de classe (círculos e quadrados) e o ponto de consulta (como um triângulo).

Figura 4 – KNN



Fonte: (RICCI; ROKACH; SHAPIRA, 2015)

1.1.2.4 Matrix factorization

Os modelos de aprendizagem de máquina, que são induzidos por fatorizar uma matriz de classificação de itens do usuário, são também conhecidos como modelos baseados em Singular Value Decomposition (SVD). Recentemente, modelos de fatoraçoão de matriz ganharam popularidade, graças à sua precisão atraente e escalabilidade, em abordagem de filtragem colaborativa, com o objetivo holístico de descobrir características latentes para identificar a semântica na recuperação de informação. No entanto, a aplicação de *SVD* a classificações explícitas no domínio de filtragem colaborativa levanta dificuldades devido à alta parcela de valores ausentes. O *SVD Convencional* é indefinido quando o conhecimento sobre a matriz é incompleto. Além disso, abordando descuidadamente poucas entradas conhecidas, tornar-se altamente propenso ao *overfitting* (é um problema

em redes neurais. Em redes modernas, que geralmente têm um grande número de pesos e vieses. Para treinar de forma eficaz, é necessário detectar quando o overfitting está acontecendo. E precisamos aplicar técnicas para reduzir).

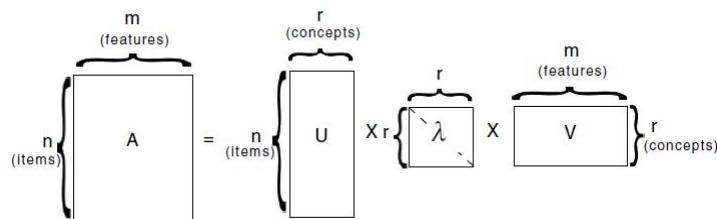
Vale ressaltar que lidamos com o fato de que as preferências por produtos pelo cliente podem variar com o tempo. A percepção do produto e a popularidade alteram constantemente à medida que surge nova seleção.

Da mesma forma, as inclinações do cliente vão evoluindo, levando-os a redefinir seu gosto. Isso leva a um modelo que aborda a dinâmica temporal para melhor acompanhar o comportamento do usuário.

O *SVD* (GOLUB; REINSCH, 1970) é uma técnica poderosa para redução de dimensionalidade. É uma realização particular da abordagem da Fatoração Matricial. A principal questão em uma decomposição de *SVD* é encontrar um espaço de caractere dimensional menor, onde os novos recursos representam os “conceitos” e a força de cada conceito no contexto da coleção é computável. Como o *SVD* permite derivar automaticamente os “conceitos” semânticos em um espaço de baixa dimensão, ele pode ser usado como a base da análise latente-semântica (DEERWESTER et al., 1990), muito popular técnica para classificação de textos em recuperação de informação.

O núcleo do algoritmo *SVD* está no seguinte teorema: é sempre possível decompor uma matriz A em $A = U \lambda V T$. Logo, os dados matriciais $n \times m$ (n itens, m características), podemos obter uma matriz $n \times r$ U (n itens, conceitos r), um $r \times r$ matriz diagonal λ (força de cada conceito), e uma matriz $m \times r$ V (m características, r conceitos). A matriz diagonal λ contém os valores singulares, que serão sempre positivos e classificados em ordem decrescente. A matriz U é interpretada como a matriz de similaridade “item do conceito”, enquanto a matriz V é a matriz de similaridade “termo do conceito”, conforme demonstrado na figura 5.

Figura 5 – Matrix factorization



Fonte: (RICCI; ROKACH; SHAPIRA, 2015)

1.1.2.5 Association rules

A Association rules se concentra principalmente, na área de mineração de dados. Baseia-se em encontrar regras que prevejam a ocorrência de um item baseado nas ocorrências de outros itens em uma transação. O fato de que dois itens são encontrados para serem relacionados significa co-ocorrência, mas não causalidade. Definimos um conjunto de itens como uma coleção de um ou mais itens (por exemplo, leite, cerveja, fralda). Um *k-itemset* é um conjunto de itens que contém *k* itens. A frequência de um dado no conjunto de itens é conhecido como contagem de suporte (por exemplo, (leite, cerveja, fralda) = 131). E o apoio do conjunto de itens é a fração de transações que o contém (por exemplo, (leite, cerveja, fralda) = 0,12). Um conjunto de itens frequente é um conjunto de itens com um suporte maior ou igual a um limite. Uma regra de associação é uma expressão da forma $X \Rightarrow Y$, onde X e Y são conjuntos de itens (por exemplo, leite, fralda \Rightarrow cerveja). Neste caso, o apoio do regra de associação é a fração de transações que contenham X e Y . Por outro lado, a confiança da regra é a frequência com que os itens em Y aparecem em transações que contêm X (RICCI; ROKACH; SHAPIRA, 2015).

1.1.2.6 Gradient boosting

O algoritmo para Boosting Trees evoluiu da aplicação de métodos de *boosting* para árvores de decisão. A ideia geral é calcular uma sequência de árvores muito simples, onde cada árvore sucessiva é construída para os resíduos de previsão da árvore anterior.

Uma abordagem semelhante é construir apenas para uma subamostra selecionada aleatoriamente do conjunto de dados completos, para cada árvore simples consecutiva. Em outras palavras, cada árvore consecutiva é construída para os resíduos de previsão (de todas as árvores precedentes) de uma amostra aleatória independentemente sorteada.

A introdução de certo grau de aleatoriedade na análise pode servir como uma poderosa proteção contra o *overfitting* (uma vez que cada árvore consecutiva é construída para uma amostra diferente de observações) e modelos de rendimento (expansões aditivas ponderadas de árvores simples) que generalizam bem as novas observações, ou seja, apresentam boa validade preditiva. Esta técnica de realização de computações de reforço consecutivas em amostras de observações desenhadas independentemente, é conhecida como reforço de gradiente estocástico[(FRIEDMAN et al., 2000), (STAT-SOFT, 2019)].

1.1.2.7 Métricas de avaliação da aprendizagem de máquina

Para avaliar o desempenho da recomendação, comumente utilizamos os métodos de precisão.

Normalmente, as classificações de um conjunto são divididas em um conjunto de treinamento e usadas por um subconjunto de teste para aprender. Para avaliar a precisão desses treinamentos, utilizamos as medidas populares dentro do campo da Aprendizagem de máquina ou *Machine Learning*:

Como de costume no processo de aprendizado de máquina, dividimos o total de amostras recuperadas em dois conjuntos de dados: teste e validação. Como estimadores de desempenho, selecionamos métricas diferentes para avaliar os pontos fortes e fracos de cada algoritmo (TORRES; SANTOS, 2018). Essas métricas se baseiam em quatro medidas:

- TP - *True Positive* (verdadeiro positivo) - que se refere à quantidade de itens positivos que foram classificadas corretamente;
- TN - *True Negative* (verdadeiro negativo) - que diz respeito à quantidade de itens negativos classificadas corretamente;
- FP - *False Positive* (falso positivo) - que se refere à quantidade de itens negativos que foram classificadas, incorretamente, como positivas;
- FN - *False Negative* (falso negativo) - que diz respeito à quantidade de itens positivos que foram classificadas, incorretamente, como negativas.

As técnicas podem ser utilizadas para avaliar os resultados de testes em ML. Algumas utilizam as métricas de Torres e Santos (2018) e outras não.

Classificação binária

As métricas de classificação binária não nos fornecem uma boa estimativa do desempenho de um modelo. Por exemplo, considere um conjunto de dados de classe binária em que 99% dos dados pertencem a uma classe e apenas 1% dos dados pertence à outra classe. Agora, se um classificador previsse sempre a classe majoritária para cada ponto de dados, ele teria 99% de precisão. Mas isso não significaria que o classificador tenha um

bom desempenho. Para tais casos, utiliza-se as métricas TP, TN, FP e FN. Para ilustrar, considere um teste que tenta determinar se uma pessoa tem câncer. Se o teste prevê que uma pessoa tem câncer, quando na verdade não tem, é um falso positivo. Por outro lado, se o teste falhar em detectar o câncer em uma pessoa que realmente está sofrendo, é um falso negativo.

Acurácia

A Acurácia é a métrica para medir a precisão mais usada para avaliar o desempenho de um modelo de classificação. É a razão entre o número de previsões corretas e o número total de previsões feitas pelo modelo.

$$\text{Acurácia} = \frac{\text{Número de previsões corretas}}{\text{Número total de previsões}} \quad (1.2)$$

MSE

Na configuração de regressão, a medida mais usada é o Mean Squared Error (MSE). O MSE será pequeno se as respostas previstas estiverem muito próximas das respostas verdadeiras, e será grande se, para algumas das observações, as respostas previstas e verdadeiras diferem substancialmente. O MSE é calculado usando os dados de treinamento que foram usados para ajustar o modelo representado pela fórmula matemática:

$$MSE = \frac{1}{n} \sum_{t=1}^n e_t^2 \quad (1.3)$$

RMSE

O Root Mean Square Error (RMSE) é uma métrica amplamente usada para avaliar o desempenho dos regressores. Matematicamente, é representado da seguinte forma:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2} \quad (1.4)$$

MAE

O Mean Absolute Error (MAE) técnica de avaliação popular para o modelo de mineração de dados. Essa métrica de avaliação é muito semelhante ao RMSE, e é calculada como erro médio entre valores previstos e reais e é dada pela seguinte equação:

$$MAE = \frac{1}{n} \sum_{t=1}^n |e_t| \quad (1.5)$$

Recall

O recall é a proporção do número de casos positivos, que foram identificados para todos os casos positivos presentes no conjunto de dados.

$$Recall = \frac{TP}{TP + FN} \quad (1.6)$$

Essas técnicas são utilizadas em vários tipos de aplicações, com destaque, nos Sistemas de Recomendação.

1.2 Sistemas de Recomendação

Os sistemas de recomendação surgiram como uma área de pesquisa independente em meados dos anos 90. Atualmente o interesse em recomendar produtos, ou serviços, aumentou dramaticamente em quase todos os negócios da Internet. Fornecer boas recomendações, sejam amigos, filmes ou mantimentos, ajudam muito na definição da experiência do usuário e na sedução de seus clientes para usar e comprar de sua plataforma.

Podemos ter uma pequena definição para sistemas de recomendação como ferramentas e técnicas de software, que fornecem sugestões para itens serem úteis para um usuário. As sugestões estão relacionadas aos vários processos de tomada de decisão, como quais itens comprar, quais músicas ouvir, ou quais notícias *on-line* ler. "Item" é o termo geral usado para denotar o que o sistema recomenda aos usuários. Os sistemas recomendadores provaram nos últimos anos ser um meio valioso para lidar com o problema da sobrecarga de informações, considerando as inúmeras diversidades disponíveis em um sistema *on-line*. Esses tipo de sistemas, usam os problemas de aprendizado de máquina com técnicas específicas, sendo que, os modelos selecionados dependem muito da quantidade e da qualidade dos dados (RICCI; ROKACH; SHAPIRA, 2015).

Os sistemas de recomendação são um tipo especial de sistemas de filtragem de informa-

ções. A filtragem de informações lida com a entrega de itens selecionados de uma grande coleção, que o usuário provavelmente achará interessante ou útil, e pode ser visto como uma tarefa de classificação e tomada de decisões no nosso dia-a-dia, como a compra mais comum de produtos *online*, sugestões de amigos sobre aplicativos sociais, recomendações de vídeos, músicas e notícias (PAZZANI; BILLSUS, 2007).

Gorakala (2016) afirmou que outras áreas, por exemplo, viagens, bancos e áreas de saúde, também têm tido implementações bem-sucedidas de sistemas de recomendação. Com base nos dados de treinamento, um modelo de usuário é induzido, permitindo que o sistema realize de filtragens para classificar itens em uma classe positiva (relevante para o usuário), ou em uma classe negativa (irrelevante para o usuário) (METEREN; SOMEREN, 2000). Em termos técnicos, os sistemas de recomendação envolvem duas áreas principais: IA e técnicas de *Data Mining*. Esse conjunto de ferramentas para analisar grandes volumes de dados e fornecer sugestões relevantes pode ser descrito pelo desenvolvimento de um modelo matemático, ou função objetiva, que pode prever quanto um usuário irá gostar de um item da seguinte maneira:

$$\text{Se } U = \{\text{usuários}\},$$

$$I = \{\text{itens}\} \text{ então,}$$

$F =$ O objetivo da função é medir a utilidade do item U , dado por: $F: U \times I \rightarrow R$

$$\text{Onde } R = \{\text{itens recomendados}\}$$

Segundo Sielis, Tzanavari e Papadopoulos (2015) um sistema de recomendação consiste em procedimentos de funcionamento cíclico divididos nas seguintes etapas: coleta de dados, filtragem de dados, classificação dos itens recomendados e apresentação de dados conforme ilustrado na figura 6. Que é composta pelos seguintes itens:

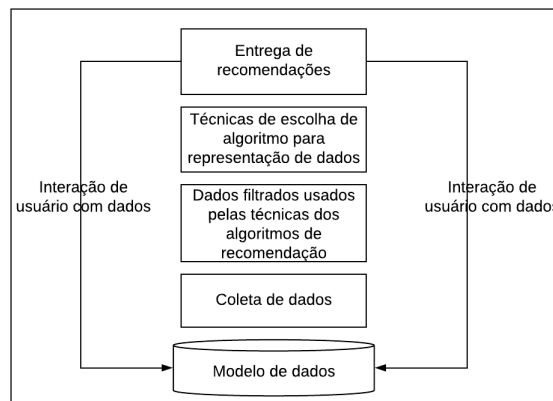
- Coleta de dados - A coleta de dados está diretamente correlacionada ao modelo de dados usado em um aplicação de software. Os dados são geralmente definidos com

base no design geral de um aplicativo de software e com base nas informações contextuais que um aplicativo de software coleta e processa. O modelo de dados geralmente depende do domínio para o qual um sistema de recomendação é construído e dos métodos de armazenamento e representação dos dados, como por exemplo: bancos de dados relacionais, bancos de dados não relacionais e banco de dados de séries temporais (time series database).

- Dados filtrados usados pelas técnicas dos algoritmos de recomendação - As técnicas de filtragem de recomendação dependem do tipo de dados que um RS está processando e do tipo de recomendações que planeja produzir. As técnicas de filtragem de recomendações de acordo com o tipo de dados será usado para o cálculo de recomendações, bem como os métodos de algoritmo computacional que eles usam.
- Técnicas de escolha de algoritmo para representação de dados - A coleta de dados e sua análise através das técnicas de filtragem de recomendações, gera um conjunto de recomendações, que cumprem as regras que um sistema de recomendação deve levar em consideração durante os cálculos. O próximo e último passo que um RS deve cumprir é apresentar o conjunto de recomendações gerado para os destinatários finais, os usuários.
- As recomendações devem ser apresentadas aos usuários classificados. A classificação das recomendações é baseada nas preferências de cada usuário e em seus interesses pessoais. Uma recomendação é considerada bem-sucedida quando a recomendação de prioridade mais alta oferecida a um usuário estiver mais próxima de seus interesses e quando essa recomendação é realmente aceita pelo usuário.

Portanto, o principal objetivo de um sistema de recomendação é fornecer sugestões de itens relevantes para os usuários, a fim de aproveitarem as melhores alternativas disponíveis. Uma boa recomendação é a personalização, levando em conta as informações do perfil do usuário digital, como registros de interações e informações sobre produtos, especificações, detalhes de *feedback* de pesquisa de usuários em transações de compra e venda, as informações demográficas e etc. Os dados históricos para fazerem essas recomendações precisam ser de fontes confiáveis. Devido a essa complexidade, a construção desses mecanismos deve levar em conta o investimento em habilidades e tecnologias apropriadas. Empresas que utilizam sistemas de recomendação para o ramo de seus negócios

Figura 6 – Arquitetura funcional dos Sistemas de Recomendação



Fonte:(SIELIS; TZANAVARI; PAPADOPOULOS, 2015)

relataram em suas experiências que: dois terços dos filmes assistidos pelos clientes da Netflix são filmes recomendados; 38 % das taxas de cliques do Google Notícias são links recomendados; 35 % das vendas da Amazon vêm de produtos recomendados; a ChoiceStream disse que 28 % das pessoas gostariam de comprar mais músicas, se eles gostarem das sugestões (GORAKALA, 2016).

1.2.1 Técnicas de recomendação

Visando identificar técnicas de recomendação, foi realizado um estudo que identificou três propostas: filtragem colaborativa, sistemas baseados em conteúdo e os sistemas híbridos, que adaptam os dois conceitos(BANIK, 2018).

1.2.1.1 Filtragem colaborativa

A filtragem colaborativa prevê um valor para cada item por meio de classificações explícitas, opiniões de usuários e de pessoas de fora do sistema. O uso da comunidade para fornecer recomendações torna os filtros colaborativos mais precisos. Esses algoritmos têm mais recursos computacionais, utilizando milhões de usuários para fornecerem recomendações de produtos em tempo real. Popularmente na indústria, esses algoritmos são encontrados em consultores de empresas como Amazon e Netflix. Este modelo também permite um treinamento atualizado todos os dias. Um dos principais pré-requisitos de um sistema de filtragem colaborativa é a disponibilidade de dados das atividades passadas. A empresa Amazon, por exemplo, aproveita os filtros colaborativos porque tem acesso aos

dados de compras de milhões de usuários.







Portanto, os filtros colaborativos sofrem com o chamado problema de partida à frio (BANIK, 2018), ou seja, significa que o recomendador não leva em consideração as preferências de um usuário individual, quando o seu acesso é realizado pela primeira vez.

Em 2003, Linden, Smith e York, da Amazon.com, Ricci, Rokach e Shapira (2015) publicaram um artigo intitulado *Item-to-Item Collaborative Filtering*, que explicava como as recomendações de produtos na Amazon funcionam. Desde então, essa classe de algoritmo passou a dominar o padrão da indústria para recomendações (RICCI; ROKACH; SHAPIRA, 2015).

A figura 7 pode ser considerada como uma matriz que tem seis usuários avaliando seis itens. Portanto, $m = 6$ e $n = 6$. O usuário 1 deu ao item 1 uma classificação de 4, portanto, $r_{i1} = 4$. Em um repositório de 20.000 produtos e 5.000 usuários, teríamos uma matriz de classificação de forma 5.000 X 20.000.

No entanto, todos os seus usuários consumirão apenas uma fração dos produtos disponíveis em seu site. Portanto, essa matriz é esparsa. Em outras palavras, a maior parte das entradas na matriz está vazia, já que a maioria dos usuários não classificou os seus produtos em sua totalidade.

Figura 7 – Recomendação baseada em filtro colaborativo

| | i1 | i2 | i3 | i4 | i5 | i6 |
|---|-----------|-----------|-----------|-----------|-----------|-----------|
|  01 | 4 | ? | 3 | ? | 5 | ? |
|  02 | ? | 2 | ? | ? | 4 | 1 |
|  03 | ? | ? | 1 | ? | 2 | 5 |
|  04 | ? | ? | 3 | ? | ? | 1 |
|  05 | 1 | 4 | ? | ? | 2 | 5 |
|  06 | 5 | ? | 2 | 1 | ? | 5 |

Fonte: (BANIK, 2018)

O problema de previsão, portanto, visa prever esses valores omissos usando todas as informações de que dispõe (as classificações registradas, os dados de produtos, os dados dos usuários e assim por diante). Se for capaz de prever os valores em falta com precisão, será capaz de dar grande recomendações. Por exemplo, se um usuário i não usou o item

j , mas o nosso sistema prevê uma classificação muito alta (denotada por ij), é altamente provável que um outro usuário nesta condição goste de j .

1.2.1.2 Sistemas baseados em conteúdo

Os sistemas baseados em conteúdo recomendam um item a um usuário considerando uma descrição de item e um perfil de interesses do usuário, ao contrário dos sistemas baseados em filtragem colaborativa, eles não aproveitam o poder da comunidade e geralmente, fornecem recomendações óbvias (PAZZANI; BILLSUS, 2007).

Esses modelos computacionais constroem algoritmos baseados na similaridade entre pares de corpos de texto (KIM et al., 2019). A figura 8 representa um sistema de recomendação baseado em conteúdo, onde um usuário realiza requerimento de um item, e o recomendador irá selecionar apenas os itens similares para sugerir. Neste caso, o sistema recomendador irá selecionar os itens que encontram-se no mesmo grupo do item desejado pelo consumidor para recomendá-los. Sistemas baseados em conteúdo recomendam itens semelhantes àqueles que um determinado usuário gostou no passado:

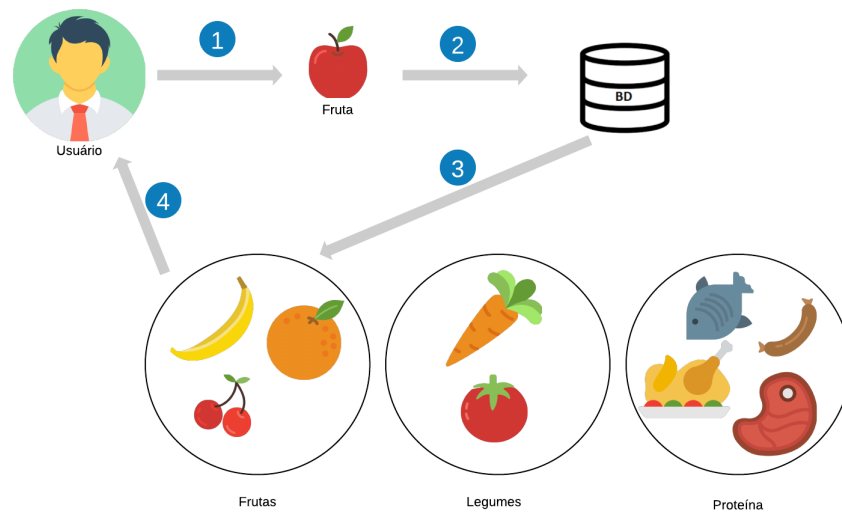
1. O usuário seleciona um item em uma lista;
2. O sistema recomendador procura na base de dados os produtos similares com o item selecionado pelo usuário;
3. O banco de dados retorna apenas o agrupamento dos itens semelhantes para o sistema; e
4. O recomendador sugere para o usuário os itens relacionados com os que ele selecionou.

Dentre as técnicas de sistemas de recomendação baseados em conteúdos, destaca-se a aplicação de técnicas de identificação de similaridade para texto que são descritas a seguir.

1.2.1.3 Sistemas híbridos

Os sistemas híbridos combinam os dois métodos para melhorar o desempenho e atender algumas necessidades, como o problema de partida a frio, que acontece nos sistemas baseados em filtragem colaborativa (BURKE, 2002). A ideia por trás de técnicas híbridas é

Figura 8 – Recomendação baseada em conteúdo



Recomendação de produtos similares.

Fonte: O autor, 2019

fazer uma combinação de algoritmos que fornecerá recomendações mais precisas e efetivas do que um único algoritmo, já que as desvantagens de um algoritmo podem ser superadas por outro algoritmo (SCHAFER et al., 2007). A combinação de abordagens pode ser feita de qualquer uma das seguintes maneiras: implementação separada de algoritmos e combinação do resultado, utilizando alguma filtragem baseada em conteúdo (ISINKAYE; FOLAJIMI; OJOKOH, 2015).

1.3 Técnicas de identificação de Similaridades entre textos

Essencialmente, os modelos baseados em conteúdo, voltados para texto, computam a similaridade entre pares de corpos de texto. Mas, é essencial quantificar numericamente a similaridade entre dois corpos de texto para fundamentar os resultados.

Neste caso, representamos os corpos de texto (daqui em diante referidos como documentos) como quantidades matemáticas. Isso é feito representando esses documentos como vetores, que contêm uma série de n números, onde cada número representa a dimensão e n é o tamanho do vocabulário dos documentos.

Os valores desses vetores dependem da técnica usada para converter os documentos em vetores. Os dois vetorizadores mais populares são o *CountVectorizer* e o *TF-IDFVectorizer*, que são brevemente descritos a seguir.

1.3.1 TF-IDFVectorizer

Em geral, os sistemas baseados em conteúdo suportam a modelagem vetorial baseada em vetores TF-IDF (term frequency–inverse document frequency) / VSM (Vector Space Model). O *VSM* é uma representação espacial de documentos de texto. Nesse contexto, cada documento é representado por um vetor em um espaço n-dimensional, onde cada tempo corresponde a um termo geral de vocabulário de uma coleção de documentos. É o mais amplamente utilizado e considerado como um dos esquemas de ponderação de termo mais apropriados. Esse *TF-IDF* é empregado para se livrar de termos com pesos menores de documentos e ajuda a aumentar a eficácia da recuperação. O *TF-IDF* é uma estatística numérica, que nos diz a importância de uma palavra para um documento em uma coleção ou corpus. É usado principalmente como um fator de ponderação em vários processos usados para recuperação de informações e mineração de texto. O aumento no valor *TF-IDF* de uma palavra é diretamente proporcional ao número de vezes que a palavra ocorre no documento, mas é neutralizado pela frequência da palavra no *corpus*, o que ajuda a equilibrar as palavras que aparecem mais frequentemente em geral.

$$\text{TF-IDF} = (\text{Term Frequency} * \text{Inverse Document Frequency}) \quad (1.7)$$

TF - mede quantas vezes um termo ocorre em um documento. Como os documentos têm diferentes comprimentos, pode acontecer de o documento mais longo conter um termo mais vezes do que os documentos que são mais curtos. Assim, para normalizá-lo, o termo é calculado da seguinte maneira (1.8):

$$\text{TF} = \text{Total number of items in a document} / \text{Number of times a term appears in a document} \quad (1.8)$$

IDF - ajuda a determinar a importância de um termo. Quando calculamos a frequência do termo, damos igual importância a todos os termos. Mas certos termos, como “o”, “aquilo” e “é”, podem aparecer com muita frequência, pois não são importantes. Então, precisamos reduzir os pesos de termos frequentes e aumentar os pesos dos termos raros, calculando o seguinte:

$$\text{IDF} = \log_2 (\text{Number of document with term } t \text{ in them}) / \text{Total number of documents} \quad (1.9)$$

1.3.2 CountVectorizer

A vetorização é o processo geral de transformar um texto em vetores de características numéricas. Essa estratégia específica (*tokenization, stop words, parsing, counting*) (KUMARI; JAIN; BHATIA, 2016) é representado por um conjunto chamado de *Bag of Words*. Os documentos são descritos por ocorrências de palavras enquanto ignoram completamente as informações de posição relativa das palavras no documento.

Tokenization é um método de separar um texto em palavras, frases, símbolos ou outros elementos significativos chamados *tokens*. A lista de *tokens* se torna entrada para outros processos, como análise ou mineração de texto. Geralmente, a *tokenização* ocorre no nível da palavra. A pontuação e o espaço em branco geralmente não são incluídos na lista final de *tokens*. Um *token* consiste em todas as sequências contíguas de caracteres ou números alfabéticos. Os tokenizadores são usados para dividir uma *string* em um fluxo de termos ou *tokens*. Um tokenizador simples pode dividir um fluxo de texto em *tokens* nos locais em que encontrar um espaço em branco ou pontuação.

Stop words é o processo de remover palavras comuns, como "se", "do que", "ou", "em", "e", "o". Isso ajuda a aumentar a eficiência e a eficácia no processo de recuperação de informações. Algumas palavras comuns que são muito importantes na seleção de documentos de acordo com a necessidade do usuário são removidas. Essas palavras são chamadas de palavras de parada. As palavras de destaque geralmente são determinadas classificando os termos por sua frequência na coleção de documentos e, em seguida, os termos mais frequentes são usados como palavras de parada, geralmente são feitas exceções para as palavras semanticamente relacionadas ao domínio dos documentos em consideração. As palavras de parada da lista de parada não são incluídas para os processos futuros, como a indexação, etc.

Parsing processa a sequência de *tokens* de texto no documento para reconhecer os elementos estruturais. É o método de analisar uma sequência de símbolos em uma linguagem, obedecendo às regras de uma gramática formal. Em computação geral, o termo

análise significa a análise formal de uma sentença, ou outra sequência de palavras em seus componentes por um computador, resultando em uma árvore de análise, que mostra a estrutura na qual eles estão relacionados entre si. A análise é complementar à modelagem, que produz saída formatada, por exemplo, títulos, *links*, cabeçalhos etc.

O *CountVectorizer* é um tipo de vetorizador e o seu processo é melhor explicado com a ajuda de um exemplo (BANIK, 2018). Imagine que temos três documentos, A, B e C, que são os seguintes:

A: Marte é descrito como o "Planeta Vermelho".

B: O meu carro é azul, azul da cor do mar.

C: Minha terra tem palmeiras.

Agora temos que converter esses documentos em seus formulários vetoriais usando o *CountVectorizer*. O primeiro passo é calcular o tamanho do vocabulário. O vocabulário é o número de palavras únicas presentes em todos os documentos. Portanto, o vocabulário para esse conjunto de três documentos é o seguinte: marte, é, descrito, como, o, planeta, vermelho, meu, carro, azul, da, cor, do, mar, minha, terra, tem, palmeiras. Conseqüentemente, o tamanho do vocabulário é 18.

É uma prática habitual não incluir *stop words*: a, o, é, como, meu, da, etc, no vocabulário. Portanto, eliminando as palavras de parada, nosso vocabulário V , é o seguinte:

V : azul, carro, cor, mar, marte, palmeiras, planeta, terra, vermelho

O tamanho do nosso vocabulário agora é nove. Portanto, nossos documentos serão representados como vetores unidimensionais e cada dimensão aqui representará o número de vezes que uma palavra específica ocorre em um documento.

Portanto, usando a abordagem do *CountVectorizer*, A, B e C agora serão representados da seguinte maneira:

A: (0, 0, 0, 0, 1, 0, 1, 0, 1)

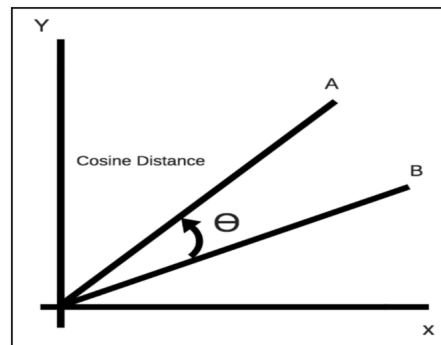
B: (2, 1, 1, 1, 0, 0, 0, 0, 0)

C: (0, 0, 0, 0, 0, 1, 0, 0, 1)

1.3.3 Cosine similarity

Cosine similarity (similaridade do *cos seno*) é uma medida de similaridade entre dois vetores não nulo de um espaço de produto interno, que mede o *cos seno do ângulo* entre dois vetores e determina se eles estão apontando aproximadamente na mesma direção. É frequentemente usado para medir a similaridade de documentos na análise de texto. A pontuação do *cos seno* pode assumir qualquer valor entre -1 e 1. Quanto mais próximos a 1, a pontuação do *cos seno*, mais semelhantes são os produtos entre si. A figura 9 representa o cálculo da pontuação do *cos seno* do ângulo entre dois vetores em um espaço n-dimensional.

Figura 9 – Medida de similaridade entre dois vetores.



Fonte: (BANIK, 2018)

Os vetores de frequência são tipicamente muito longos e esparsos (ou seja, possuem muitos valores 0). Os aplicativos que usam essas estruturas incluem recuperação de informações, agrupamento de documentos de texto, taxonomia biológica e mapeamento de recursos genéticos (COSINE-SIMILARITY, 2019).

Seja x e y dois vetores para comparação. Usando a medida de *cos seno* como uma função de similaridade, temos:

$$\text{similarity}(\mathbf{x}, \mathbf{y}) = \cos \theta = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \quad (1.10)$$

onde $\|\mathbf{x}\|$ é a norma euclidiana do vetor $\vec{x} = (x_1, x_2, x_p, \dots)$, definido como

$$\sqrt{(x_1^2, x_2^2, x_p^2, \dots)}$$

Conceitualmente, é o comprimento do vetor. Da mesma forma, $\|\mathbf{y}\|$ é a norma euclidi-

ana do vetor y . A medida calcula o cosseno do ângulo entre os vetores x e y . Desse modo, um valor de cosseno igual a 0 significa que os dois vetores estão 90 graus um com o outro (ortogonais) e não têm correspondência. Quando a pontuação do *cosseno* é 1 (ou o ângulo é 0), os vetores são exatamente semelhantes. Por outro lado, uma pontuação de *cosseno* de -1 (ou ângulo de 180 graus) indica que os dois vetores são exatamente diferentes um do outro.

1.4 Comentários Finais

Nesse capítulo foram apresentados conceitos relacionados a Inteligência Artificial e aprendizado de máquina ou ML, com o intuito de fornecer embasamento teórico para o desenvolvimento desta dissertação.

Os Tipos de sistemas de recomendação foram contextualizado brevemente, com exemplificação para cada tipo de abordagem, que são frequentemente referenciadas na literatura. Os algoritmos de ML voltado para sistemas de recomendação, foram também apresentados visando esclarecer como funciona a formação de uma base de dados, assim como, o treinamento e teste do conjunto de dados. O sistemas de recomendação baseado em conteúdo foi descrito, com ênfase nos métodos: *CountVectorizer* e o *TF-IDFVectorizer* para fazer a criação de um vetor de palavras utilizando o conceito de vetorização. Estas técnicas serão utilizadas para o desenvolvimento da proposta desde trabalho.

2 TRABALHOS CORRELATOS

Neste capítulo, são descritos e analisados alguns aplicativos e artigos, que apoiam o processo de compra de produtos em supermercados.

2.1 Aplicativos

Visando encontrar aplicativos em plataformas móveis que apoiassem compras, com diferentes tipos de tecnologia e utilização de recomendações, além de sistemas que tivessem relação com supermercados, foi realizada uma busca na loja de aplicativos Android, iOS (iphone) e Windows phone utilizando as palavras-chaves: "lista"+"de"+"compras" e "supermercado". Foram encontrados 272 resultados de aplicativos e após análise dos objetivos de cada sistema, consideramos 18 aplicativos, dos quais estão relacionados com compras de supermercados, cujas funcionalidades estão listadas na tabela 1.

Tabela 1 – Aplicativos móveis para compra de produtos de supermercado

| | Nome | Descrição |
|-------|----------------------|---|
| i | AnyList Grocery List | Cria lista de compras, receitas, compartilha com usuários. |
| ii | Boa Lista | Lê código de barras dos produtos e compara preços de lojas. |
| iii | Buscapé Mobile | Compara preços dos produtos utilizando a câmera. |
| iv | Buy Me a Pie | Cria lista de compras e compartilha com usuários. |
| v | Clube Extra | Cria lista de compras, programa de relacionamento. |
| vi | Desrotulando | Recomenda o produto mais saudável, disponível para ios. |
| vii | Facilista | Apresenta o valor do produto e onde encontrá-lo. |
| viii | Home Refill | Cria lista de compras por perfil cadastrado. |
| ix | iList Touch | Cria lista de compras com preços, categoria e fotos. |
| x | Listonic | Cria lista de compras, compartilha e classifica produtos. |
| xi | Mercadapp | Compras de produtos online. |
| xii | Meu Carrinho | Cria lista de compras e compara preço de lojas. |
| xiii | Pão de Açúcar Mais | Cria lista de compras, programa de relacionamento. |
| xiv | QQFalta | Cria lista de compras e compartilha com usuários. |
| xv | Soft List | Cria lista de supermercado consulta valor de produtos. |
| xvi | Shopping List | Cria lista de supermercado com o valor total em reais. |
| xvii | ShopWell | Recomenda o produto mais saudável (indisponível no Brasil). |
| xviii | Shopper | Sistema de reabastecimento de itens de consumo doméstico. |

Fonte: O autor, 2019

A seguir, os produtos da tabela 1 são descritos com mais detalhes:

- (i) AnyList é um aplicativo disponível somente para o sistema iOS (iphone) que promete criar listas organizadas rapidamente. Dentre suas funcionalidades, compreende sugerir itens comuns, enquanto o usuário digita e agrupa automaticamente itens por categoria para ajudar a economizar tempo na loja. Sincroniza com a família e os amigos compartilhando uma lista, atualizando qualquer alterações instantaneamente feitas por eles em uma lista compartilhada. Permite adicionar itens via comando de voz utilizando a Siri (A Siri é uma assistente inteligente que ajuda a deixar tudo mais rápido e fácil nos seus aparelhos Apple. Mesmo antes de você pedir (SIRI,

2019)). E sugere itens de mercearia comuns enquanto o usuário digita, e esses itens são automaticamente agrupados em categorias como laticínios, produtos e carne (ANY-LIST, 2019).

- (ii) Boa Lista oferece o serviço de acesso à câmera do aparelho de celular para ler códigos de barras e consultar os preços enquanto faz as compras. Assim, possibilita saber a cada minuto quanto está gastando no supermercado.
- (iii) O Buscapé é um comparador de preços de produtos, disponível para os sistemas Android e iOS. Proporciona consultar antes de todas as compras, pois compara os preços, lojas e produtos (BUSCAPE, 2019).
- (iv) O Buy Me a Pie é um aplicativo de lista de supermercado, que facilita sincronizar com a família ou amigos, compartilhando a conta do usuário. Podendo adicionar ou alterar os itens em qualquer lugar usando um dispositivo móvel com um sistema Android ou iOS (BUY-ME-A-PIE, 2019).
- (v) O Clube Extra é um aplicativo disponível para Android e iOS (iphone), permite que um usuário crie listas de compras de supermercado no local de sua preferência, ainda oferece vantagens para quem é clientes (CLUBE-EXTRA, 2019).
- (vi) O Desrotulando é o primeiro aplicativo de *food score* do Brasil feito por nutricionistas, que está disponível para a plataforma Android e iOS. As informações importantes do rótulo dos produtos são traduzidas em uma nota, de 0 a 100. Possibilita consultar os produtos pelo código de barras, por nome, marca ou categoria (DESROTULANDO, 2019).
- (vii) Facilista é disponibilizado gratuitamente para sistemas Android e iOS, mas se encontra em versão de testes. É uma plataforma de comparação de preços de produtos de supermercados por geolocalização e permite que os consumidores verifiquem, em tempo real, as ofertas mais próximas a eles. Além disso, é possível criar, editar e compartilhar a lista (FACILISTA, 2019).
- (viii) HomeRefill disponível para os sistemas Android, permite aprender com a lista de compras o que é essencial para o consumidor, para evitar filas, perda de tempo e o desperdício de dinheiro e produtos. O sistema controla os gastos da despensa do

usuário por departamento e categoria, também permite determinar o período de refill para a despensa (HOME-REFILL, 2019).

- (ix) O iList Touch é um aplicativo para organizar as compras semanais e ainda, manter o controle financeiro sobre as despesas. É compatível com o ios (iPhone) e proporciona a criação de uma lista de compras personalizável, conforme as necessidades de um usuário (ILISTOUCH, 2019).
- (x) Listonic é um aplicativo para Android, que cria uma lista de compras de supermercado, compartilha com outros usuários, acrescenta itens com comando de voz, classifica produtos por categoria e calcula o valor da lista (LISTONIC, 2019).
- (xi) Mercadapp está disponível para Android e iOS, para criar uma lista de compras em um supermercado de preferência de um usuário, possibilitando selecionar a forma de pagamento e agendar o horário de entrega (ILISTOUCH, 2019).
- (xii) Meu Carrinho é um aplicativo para dispositivos móveis Android e iOS que permite, criar listas de compras, comparar os preços dos produtos nos supermercados da região selecionada por um usuário (MEU-CARRINHO, 2019).
- (xiii) Pão de Açúcar Mais é um aplicativo disponível para Android e iOS (iphone), permite que um usuário crie listas de compras nas unidades de sua preferência, ainda oferece benefícios para quem é cliente (PAODEACUCAR, 2019).
- (xiv) QQFalta é serviço interativo que proporciona aos usuários a facilidade de criar suas listas de compras de supermercados acessando diretamente do site ou do dispositivo móvel com o sistema Android (QQFALTA, 2019).
- (xv) O aplicativo SoftList, disponível para a plataforma android, oferece criar uma lista de compras de forma rápida utilizando o catálogo de produtos e com o auxílio do recurso autocompletar. Adicionando os nomes dos produtos ou listas completas, adicionando preço, unidade de medida, categoria, observação e foto, calcula o total e salva o histórico da compra, caso o usuário tenha comprado o mesmo produto em lojas diferentes. Também realiza a comparação dos preços, e compartilhar listas com outros usuários por e-mail, SMS e WhatsApp, e mantém todos os seus dados salvos na nuvem através de backups automáticos (SOFT-LIST, 2019).

- (xvi) O ShoppingList está disponível para Android e permite criar lista de compras de supermercado preenchendo itens de um carrinho de compras através da leitura de código de barras ou usando a voz (SHOPPING-LIST, 2019).
- (xvii) O ShopWell fornece pontuações nutricionais personalizadas de produtos, que varia de 1 a 100, através de digitalização do código de barras de qualquer item para criar listas com determinado perfil alimentar, idade e sexo. Os usuários conseguem certificar quais produtos e marcas são mais saudáveis. O aplicativo está disponível para Android e iOS (SHOPWELL, 2019).
- (xviii) Shopper é um sistema web de reabastecimento de itens de consumo doméstico: produtos de limpeza, higiene pessoal e alimentos não perecíveis. O acesso pode ser feito de qualquer dispositivo com um navegador e internet. Criada por ex-alunos Insper e da USP, que se basearam em um modelo similar nos Estados Unidos e decidiram implantar no Brasil, para melhorar a qualidade de vida das pessoas. O usuário cria uma lista de compras e ao finalizar solicita a entrega, o serviço garante economia por não possuir loja física, ainda lembra o cliente de suas compras através do envio de sms (mensagem para celular) (SHOPPER, 2019).

2.2 Artigos

Tendo em vista encontrar artigos que respaldassem a utilização de aplicativos por usuários de supermercado ao criarem seus carrinhos de compras, foi realizada uma busca no World Wide Science e no Researchgate com o range de data iniciando em 2017 até o ano presente da publicação deste trabalho. As palavras-chave utilizadas na busca foram: "grocery shopping product recommendation". Foram encontrados 188 artigos. Dentre os artigos encontrados, verificou-se a relação com outras ferramentas, tecnologias ou metodologias, que foram consideradas as palavras-chave complementares: "system grocery shopping product recommendation architecture in real-time", ao todo com 50 alternativas. Ponderamos 3 artigos, e listamos na tabela 2 os comparativos do uso de tecnologia com os aplicativos comercializados especificados na tabela 1.

Aiolfi (2019) considera que os pesquisadores realizaram poucos estudos sobre a adoção de aplicativos móveis para compras de supermercado e corroborando com o que foi proposto neste projeto. O estudo apresentou o desenvolvimento de uma nova versão do

modelo de *smart mobile* no mercado italiano de aplicativo móvel inteligente para compras de supermercado. A implementação inclui variáveis relacionadas ao fenômeno do uso do dispositivo móvel, fora da loja, como meio de preparação das compras, e na loja, como uma ferramenta de auto-regulação. Essa pesquisa possibilitaria ajudar pesquisadores e gerentes a entender melhor comportamento dos compradores no novo cenário do varejo. O estudo sobre a adoção de aplicativos móveis para compras de supermercado foi baseado no modelo TAM de (DAVIS, 1989), (DAVIS; BAGOZZI; WARSHAW, 1989). O modelo de usabilidade TAM é um sólido ponto de partida teórico e metodológico para prever o comportamento de usuários potenciais e reais e sua atitude em relação a uma tecnologia. Foram mensuradas todas as variáveis consideradas no modelo com escalas de itens múltiplos, com escala de medida Likert. Especificamente, todas as escalas usadas na pesquisa são de pesquisas anteriores sobre compradores e, uma vez traduzidas para italiano, foram adaptadas para o modelo ao contexto de compras de supermercado e medidos de 1 (discordo) a 7 (aceita).

De acordo com Mantha et al. (2019), a crescente adoção pelo consumidor, de compras *on-line* usando plataformas como Amazon Fresh, Instacart e Walmart mercado, ressaltou a necessidade de recomendações relevantes para os clientes. O estudo introduziu uma recomendação de produção dentro do cesto de compras utilizando Real-Time Triple2Vec (RTT2Vec), (FIONDA; PIRRÓ, 2019). Os resultados deste estudo proposto confirmou a validade do modelo Technology Acceptance Model (TAM) para medir os fatores subjacentes à adoção de um dispositivo móvel para supermercados na Itália. Considerando a crescente penetração de dispositivos móveis e a ampla conectividade que eles oferecem, os indivíduos estão se tornando cada vez mais dependentes da tecnologia móvel para realizar atividades do dia-a-dia. Um consumidor mais planejado e bem preparado deve incentivar os varejistas a explorarem a dependência dos consumidores e influenciar seu comportamento durante todo o processo de compras. Portanto, cabe aos varejistas desenvolver uma solução adaptada à necessidade de seus consumidores.

O principal objetivo do estudo foi apresentar o Triple2Vec, uma arquitetura de inferência em tempo real, para servir recomendações dentro da cesta de compras. Foram realizados exaustivos experimentos *offline* em dois conjuntos de dados de compras de supermercado e forneceram as seguintes contribuições: a introdução de um método de inferência aproximado, que transforma a fase de inferência de uma recomendação dentro da

cesta sistema em uma recuperação de incorporação de Approximate Nearest Neighbour, ou Vizinho Mais Próximo Aproximado (ANN), referido na seção 1.1.2; e um sistema de recomendação em tempo real de produção, que atende milhões de clientes *on-line*, mantendo alto rendimento, baixa latência e baixos requisitos de memória.

O sistema proposto é composto de componentes *offline* e *online*. O sistema *online* consiste em um cache distribuído armazenado em cache, um sistema de *streaming*, um mecanismo de inferência em tempo real e um cliente *front-end*. O sistema *offline* abrange um repositório de dados, um repositório de recursos que atende a todos os mecanismos de recomendação do Walmart, respeitando o modelo que emprega (usuário u , item i , item j) triplica, denotando dois itens (i, j) comprados pelo usuário u na mesma cesta, e aprende a representação u para o usuário u e um conjunto duplo de incorporações (p_i, q_j) para o par de itens (i, j) , e uma estrutura de treinamento de modelo *offline* implantado em um *cluster* de GPUs. A avaliação experimental foi realizada em um conjunto de dados público e um conjunto de dados proprietário. Ambos os conjuntos de dados são divididos em treinamento, validação, e conjuntos de teste. O conjunto de dados público Instacart já está dividido em anterior, conjuntos de treino e teste. Para o conjunto de dados do Walmart Grocery, os conjuntos de testes, validação e treino compreendem um ano, os próximos 15 dias e o próximo mês de transações, respectivamente. O Walmart Grocery, proporcionou um grande volume de clientes e interações, fluindo em várias velocidades. O objetivo principal de implantar um mecanismo de inferência em tempo real é para fornecer recomendações personalizadas, garantindo um alto rendimento e fornecendo uma experiência de baixa latência ao cliente. O mecanismo de inferência em tempo real utiliza ANN, construído a partir de incorporações treinadas, e implantado como um *microserviço*. Esse mecanismo interage com o cliente *front-end* para obter o contexto do usuário e da cesta e gera recomendações personalizadas dentro da cesta em tempo real. Foi avaliado o desempenho preditivo do modelo e a latência do sistema. Para cada cesta no conjunto de teste, Foi utilizado 80% dos itens como entrada e os 20% restantes como itens os itens relevantes a serem previstos. Os sistema supera todos os outros modelos nos dois conjuntos de dados melhorando o Recall e o NDCG em 9,37% (5,75%) e 21,5% (9,01%) para os conjuntos de dados Instacart (Walmart) quando comparado ao modelo atual de ponta *triple2vec*. Sendo assim, viabilizou servir recomendações personalizadas de itens em larga escala, com latência ao executar inferência exata em tempo real, resultando em

um aumento no tamanho médio da cesta, permitindo o *check-out* mais rápido do usuário.

Villegas e Saito (2017) desenvolveram um protótipo de sistema para aplicativo móvel destinado a diminuir o fardo das compras de supermercado dos usuários idosos, sabendo que segundo as Nações Unidas, entre 2015 e 2030, o número de pessoas no mundo com 60 anos, ou mais, deve crescer 56%. O sistema recomenda produtos de acordo com os que estão atualmente em um carrinho de compras e apoia os usuários para encontrar o caminho da posição atual, até a localização de um item determinado dentro da loja. O protótipo, é um aplicativo cliente-servidor, que foi desenvolvido para *iPads*, na linguagem de programação Swift. Para possibilitar as funcionalidades de assistência na loja e rotulagem de produto, foi utilizado a biblioteca AVFoundation e as mensagens JavaScript Object Notation (JSON) para leitura dos códigos de barras. Por outro lado, a leitura os IDs de luz LED, foi utilizada uma solução baseada em Arduino, adaptando uma câmera conectada a um microcontrolador que captura os IDs das luzes via VLC e os transmite para um *iPad* via Bluetooth Low Energy (BLE). No lado do servidor, composto por ferramentas de análise de *big data*, foi configurado com o sistema operacional Ubuntu (versão 14.04.5) e uma aplicação desenvolvida em R (3.4.0). A comunicação entre clientes e servidor foi realizada com uma API Representational State Transfer (REST). Dessa forma, esse sistema permite a navegação interna, recomendação de produtos, gerenciamento de listas de compras e leitura de etiquetas de produtos.

A tabela 2 destaca algumas características observadas no resultado da pesquisa de aplicativos e artigos relacionados com compra de supermercado. Constatou-se que eles oferecem recomendações de produtos alimentícios para nutrição, mas nenhum deles oferece um sistema que cria uma lista de compras personalizada, contendo itens mais saudáveis e de melhor qualidade. Embora, os aplicativos comercializados, citados na tabela 1 realizam a tarefa de facilitar o reconhecimento dos produtos na tarefa de criar a lista de compra para os usuários, esses sistemas não participam com sugestões inteligentes para nutrição.

Tabela 2 – Comparativo do uso de tecnologia dos aplicativos móvel para nutrição.

| Aplicativo | Usa I.A | Cria lista | Multiplataforma | Recomenda |
|-------------------------|----------------|-------------------|------------------------|------------------|
| AnyList Grocery List | Sim | Sim | Não | Sim |
| Boa Lista | Não | Não | Sim | Não |
| Buscapé Mobile | Não | Não | Sim | Sim |
| Buy Me a Pie | Não | Não | Sim | Não |
| Clube Extra | Não | Sim | Sim | Não |
| Desrotulando | Não | Não | Sim | Não |
| Facilista | Não | Sim | Sim | Não |
| HomeRefill | Não | Sim | Não | Não |
| iList Touch | Não | Sim | Não | Não |
| Listonic | Sim | Sim | Não | Sim |
| (MANTHA et al., 2019) | Sim | Sim | - | Sim |
| Mercadapp | Não | Sim | Sim | Não |
| MeuCarrinho | Não | Sim | Sim | Não |
| Pão de Açúcar Mais | Não | Sim | Sim | Não |
| QQFalta | Não | Sim | Não | Não |
| SoftList | Não | Não | Sim | Sim |
| ShoppingList | Sim | Sim | Não | Não |
| ShopWell | Não | Não | Sim | Não |
| Shopper | Não | Sim | Sim | Não |
| (VILLEGAS; SAITO, 2017) | Sim | Sim | Não | Sim |

Fonte: O autor, 2019

2.3 Considerações sobre o capítulo

Os resultados das pesquisas neste capítulo foram divididas em duas partes: aplicativos comercializados no mercado e artigos, que apresentam a criação e implantação de sistemas móveis relacionados com a nutrição. A pesquisa foi realizada buscando conteúdos de diversa fontes bibliográficas e da web, para identificar requisitos importantes para incluir na proposta deste trabalho. Do mesmo modo, foi importante verificar quais as tecnologias e técnicas de desenvolvimento para aplicações móveis, são mais utilizadas e principal-

mente, conhecer a sua operacionalidade para definição da metodologia de implantação do aplicativo Lista Saudável.

3 IMPLEMENTAÇÃO E AVALIAÇÃO DE SISTEMA DE RECOMENDAÇÃO PARA NUTRIÇÃO

A partir dos estudos apresentados nos capítulos 1 e 2 verificou-se que os sistemas estudados no capítulo 2 não exploram técnicas mais inteligentes, que permitiriam oferecer listas de compras mais individualizadas e mais saudáveis. Logo, neste capítulo, são apresentados a metodologia utilizada para o implementação do sistema de recomendação Lista Saudável, assim como, são descritos a metodologia para realizar a experimentação da aplicação.

Foram definidos os seguintes objetivos específicos para o desenvolvimento do sistema:

- (i) Selecionar um *Dataset* contendo produtos alimentícios para análise, treinamento e testes. Determinar quais serão os *features* de entrada nos algoritmos. Sendo assim, investigar como realizar a tarefa de leitura dos rótulos para qualificar os produtos que são mais saudáveis em relação aos menos saudáveis;
- (ii) definir como será o uso dos algoritmos de sistema de recomendação baseado em conteúdo;
- (iii) desenvolver um protótipo do aplicativo para dispositivos móveis, utilizando o *Dataset* treinado, contendo produtos alimentícios para recomendar itens mais saudáveis; e
- (iv) avaliar o desempenho de métodos de recomendação baseado em conteúdo selecionados, usando métricas de avaliação para sistemas de recomendação;

3.1 Recomendação baseada no Nutriscore

O NutriScore é um logotipo que mostra a qualidade nutricional de produtos alimentícios com notas variando de A a E. Com o NutriScore, os produtos podem ser comparados fácil e rapidamente, desconsiderando o valor sugerido para venda por cada fabricante. Para implementação do sistema de recomendação baseado no Nutriscore, foi utilizado o

Dataset Open Food Facts (OPENFOODFACTS, 2019), que é um banco de dados colaborativo liderado pelo francês Stéphane Gigandet, que recebeu a *Open Knowledge Foundation* na França (OKFN, 2019). Mais de 1800 colaboradores contribuíram com mais de 75.000 produtos de 150 países usando o aplicativo fornecido por eles e possibilitando configurar em *Android*, *iPhone*, ou *Windows Phone*, a câmera para escanear códigos de barras e carregar fotos de produtos e seus rótulos. Esse banco de dados de produtos alimentícios contém os ingredientes, alérgenos, informações nutricionais e todas as descrições que podemos encontrar nos rótulos dos produtos. Os dados sobre comida são de interesse público e devem ser abertos, logo, o banco de dados completo é publicado como dados abertos e pode ser reutilizado por qualquer pessoa e para qualquer uso.

O PNNS (PNNS, 2019) usa os dados do *Open Food Facts* para validar a fórmula do seu índice de qualidade nutricional e dos graus de nutrição. Dessa forma, a base francesa possui a classificação da qualidade nutricional de cada produto que varia de A até E, onde, os categorizados com o valor A podem ser considerados produtos mais saudáveis, e os que possuem o valor E, menos saudáveis. Sendo assim, depois de realizar a limpeza no *Dataset*, retirando todas as colunas vazias e filtrando os produtos da França, ficamos com um conjunto de 54.994 produtos para realização dos testes.

"O *PNNS* é um plano de saúde pública que visa melhorar o estado de saúde da população francesa agindo sobre um dos mais importantes determinantes: a nutrição. Para a *PNNS*, nutrição é entendida como o equilíbrio entre as contribuições relacionadas à alimentação e os gastos ocasionados pela atividade física. O *PNNS* tem uma série de estratégias e planos de ações nos setores primários, secundários e terciários de saúde (VERAKIS, 2019)."

Foram utilizadas as colunas "code", "url", "product name", "nutrition grade fr", "categories", "main category" e "nutrition score-fr 100g code", sendo que o primeiro passo foi construir um vetor contendo os nomes dos produtos. E, utilizando algoritmos de *machine learning*, fundamentado na similaridade, de acordo com a descrição de 1.3, obtivemos uma lista contendo itens homogêneos, atendendo o primeiro objetivo desta proposta.

Os estudos apresentados forneceram a base teórica e prática para que a terceira fase do desenvolvimento da proposta seja realizada. Logo, será descrito um aplicativo móvel, mais precisamente, para as plataformas de sistema operacional *Android*, *IOS* ou *windows*

phone, que se propõe a auxiliar usuários a modificarem suas listas de supermercado, optando por alimentos mais saudáveis do que os anteriormente selecionados.

3.1.1 Implementação de Sistema de recomendação para nutrição

Para implementação do sistema de recomendação baseada no *Nutriscore* foi utilizada a linguagem de programação *Python* (PYTHON, 2019) em sua versão 3.7, que é de código aberto e suporta muitos protocolos de internet, assim como, bibliotecas para dados científicos e de educação. O *Jupyter notebook* (JUPYTER, 2019) e o *Anaconda* foram utilizados como ambientes de desenvolvimento *python*, por serem gratuitos e possibilitarem o desenvolvimento de sistemas com *machine learning*, apoiados em um navegador e um servidor remoto (ANACONDA, 2019). As bibliotecas que possuem os métodos e classes de *machine learning*, foram obtidas no pacote *Scikit-learn* (SCIKIT-LEARN, 2019).

Dentre as técnicas apresentadas no Capítulo 1, optamos pelos modelos baseados em conteúdo, que computam a similaridade entre pares de corpos de texto. Neste caso, consideramos o *CountVectorizer* (COUNT-VECTORIZER, 2019), que possui a classe extratora de recursos de texto e o *TfidfVectorizer* (TF-IDF VECTORIZER, 2019), que divide o número de ocorrências de cada palavra em um documento pelo número total de palavras no documento, diminuindo os pesos para palavras que ocorrem muitas vezes no corpus (documento). Conforme apresentado no Capítulo 1 podemos utilizar tanto o *TF-IDF*, como o *CountVectorizer*, para criar o vetor de palavras contendo o nome dos produtos de supermercado.

Nos trabalhos de Cataltepe et al. (CATALTEPE; ULUYAĞMUR; TAYFUR, 2016), Pazzani (PAZZANI, 1999) e Chen (CHEN et al., 2007), foram desenvolvido sistemas de recomendação que utilizaram o *TF-IDF* para compor o vetor de palavras. Kunkel e Farrell (2018) comprovaram que a acurácia, detalhado na seção 1.1.2.7, tem valores mais altos utilizando o *TF-IDF* em relação ao *CountVectorizer*. Embora, os testes confirmem essa eficácia, decidimos que neste trabalho, utilizaremos ambos, o *TF-IDF* e o *CountVectorizer*, como complemento dos testes para criação do *Bag of Words*. Ainda, relacionamos o uso desses vetorizadores com o cálculo da pontuação do *coseno* (ZUVA et al., 2012), para medir a distância entre os vetores de produtos, e assim, obter o quanto esses produtos são similares.

O *Linearkernel* (KERNELS, 2019) é uma biblioteca dentro do *Scikit-learn* que imple-

menta utilitários para avaliar distâncias em pares, ou afinidade de conjuntos de amostras (KERNELS, 2019). Essa classe foi utilizada para realizar o cálculo do produto de pontos de dois vetores numéricos e normalizar pelo produto dos comprimentos de vetor. Para realizar os treinamentos e testes, para garantir que o *dataset* tenha o *nutriscore* adequado, foi utilizada a classe *GradientBoostingClassifier* do pacote *sklearn.ensemble*, que utiliza o algoritmo de *Boosting* para construir um modelo aditivo de maneira progressiva. Adotamos ainda, a abordagem fundamentada em árvore de decisão, considerando a cada estágio, ajustes associados ao gradiente negativo da função de perda de desvio *binomial*, ou *multinomial* (BOOSTING-CLASSIFIER, 2019).

Os testes foram realizados considerando apenas o conjunto de dados, que correspondem ao Brasil no *dataset Open Food Facts*.

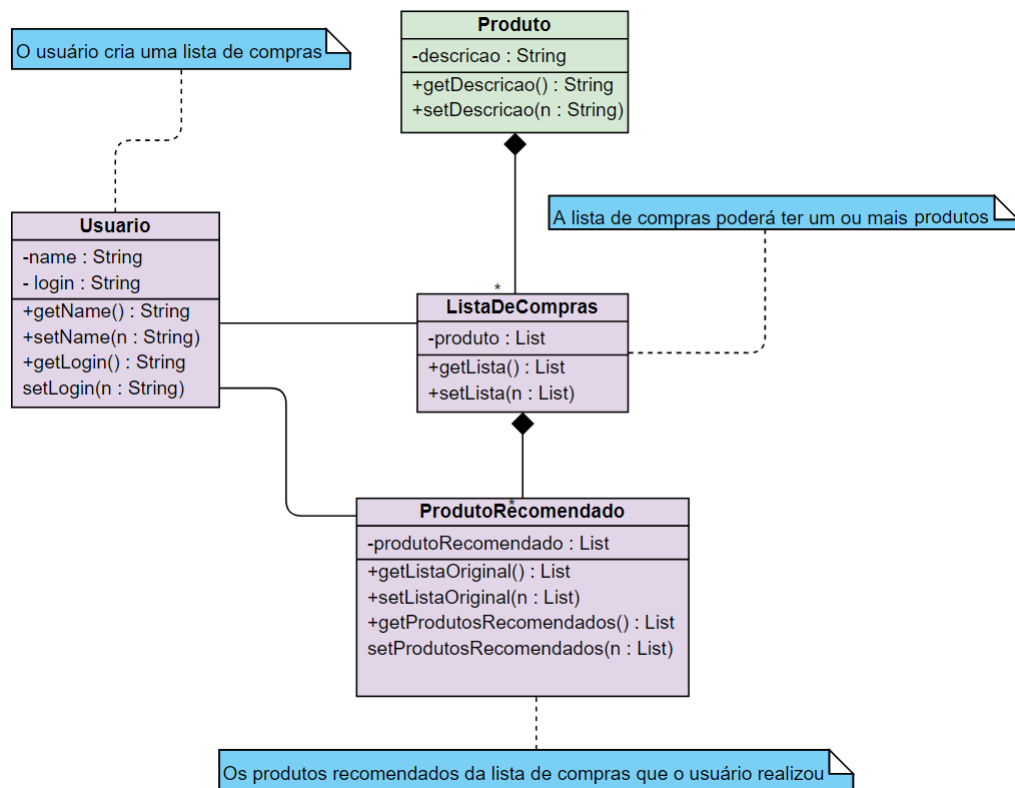
3.2 Metodologia

Após verificar as características dos aplicativos móveis descrito na tabela 1, constatamos que, nenhuma das propostas apresentadas realiza a criação de uma lista de compras contendo produtos alimentícios selecionados por um usuário e recomenda uma outra lista com itens similares, baseada na lista anterior, porém, contendo produtos mais saudáveis. Logo, a partir de um conjunto de atributos intrínsecos, definimos os requisitos para o desenvolvimento desta aplicação: o *dataset* com produtos já classificado, dentre mais saudável e menos saudável; uma linguagem de programação para o *front-end* e o *back-end* do software; quais plataformas o sistema suportará e; quais serão os perfis dos usuários que utilizarão esse aplicativo. Após verificarmos junto ao INPI (Instituto Nacional da Propriedade Industrial) que o nome Lista Saudável e a marca proposta são inexistentes, o sistema foi definido. A partir dos critérios descritos em Anton et al. (2018) para desenvolvimento de sistemas móveis, considerou-se desenvolver o *front-end*, seguido do *back-end*.

A figura 10 apresenta o diagrama de classes, onde a classe de usuário mantém relacionamento com a classe de ListaDeCompras, que possibilita a criação de uma lista de compras, e a partir da utilização desta lista o sistema recomenda outra, com os produtos mais saudáveis, representada pela classe ProdutoRecomendado, onde é disparado previamente o evento processado pelo algoritmo 1, gerando sugestões de outros novos itens.

A figura 11 apresenta o *front-end* da página de login do usuário. Nessa etapa, também

Figura 10 – Diagrama de classes do Lista Saudável



Fonte: O autor, 2019

foram realizados estudos que envolvem a criação e desenvolvimento de um *logotipo*.

O problema do *back-end* do sistema foi elucidado, com a implementação do algoritmo 1 - Similaridade de produto baseado em Nutriscore. Constitui um sistema de recomendação baseado em conteúdo, que foi descrito no capítulo 1.2. Essa técnica foi a escolhida por não considerar dados relacionados ao passado histórico do usuário.

O algoritmo recebe como parâmetros de entrada as colunas: "product name", "nome do produto", "nutrition grade fr", "categories" e "main category" para utilizar o conceito de similaridade. Assim, o Bag of Words citado na seção 1.3, será formado por um vetor contendo o nome do produto, para então aplicar a chamada dos métodos de vetorização `CountVectorizer` e o `TF-IDFVectorizer`, compondo o vetor de produtos similares.

Entretanto, para ser incluído no vetor de saída há uma verificação, que compara o valor do nutriscore do produto selecionado pelo usuário com o valor dos produtos similares, contidos no vetor recentemente gerado. Desse modo, serão adicionados na lista apenas os itens que contiverem valores de nutriscore maiores do que o produto do parâmetro

Figura 11 – Tela de login do aplicativo Lista Saudável



Fonte: O autor, 2019

de entrada. O algoritmo implementado encontra-se detalhado no apêndice 1. Ele faz a recomendação de produtos similares baseado em conteúdo e levando em consideração a classificação do valor de nutriscore realizada previamente no *dataset*.

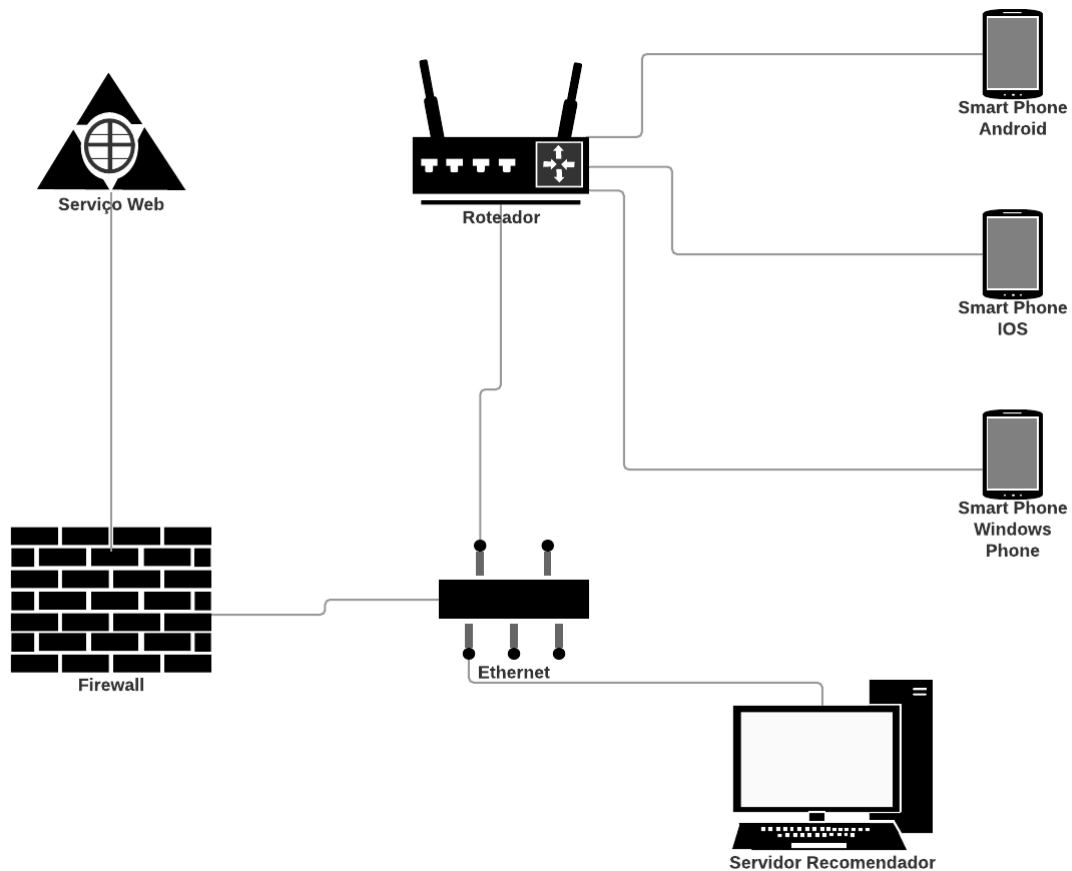
3.3 Recomendação de lista de compra utilizando o Lista Saudável

Idealmente, o sistema de recomendação foi projetado para atender os clientes que acessarão o servidor projetado na figura 12 a partir de seus *smart's phones* com plataforma *Android*, *IOS* ou *windows phone*, que fará sugestões de produtos mais saudáveis para compor suas listas de supermercado. A principal ideia é permitir que, a partir de qualquer dispositivo móvel com acesso à internet, os usuários possam acessar o sistema recomendador.

O aplicativo Lista Saudável foi implementado utilizando linguagens web como HTML5, CSS3 e Javascript, para possibilitar que o mesmo código pode ser compilado para sistemas operacionais diferente. Essa abordagem proporciona a criação de um sistema para dispositivo móvel multiplataforma.

A figura 13 é a tela inicial de criação de uma lista de supermercado, após o usuário se

Figura 12 – Representação do projeto do Sistema de recomendação do Lista Saudável



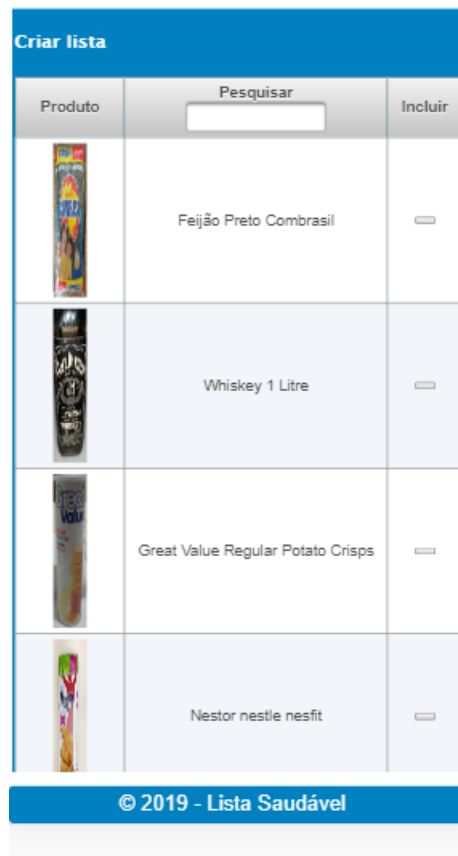
Fonte: O autor, 2019

logar no sistema. É apresentada a listagem dos itens com o limite fixo de 10 produtos, e um campo para buscar qualquer produto que não esteja sendo visualizado. Para incluir o produto na lista, basta o usuário clicar no botão de incluir, para que o produto escolhido seja exibido na lista de criação. Nesta etapa, os produtos ainda estão armazenados em memória na lista temporária de produtos escolhidos.

A criação de uma lista com produtos escolhidos pelo usuário, será exibida logo abaixo da listagem de todos os produtos disponíveis, caso o usuário queira remover algum produto da lista de produtos escolhidos, basta clicar no botão de excluir. Para salvar a lista de compras, o usuário deverá clicar no botão de salvar, deslizando para baixo até que o último produto escolhido seja exibido como na figura 14. A exibição da lista dos produtos escolhidos segue a ordem de sua inclusão.

Finalmente, após clicar no botão de salvar, o aplicativo Lista Saudável irá primeiramente, salvar a lista original no banco de dados (remoto), e sugerir ao usuário uma

Figura 13 – Tela de criação de lista de produtos do aplicativo Lista Saudável

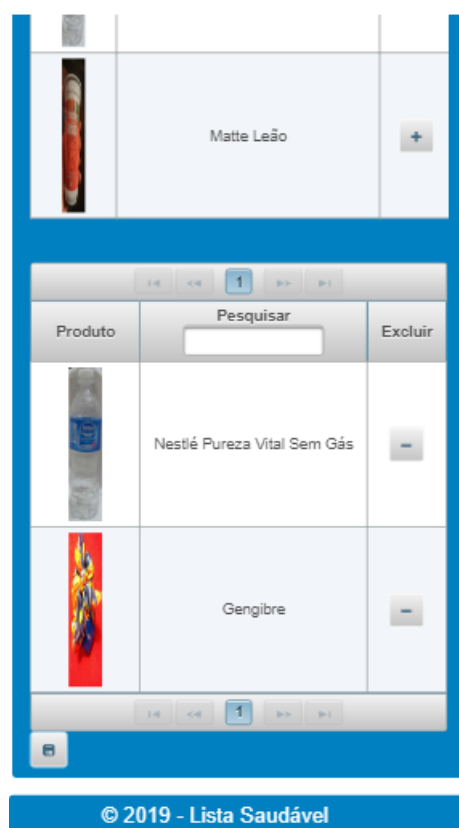


Fonte: O autor, 2019

nova lista com a mesma quantidade de produtos da lista temporária, porém poderá haver dentre os itens, algum que foi alterado pelo sistema. No caso em que não tenha sido encontrado um outro produto similar com nutriscore maior que o da lista original, será exibido o mesmo produto que foi inserido anteriormente. Em seguida, será apresentada a tela contendo a lista de produtos recomendados.






O sistema possibilita que o usuário recuse a recomendação e salve apenas a lista de compras escolhida, basta deslizar com a tela para baixo até o último item da lista recomendada para visualizar os botões com essas opções.

Figura 14 – Tela de criação de lista de produtos do aplicativo Lista Saudável



Fonte: O autor, 2019

Figura 15 – Tela da lista recomendada pelo aplicativo Lista Saudável

| Produtos Recomendados | |
|---|------------------------|
| Produto | Descrição |
|  | Feijão Preto Combrasil |
|  | Nestor nestle nesfit |
|  | Gengibre |
|  | Terrabusi |
|  | Tic tac 100 |

Salvar lista Recomendada

Salvar lista Original

© 2019 - Lista Saudável

Fonte: O autor, 2019

3.4 CONSIDERAÇÕES FINAIS

Este capítulo apresentou a metodologia utilizada para o desenvolvimento do aplicativo móvel baseado em nutriscore. Os aspectos observados nos trabalhos correlatos foram fundamentais para a definição das técnicas a serem utilizadas no aplicativo. Foram consideradas várias tecnologias atuais e que podem ser utilizadas em diferentes plataformas móveis. Foram adotadas técnicas de vetorização, que foram estudadas e apresentadas no capítulo anterior, para formar a base das escolhas do sistema.

Visando observar o desempenho do aplicativo, o próximo capítulo apresenta resultados preliminares de experimentações realizadas com o dataset Open Food Facts, verificando a aplicação do algoritmo desenvolvido, para selecionar produtos similares mais bem qualificados em relação aos seus valores nutricionais.

4 TESTES

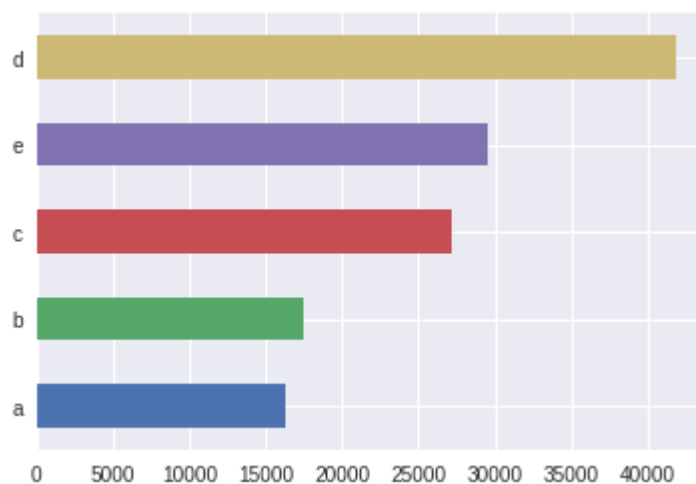
Neste capítulo, são apresentados resultados preliminares de testes realizados com o *Dataset Open Food Facts*.

4.1 Criação de um dataset para os testes

Durante a análise exploratória de dados foi verificado como os produtos estão distribuídos no *Dataset Open Food Facts*. Observou-se que a maior parte dos itens, está concentrada na classificação D do campo do *Nutriscore*, considerando o filtro da França.

Em seguida, realizamos a verificação dos dados dos produtos por categoria cadastrada. Essas informações serviram de ponto de partida para construir o algoritmo que recomenda produtos baseado em similaridade e em *Nutriscore*, conforme as figuras 16 e 17.

Figura 16 – Quantidade de produtos da França por nutriscore

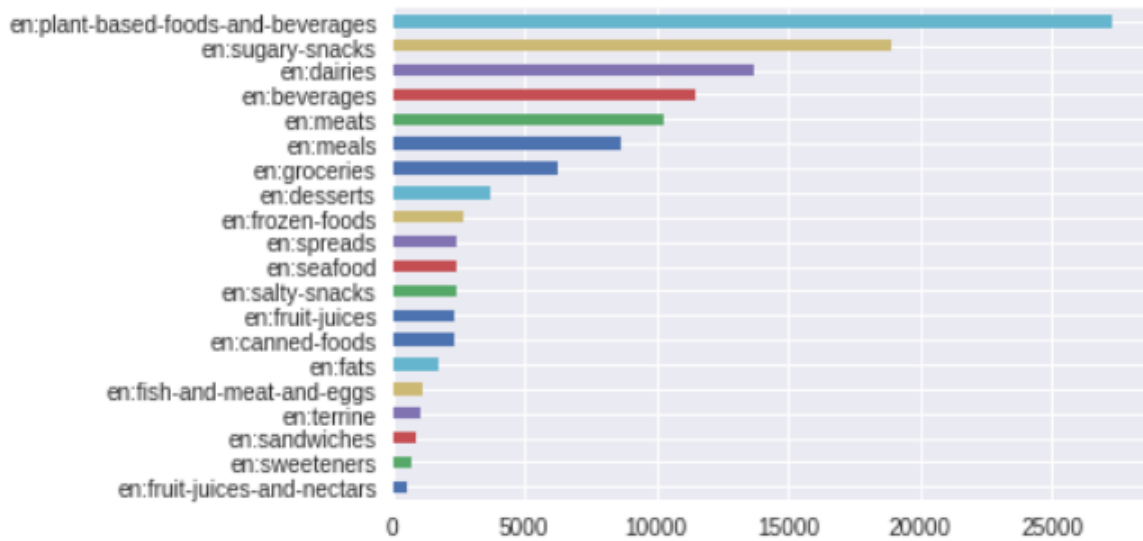


Fonte: O autor, 2019

Para criar o *Dataset* com produtos comercializados no Brasil, primeiramente, foi realizado um filtro sobre a coluna "countries" com o valor igual a "Brasil" a partir do *Open Food Facts*, que mantém uma base com produto de 150 países. O resultado dessa operação foi a geração de um subconjunto contendo 409 itens brasileiros.

Após realizar a limpeza dos dados, ou seja, retirar as colunas com valor vazios ou nulos, restaram 378 produtos para treinamento e testes. O Brasil não possui um programa

Figura 17 – Quantidade de produtos da França por categoria



Fonte: O autor, 2019

análogo PNNS, sendo assim, a coluna "nutrition grade fr" obviamente, está vazia para este filtro. Para solucionar esse problema, mais uma vez, foram utilizados os conceitos descritos no capítulo 1.1.2, consequentemente, utilizar a técnica de *Regressão logística* para treinar o *Dataset* que contém apenas produtos brasileiros, a partir dos dados do *Dataset* composto por produtos franceses, que contém o *Nutriscore* de cada item. Desse modo o algoritmo prediz o valor da coluna "nutrition grade fr".

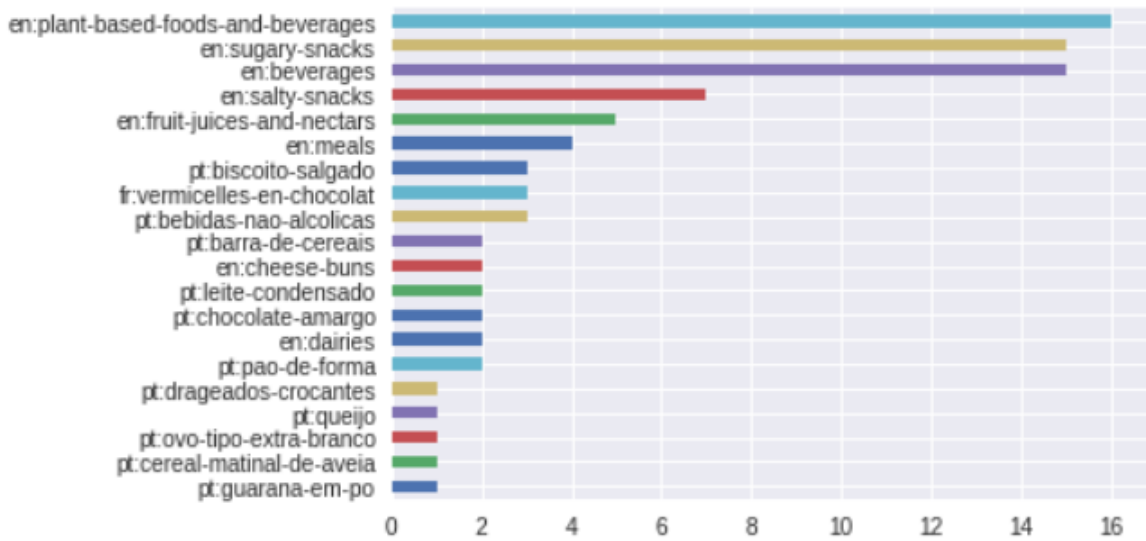
Os testes foram realizados considerando apenas os dados que corresponde ao Brasil no *Dataset Open Food Facts*. Durante a fase de análise exploratória dos dados brasileiros, averiguamos a quantidade de produtos por categoria, detalhadas na figura 18.

Nesta etapa, foi utilizada a *Regressão logística* para possibilitar a previsão dos cinco valores possíveis de Nutriscore, com valores compreendidos de A até E, correspondido pela coluna 'nutrition grade fr code'.

Dessa forma, a partir do *Dataset* francês já treinado, foi possível o preenchimento da coluna vazia no *Dataset* do Brasil. Os resultados da implementação dos testes demonstrados a seguir, foram obtidos com 4 parâmetros de entrada: 'product name code', 'main category code', 'categories' e 'nutrition grade fr code':

Os resultados apresentados na tabela 3, consideram a classificação do Nutriscore dos produtos do *Dataset* francês, tendo sido aplicados os algoritmos para classificação: Boosting, Logistic Regression, KNN, reservando 75% do *Dataset* para treino e 25% para teste.

Figura 18 – Quantidade de produtos do Brasil por categoria



Fonte: O autor, 2019

O melhor resultado obtido neste caso foi de 0.63 de *Acurácia*, utilizando o algoritmo de *Boosting* e de Regressão logística, o que não é considerado um resultado satisfatório.

Tabela 3 – Resultados com 75% do *Dataset* para treino e 25% para teste.

| Algoritmo | Acurácia | Recall | MAE | MSE | RMSE |
|---------------------|----------|--------|------|------|------|
| Boosting | 0.63 | 0.58 | 0.77 | 1.32 | 1.15 |
| Logistic Regression | 0.63 | 0.58 | 0.77 | 1.32 | 1.15 |
| KNN | 0.55 | 0.75 | 0.66 | 1.24 | 1.11 |

Fonte: O autor, 2019

Sendo assim, após avaliar os resultados apresentados na tabela 3, ajustamos o *Dataset* para 70% para treino e 30% para teste. Desta forma, melhorou o valor da *Acurácia* para 0.65, quando foi utilizado o algoritmo de *Boosting*, conforme a tabela 4.

Tabela 4 – Resultados com 70% para treino e 30% para teste.

| Algoritmo | Acurácia | Recall | MAE | MSE | RMSE |
|---------------------|-----------------|---------------|------------|------------|-------------|
| Boosting | 0.65 | 0.58 | 0.77 | 1.35 | 1.16 |
| Logistic Regression | 0.62 | 0.59 | 0.71 | 1.18 | 1.09 |
| KNN | 0.55 | 0.76 | 0.66 | 1.23 | 1.11 |

Fonte: O autor, 2019

Após análise dos resultados da tabela 4, ajustamos o *Dataset* para 60% para treino e 40% para teste, mas não obtivemos melhora considerando o valor da *Acurácia* para o algoritmo de *Boosting* nem para o de Regressão Logística. Porém, apresentou pequena melhora no valor de *Recall*, para 0.59.

Tabela 5 – Resultados com 60% para treino e 40% para teste.

| Algoritmo | Acurácia | Recall | MAE | MSE | RMSE |
|---------------------|-----------------|---------------|------------|------------|-------------|
| Boosting | 0.63 | 0.59 | 0.77 | 1.33 | 1.15 |
| Logistic Regression | 0.63 | 0.59 | 0.77 | 1.33 | 1.15 |
| KNN | 0.54 | 0.74 | 0.69 | 1.30 | 1.14 |

Fonte: O autor, 2019

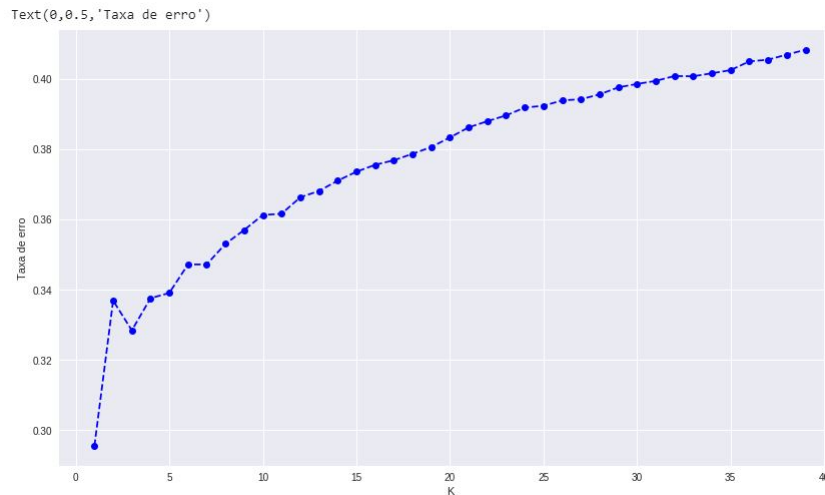
Logo, observou-se a necessidade de fazer outros ajustes no modelo.

4.2 Ajuste dos parâmetros

Os parâmetros de entrada para o algoritmo de classificação, inicialmente, foram as colunas: "product name", "nutrition grade fr", "categories" e "main category". Logo, percebeu-se a necessidade de considerar mais um parâmetro sobre os dados, para obter melhores respostas desse algoritmo. Após análise de outros possíveis valores nas colunas do *Dataset*, constatamos que o valor de "nutrition score fr 100g code" é preenchida com um valor numérico, que representa a quantidade do valor nutricional para cada produto, ou seja, tornando-se mais um dado categórico de entrada. Para o algoritmo de *KNN* o

valor de k foi ajustado para 1 de acordo com o resultado utilizando o método do cotovelo (KODINARIYA; MAKWANA, 2013) e descrito na Figura 19:

Figura 19 – Melhor valor de k para o algoritmo - KNN



Fonte: O autor, 2019

4.3 Avaliação do sistema de recomendação

Para selecionar o melhor subconjunto possível de algoritmos de classificação em termos de desempenho, foi realizada uma fase inicial de pré-seleção com os algoritmos selecionados: *Boosting*, *KNN* e *Logistic Regression*. Esses algoritmos receberam o mesmo conjunto de dados para treinamento e avaliados usando os conjuntos de teste e validação mencionados na seção 1.1.2. Como estimadores de desempenho, selecionamos cinco métricas (*Acurácia*, *Recall*, *MAE*, *MSE*, *RMSE*) para avaliar os pontos fortes e fracos de cada algoritmo.

Ao realizarmos os experimentos, inicialmente, verificamos que o algoritmo *Gradient boosting*, com 30% dos dados do dataset reservados para teste e 70% para treino, obteve a melhor resposta, durante o treinamento da previsão do *Nutriscore* com o *Dataset* brasileiro a partir dos dados franceses, confirmando os resultados descritos nas tabelas 3, 4 e 5.

O Algoritmo 1 implementou um recomendador baseado em conteúdo para recomendação de produtos similares. Este recebe como parâmetros de entrada as colunas: "product name", "nutrition grade fr", "categories" e "main category". A coluna de *Nutriscore* ("nutrition grade fr") do *Dataset* brasileiro foi preenchida pelos valores obtidos através dos resultados do treinamento da classificação. Na criação do chamado *Bag of Words*, citado

na seção 1.3, foram utilizados os métodos: *CountVectorizer* e o *TF-IDFVectorizer*. Foi observado que não houve distinção de uso desses métodos para o *dataset* utilizado, ambos apresentaram os resultados satisfatórios, para compor o vetor de palavras (*Bag of Words*).

Na página web do Open Food Facts encontraremos informações como nome do produto, categoria, nutriscore, sua tabela nutricional, dentre outras.

Para exemplificar o funcionamento da decisão de escolha de um produto no Lista Saudável, ilustramos na figura 20 o caso em que o usuário escolheu o item com descrição igual a do feijão fradinho e incluído em sua lista de compras. Esse produto tem o nutriscore igual a B.

Figura 20 – Produto cadastrado no Open Food Facts - Feijão fradinho pote



Fonte: (OPENFOODFACTS, 2019)

Quando o usuário finalizar a lista (ação do botão salvar), o sistema deverá se basear em similaridade do nome do produto para realizar uma busca por outro similar mais saudável, assim, espera-se uma recomendação de outro item do mesmo tipo, feijão, mas com o nutriscore maior, por exemplo, poderia sugerir o produto demonstrado na figura 21 que tem o nutriscore igual a A.

Dessa forma, esse passo será realizado para cada um dos produtos e acrescentados em uma lista temporária, e ao finalizar esse processo, o usuário receberá a tela com a lista recomendada pelo sistema como na figura 15, contendo produtos mais saudáveis do que

Figura 21 – Produto cadastrado no Open Food Facts - Feijão manteiga cozido

Feijão Manteiga Cozido - O Cultivador - 850 g / 500 g escorrido
 Barcode: 22043108

Product characteristics
 Common name: Feijão-manteiga cozido
 Quantity: 850 g / 500 g escorrido
 Packaging: Lata, Conserva
 Brands: O Cultivador, ALDI, MEPS, Maria Emília Pereira Soares & Filhos Lda
 Categories: Plant based foods and beverages, Plant based foods, Legumes and their products, Canned foods, Legumes, Canned plant based foods, Canned legumes, Canned common beans, p_0ear1
 Manufacturing or processing places: Bouças, Fafe, Portugal
 Stores: ALDI
 Countries where sold: Portugal

Ingredients
 Ingredients list: Feijão-manteiga, água e sal.

Nutrition facts
 NutriScore color nutrition grade: **E**
 Nutrient levels for 100 g:
 - 0.7 g Fat in low quantity
 - 0.8 g Saturated fat in low quantity
 - 0.8 g Sugars in low quantity
 - 0.66 g Salt in moderate quantity

Ingredients/Ingrédients:
 Feijão-manteiga, água e sal.

| VALORES NUTRICIONAIS MÉDIOS (aproximado) | | VALORES NUTRITIONNELS MOYENS (approximé) | |
|--|----------|--|-----|
| | Por 100g | Por 100g | % |
| Energia / Energie | 330 kJ | 77 kcal | 14 |
| Lipídios / Graisses | 0.7 g | 0.2 g | 0.1 |
| - dos quais saturados / - dont saturés | 0.1 g | 0.1 g | 0.2 |
| Glúcidos de carbono / Sucres | 0.8 g | 0.8 g | 1.6 |
| - dos quais açúcares / - dont sucres | 0.8 g | 0.8 g | 1.6 |
| Proteínas / Protéines | 8.4 g | 8.4 g | 16 |
| Sal / Sel | 0.66 g | 0.66 g | 13 |

Fonte: (OPENFOODFACTS, 2019)

os escolhidos anteriormente.

Assim, consideramos que o usuário do sistema realizou uma lista de compras contendo os itens: água de coco, biscoito e manteiga e os nutriscore E, D e E respectivamente, conforme a tabela 6. Nesse caso, o sistema seguirá os passos já mencionados para salvar a lista e recomendaria, por exemplo, os alimentos descritos na tabela 7. A recomendação levou em consideração itens de mesma categoria, porém com os nutriscore maiores, e com valores iguais a D, B e D.

Tabela 6 – Lista de produtos escolhidos pelo usuário

| Produto | NutriScore | Tabela Nutricional | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|--|-----------------------|----------------------------|----------------------------|-----------------------|-------------|----------|---------------|----------|---------------|----------|---------------------|-----|--------|---------------------|-----------------|-----------------|------|-------|-----------------------|--------|-----------------|-----------------------|---------|-------|---------------|--------|--------|---------------|----------|-------|----------|-------|--------|--------------------------|---------|-------|-------------|-------|---------|------|-------|--------|---------|--------|--------|-----------|--------|------|--------------------------|---------|----|--------------------------|-------------|-----|-------------|---|---|
| <p>Open Food Facts Country Discover Contribute</p> <p>Agua de coco integral - Obrigado</p> <p>Barcode: 789973270501 (EAN / EAN-13)</p> <p>Product characteristics</p> <p>Brands: Obrigado</p> <p>Categories: Plant-based foods and beverages, Beverages, Plant-based beverages, Non-Alcoholic beverages, Coconut waters, Unsweetened beverages</p> <p>Countries where sold: Brazil</p> | <p>NUTRI-SCORE</p> <p>A B C D E</p> <p>Warning: the amount of fiber is not specified, their possible positive contribution to the grade could not be taken into account.</p> <p>Details of the calculation of the Nutri-Score ></p> <p>Nutrient levels for 100 g</p> <ul style="list-style-type: none"> 0 g Fat in low quantity 0 g Saturated fat in low quantity 9.4 g Sugars in high quantity 0 g Salt in low quantity <p>Comparison to average values of products in the same category:</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Coconut waters (218 products) <input type="checkbox"/> Non-Alcoholic beverages (2432 products) <input type="checkbox"/> Unsweetened beverages (3570 products) <input type="checkbox"/> Plant-based beverages (30939 products) <input type="checkbox"/> Beverages (86328 products) <input type="checkbox"/> Plant-based foods and beverages (200464 products) <p>• % of difference ○ value for 100 g / 100 ml</p> | <table border="1"> <thead> <tr> <th>Nutrition facts</th> <th>As sold for 100 g / 100 ml</th> <th>Coconut waters</th> </tr> </thead> <tbody> <tr> <td>Energy (kJ)</td> <td>?</td> <td>94 kJ</td> </tr> <tr> <td>Energy (kcal)</td> <td>19 kcal</td> <td>-5%</td> </tr> <tr> <td>Energy</td> <td>79 kJ (19 kcal)</td> <td>-9%</td> </tr> <tr> <td>Fat</td> <td>0 g</td> <td>-100%</td> </tr> <tr> <td>- Saturated fat</td> <td>0 g</td> <td>-100%</td> </tr> <tr> <td>Carbohydrates</td> <td>9.4 g</td> <td>+89%</td> </tr> <tr> <td>- Sugars</td> <td>9.4 g</td> <td>+132%</td> </tr> <tr> <td>Proteins</td> <td>0 g</td> <td>-100%</td> </tr> <tr> <td>Salt</td> <td>0 g</td> <td>-100%</td> </tr> <tr> <td>Sodium</td> <td>0 g</td> <td>-100%</td> </tr> <tr> <td>Nutrition score - France</td> <td>10</td> <td>+228%</td> </tr> <tr> <td>Nutri-Score</td> <td>E</td> <td>E</td> </tr> </tbody> </table> | Nutrition facts | As sold for 100 g / 100 ml | Coconut waters | Energy (kJ) | ? | 94 kJ | Energy (kcal) | 19 kcal | -5% | Energy | 79 kJ (19 kcal) | -9% | Fat | 0 g | -100% | - Saturated fat | 0 g | -100% | Carbohydrates | 9.4 g | +89% | - Sugars | 9.4 g | +132% | Proteins | 0 g | -100% | Salt | 0 g | -100% | Sodium | 0 g | -100% | Nutrition score - France | 10 | +228% | Nutri-Score | E | E | | | | | | | | | | | | | | | | | | |
| Nutrition facts | As sold for 100 g / 100 ml | Coconut waters | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy (kJ) | ? | 94 kJ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy (kcal) | 19 kcal | -5% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy | 79 kJ (19 kcal) | -9% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fat | 0 g | -100% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Saturated fat | 0 g | -100% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Carbohydrates | 9.4 g | +89% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Sugars | 9.4 g | +132% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Proteins | 0 g | -100% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Salt | 0 g | -100% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Sodium | 0 g | -100% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutrition score - France | 10 | +228% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutri-Score | E | E | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Open Food Facts Country Discover Contribute</p> <p>Bolacha Maria - Cuétara - 800g</p> <p>Barcode: 8634185479852 (EAN / EAN-13)</p> <p>Product characteristics</p> <p>Quantity: 800g</p> <p>Brands: Cuétara</p> <p>Categories: Biscuits, Sweet snacks, Biscuits and cakes, Biscuits, Biscuits edulcorés</p> <p>Manufacturing or processing place: Espagne</p> <p>EMSC code: L92013A</p> <p>Stores: Casasa</p> <p>Countries where sold: France</p> | <p>NUTRI-SCORE</p> <p>A B C D E</p> <p>Details of the calculation of the Nutri-Score ></p> <p>Nutrient levels for 100 g</p> <ul style="list-style-type: none"> 8 g Fat in moderate quantity 1.8 g Saturated fat in moderate quantity 24 g Sugars in high quantity 0.93 g Salt in moderate quantity <p>Serving size: 5.3g</p> <p>Comparison to average values of products in the same category:</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> fr:Biscuits edulcorés (51 products) <input type="checkbox"/> Biscuits (22741 products) <input type="checkbox"/> Biscuits and cakes (49353 products) <input type="checkbox"/> Sweet snacks (86104 products) <input type="checkbox"/> Snacks (134036 products) <p>• % of difference ○ value for 100 g / 100 ml</p> | <table border="1"> <thead> <tr> <th>Nutrition facts</th> <th>As sold for 100 g / 100 ml</th> <th>As sold per serving (5.3g)</th> <th>fr:Biscuits edulcorés</th> </tr> </thead> <tbody> <tr> <td>Energy (kJ)</td> <td>?</td> <td>?</td> <td>1,780 kJ</td> </tr> <tr> <td>Energy (kcal)</td> <td>424 kcal</td> <td>22.5 kcal</td> <td>0%</td> </tr> <tr> <td>Energy</td> <td>1,774 kJ (424 kcal)</td> <td>94 kJ (22 kcal)</td> <td>-0%</td> </tr> <tr> <td>Fat</td> <td>8 g</td> <td>0.424 g</td> <td>-54%</td> </tr> <tr> <td>- Saturated fat</td> <td>1.8 g</td> <td>0.095 g</td> <td>-72%</td> </tr> <tr> <td>Carbohydrates</td> <td>80 g</td> <td>4.24 g</td> <td>+29%</td> </tr> <tr> <td>- Sugars</td> <td>24 g</td> <td>1.27 g</td> <td>+967%</td> </tr> <tr> <td>Fibers</td> <td>2 g</td> <td>0.106 g</td> <td>-69%</td> </tr> <tr> <td>Proteins</td> <td>7 g</td> <td>0.371 g</td> <td>-14%</td> </tr> <tr> <td>Salt</td> <td>0.93 g</td> <td>0.049 g</td> <td>+54%</td> </tr> <tr> <td>Sodium</td> <td>0.372 g</td> <td>0.02 g</td> <td>+54%</td> </tr> <tr> <td>Nutrition score - France</td> <td>13</td> <td>13</td> <td>+74%</td> </tr> <tr> <td>Nutri-Score</td> <td>D</td> <td>D</td> <td>D</td> </tr> </tbody> </table> | Nutrition facts | As sold for 100 g / 100 ml | As sold per serving (5.3g) | fr:Biscuits edulcorés | Energy (kJ) | ? | ? | 1,780 kJ | Energy (kcal) | 424 kcal | 22.5 kcal | 0% | Energy | 1,774 kJ (424 kcal) | 94 kJ (22 kcal) | -0% | Fat | 8 g | 0.424 g | -54% | - Saturated fat | 1.8 g | 0.095 g | -72% | Carbohydrates | 80 g | 4.24 g | +29% | - Sugars | 24 g | 1.27 g | +967% | Fibers | 2 g | 0.106 g | -69% | Proteins | 7 g | 0.371 g | -14% | Salt | 0.93 g | 0.049 g | +54% | Sodium | 0.372 g | 0.02 g | +54% | Nutrition score - France | 13 | 13 | +74% | Nutri-Score | D | D | D | |
| Nutrition facts | As sold for 100 g / 100 ml | As sold per serving (5.3g) | fr:Biscuits edulcorés | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy (kJ) | ? | ? | 1,780 kJ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy (kcal) | 424 kcal | 22.5 kcal | 0% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy | 1,774 kJ (424 kcal) | 94 kJ (22 kcal) | -0% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fat | 8 g | 0.424 g | -54% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Saturated fat | 1.8 g | 0.095 g | -72% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Carbohydrates | 80 g | 4.24 g | +29% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Sugars | 24 g | 1.27 g | +967% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fibers | 2 g | 0.106 g | -69% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Proteins | 7 g | 0.371 g | -14% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Salt | 0.93 g | 0.049 g | +54% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Sodium | 0.372 g | 0.02 g | +54% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutrition score - France | 13 | 13 | +74% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutri-Score | D | D | D | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Open Food Facts Country Discover Contribute</p> <p>Manteiga com sal - Mimosas - 250 g</p> <p>Barcode: 3601049812336 (EAN / EAN-13)</p> <p>Product characteristics</p> <p>Common name: Manteiga com sal</p> <p>Quantity: 250 g</p> <p>Packaging: Plástico</p> <p>Brands: Mimosas</p> <p>Categories: Dairies, Spreads, Fats, Spreadable fats, Animal fats, Milkfat, Dairy spread, Butters, Salted butters</p> <p>EMSC code: PT BL 7 CE</p> <p>Countries where sold: France, Portugal</p> | <p>Nutrition facts</p> <p>NutriScore color nutrition grade</p> <p>NUTRI-SCORE</p> <p>A B C D E</p> <p>Details of the calculation of the Nutri-Score ></p> <p>Nutrient levels for 100 g</p> <ul style="list-style-type: none"> 81.6 g Fat in high quantity 51 g Saturated fat in high quantity 0.7 g Sugars in low quantity 1.3 g Salt in moderate quantity <p>Comparison to average values of products in the same category:</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Salted butters (186 products) <input type="checkbox"/> Butters (2179 products) <input type="checkbox"/> Dairy spread (2183 products) <input type="checkbox"/> Milkfat (2195 products) <input type="checkbox"/> Animal fats (2356 products) <input type="checkbox"/> Spreadable fats (3264 products) <input type="checkbox"/> Fats (17047 products) <input type="checkbox"/> Spreads (32064 products) <input type="checkbox"/> Dairies (71276 products) <p>• % of difference ○ value for 100 g / 100 ml</p> | <table border="1"> <thead> <tr> <th>Nutrition facts</th> <th>As sold for 100 g / 100 ml</th> <th>Salted butters</th> </tr> </thead> <tbody> <tr> <td>Energy (kJ)</td> <td>?</td> <td>3,010 kJ</td> </tr> <tr> <td>Energy (kcal)</td> <td>727 kcal</td> <td>+1%</td> </tr> <tr> <td>Energy</td> <td>3,042 kJ (727 kcal)</td> <td>+1%</td> </tr> <tr> <td>Fat</td> <td>81.6 g</td> <td>+3%</td> </tr> <tr> <td>- Saturated fat</td> <td>51 g</td> <td>+3%</td> </tr> <tr> <td>- Monounsaturated fat</td> <td>24.1 g</td> <td>+13%</td> </tr> <tr> <td>- Polyunsaturated fat</td> <td>5.7 g</td> <td>+325%</td> </tr> <tr> <td>- Cholesterol</td> <td>230 mg</td> <td>+11%</td> </tr> <tr> <td>Carbohydrates</td> <td>0.7 g</td> <td>+126%</td> </tr> <tr> <td>- Sugars</td> <td>0.7 g</td> <td>+71%</td> </tr> <tr> <td>Fibers</td> <td>0 g</td> <td>-100%</td> </tr> <tr> <td>Proteins</td> <td>0.6 g</td> <td>+91%</td> </tr> <tr> <td>Salt</td> <td>1.3 g</td> <td>-18%</td> </tr> <tr> <td>Sodium</td> <td>0.52 g</td> <td>-18%</td> </tr> <tr> <td>Vitamin A</td> <td>724 µg</td> <td>-12%</td> </tr> <tr> <td>Vitamin E</td> <td>2.33 mg</td> <td>?</td> </tr> <tr> <td>Nutrition score - France</td> <td>23</td> <td>-3%</td> </tr> <tr> <td>Nutri-Score</td> <td>E</td> <td>E</td> </tr> </tbody> </table> | Nutrition facts | As sold for 100 g / 100 ml | Salted butters | Energy (kJ) | ? | 3,010 kJ | Energy (kcal) | 727 kcal | +1% | Energy | 3,042 kJ (727 kcal) | +1% | Fat | 81.6 g | +3% | - Saturated fat | 51 g | +3% | - Monounsaturated fat | 24.1 g | +13% | - Polyunsaturated fat | 5.7 g | +325% | - Cholesterol | 230 mg | +11% | Carbohydrates | 0.7 g | +126% | - Sugars | 0.7 g | +71% | Fibers | 0 g | -100% | Proteins | 0.6 g | +91% | Salt | 1.3 g | -18% | Sodium | 0.52 g | -18% | Vitamin A | 724 µg | -12% | Vitamin E | 2.33 mg | ? | Nutrition score - France | 23 | -3% | Nutri-Score | E | E |
| Nutrition facts | As sold for 100 g / 100 ml | Salted butters | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy (kJ) | ? | 3,010 kJ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy (kcal) | 727 kcal | +1% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy | 3,042 kJ (727 kcal) | +1% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fat | 81.6 g | +3% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Saturated fat | 51 g | +3% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Monounsaturated fat | 24.1 g | +13% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Polyunsaturated fat | 5.7 g | +325% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Cholesterol | 230 mg | +11% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Carbohydrates | 0.7 g | +126% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Sugars | 0.7 g | +71% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fibers | 0 g | -100% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Proteins | 0.6 g | +91% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Salt | 1.3 g | -18% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Sodium | 0.52 g | -18% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Vitamin A | 724 µg | -12% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Vitamin E | 2.33 mg | ? | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutrition score - France | 23 | -3% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutri-Score | E | E | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Fonte: O autor, 2019

Tabela 7 – Lista de produtos recomendados pelo Lista Saudável

| Produto | NutriScore | Tabela Nutricional | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|--|-----------------|----------------------------|------------------------------|----------------|-------------|----------|---------------|----------|---------------|--------|---------------------|-------------------|--------|-----------------|-----------------|-----------------|------|-------|---------------|-------|-----------------|----------|------|-------|---------------|-------|--------|--------|----------|--------|----------|--------|--------|--------------------------|-------|--------|-------------|--------|--------|--------------------------|------|---------|-------------|------|--------|---------|---------|------|---------|---------|---------|---------|-----------|--------|--------|------|--------------------------|---|---|------|-------------|---|---|---|
| <p>Original Coco - Isola Bio - 500 ml</p> <p>Barcode: 80023876282305 (EAN / EAN-13)</p> <p>Quantity: 500 ml</p> <p>Brand: Isola Bio</p> <p>Categories: Plant-based foods and beverages, Beverages, Plant-based beverages, Fruit-based beverages, Non-Alcoholic beverages, Coconut waters, Unsweetened beverages</p> <p>Labels, certifications, awards: Organic, EU Organic, EF-03-008</p> <p>Origin of ingredients: Brazil</p> <p>Manufacturing or processing places: Italia</p> <p>Link to the product page on the official site of the producer: http://www.isolabio.com/pt/originais/coco</p> <p>Brand: Original Coco</p> <p>Countries where sold: Portugal, Spain</p> | <p>NutriScore color nutrition grade D</p> <p>NUTRI-SCORE</p> <p>A B C D E</p> <p>Details of the calculation of the Nutri-Score »</p> <p>Nutrient levels for 100 g</p> <ul style="list-style-type: none"> 0 g Fat in low quantity 0 g Saturated fat in low quantity 3.8 g Sugars in moderate quantity 0.023 g Salt in low quantity <p>Serving size: 100 ml</p> <p>Comparison to average values of products in the same category:</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Coconut waters (218 products) <input type="checkbox"/> Non-Alcoholic beverages (2432 products) <input type="checkbox"/> Unsweetened beverages (3570 products) <input type="checkbox"/> Fruit-based beverages (12033 products) <input type="checkbox"/> Plant-based beverages (30939 products) <input type="checkbox"/> Beverages (86328 products) <input type="checkbox"/> Plant-based foods and beverages (200464 products) <p>• % of difference ○ value for 100 g / 100 ml</p> | <table border="1"> <thead> <tr> <th>Nutrition facts</th> <th>As sold for 100 g / 100 ml</th> <th>As sold per serving (100 ml)</th> <th>Coconut waters</th> </tr> </thead> <tbody> <tr> <td>Energy (kJ)</td> <td>85 kJ</td> <td>85 kJ</td> <td>-10%</td> </tr> <tr> <td>Energy (kcal)</td> <td>?</td> <td>?</td> <td>20 kcal</td> </tr> <tr> <td>Energy</td> <td>85 kJ (20 kcal)</td> <td>85 kJ (20 kcal)</td> <td>-2%</td> </tr> <tr> <td>Fat</td> <td>0 g</td> <td>0 g</td> <td>-100%</td> </tr> <tr> <td>- Saturated fat</td> <td>0 g</td> <td>0 g</td> <td>-100%</td> </tr> <tr> <td>Carbohydrates</td> <td>5 g</td> <td>5 g</td> <td>+0%</td> </tr> <tr> <td>- Sugars</td> <td>3.8 g</td> <td>3.8 g</td> <td>-6%</td> </tr> <tr> <td>Fibers</td> <td>0 g</td> <td>0 g</td> <td>-100%</td> </tr> <tr> <td>Proteins</td> <td>0 g</td> <td>0 g</td> <td>-100%</td> </tr> <tr> <td>Salt</td> <td>0.023 g</td> <td>0.023 g</td> <td>-56%</td> </tr> <tr> <td>Sodium</td> <td>0.009 g</td> <td>0.009 g</td> <td>-56%</td> </tr> <tr> <td>Alcohol</td> <td>0 % vol</td> <td>0 % vol</td> <td>0 % vol</td> </tr> <tr> <td>Potassium</td> <td>220 mg</td> <td>220 mg</td> <td>+22%</td> </tr> <tr> <td>Nutrition score - France</td> <td>6</td> <td>6</td> <td>+97%</td> </tr> <tr> <td>Nutri-Score</td> <td>D</td> <td>D</td> <td>D</td> </tr> </tbody> </table> | Nutrition facts | As sold for 100 g / 100 ml | As sold per serving (100 ml) | Coconut waters | Energy (kJ) | 85 kJ | 85 kJ | -10% | Energy (kcal) | ? | ? | 20 kcal | Energy | 85 kJ (20 kcal) | 85 kJ (20 kcal) | -2% | Fat | 0 g | 0 g | -100% | - Saturated fat | 0 g | 0 g | -100% | Carbohydrates | 5 g | 5 g | +0% | - Sugars | 3.8 g | 3.8 g | -6% | Fibers | 0 g | 0 g | -100% | Proteins | 0 g | 0 g | -100% | Salt | 0.023 g | 0.023 g | -56% | Sodium | 0.009 g | 0.009 g | -56% | Alcohol | 0 % vol | 0 % vol | 0 % vol | Potassium | 220 mg | 220 mg | +22% | Nutrition score - France | 6 | 6 | +97% | Nutri-Score | D | D | D |
| Nutrition facts | As sold for 100 g / 100 ml | As sold per serving (100 ml) | Coconut waters | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy (kJ) | 85 kJ | 85 kJ | -10% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy (kcal) | ? | ? | 20 kcal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy | 85 kJ (20 kcal) | 85 kJ (20 kcal) | -2% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fat | 0 g | 0 g | -100% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Saturated fat | 0 g | 0 g | -100% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Carbohydrates | 5 g | 5 g | +0% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Sugars | 3.8 g | 3.8 g | -6% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fibers | 0 g | 0 g | -100% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Proteins | 0 g | 0 g | -100% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Salt | 0.023 g | 0.023 g | -56% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Sodium | 0.009 g | 0.009 g | -56% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Alcohol | 0 % vol | 0 % vol | 0 % vol | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Potassium | 220 mg | 220 mg | +22% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutrition score - France | 6 | 6 | +97% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutri-Score | D | D | D | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Bolacha de cereais enriquecida com ferro, magnésio, vitaminas B6 e B9. - Mondelez - 250 g</p> <p>Barcode: 762216050708 (EAN / EAN-13)</p> <p>Quantity: 250 g</p> <p>Brand: Mondelez, Belvita</p> <p>Categories: pt Cerealizche</p> <p>Origin of ingredients: Portugal</p> <p>Manufacturing or processing places: P2810-171, Amadora, Portugal</p> <p>Link to the product page on the official site of the producer: http://www.belvita.com/pt/originais/bolacha</p> <p>Brand: Bolacha</p> <p>Countries where sold: France, Poland, Portugal</p> | <p>NUTRI-SCORE</p> <p>A B C D E</p> <p>Details of the calculation of the Nutri-Score »</p> <p>Nutrient levels for 100 g</p> <ul style="list-style-type: none"> 12 g Fat in moderate quantity 1 g Saturated fat in low quantity 18 g Sugars in high quantity 0.7 g Salt in moderate quantity <p>Serving size: 50g</p> | <table border="1"> <thead> <tr> <th>Nutrition facts</th> <th>As sold for 100 g / 100 ml</th> <th>As sold per serving (50g)</th> </tr> </thead> <tbody> <tr> <td>Energy (kJ)</td> <td>?</td> <td>?</td> </tr> <tr> <td>Energy (kcal)</td> <td>385 kcal</td> <td>192 kcal</td> </tr> <tr> <td>Energy</td> <td>1,611 kJ (385 kcal)</td> <td>806 kJ (192 kcal)</td> </tr> <tr> <td>Fat</td> <td>12 g</td> <td>6 g</td> </tr> <tr> <td>- Saturated fat</td> <td>1 g</td> <td>0.5 g</td> </tr> <tr> <td>Carbohydrates</td> <td>63 g</td> <td>31.5 g</td> </tr> <tr> <td>- Sugars</td> <td>18 g</td> <td>9 g</td> </tr> <tr> <td>- Starch</td> <td>39 g</td> <td>19.5 g</td> </tr> <tr> <td>Fibers</td> <td>6.7 g</td> <td>3.35 g</td> </tr> <tr> <td>Proteins</td> <td>5.9 g</td> <td>2.95 g</td> </tr> <tr> <td>Salt</td> <td>0.7 g</td> <td>0.35 g</td> </tr> <tr> <td>Sodium</td> <td>0.28 g</td> <td>0.14 g</td> </tr> <tr> <td>Nutrition score - France</td> <td>2</td> <td>2</td> </tr> <tr> <td>Nutri-Score</td> <td>B</td> <td>B</td> </tr> </tbody> </table> | Nutrition facts | As sold for 100 g / 100 ml | As sold per serving (50g) | Energy (kJ) | ? | ? | Energy (kcal) | 385 kcal | 192 kcal | Energy | 1,611 kJ (385 kcal) | 806 kJ (192 kcal) | Fat | 12 g | 6 g | - Saturated fat | 1 g | 0.5 g | Carbohydrates | 63 g | 31.5 g | - Sugars | 18 g | 9 g | - Starch | 39 g | 19.5 g | Fibers | 6.7 g | 3.35 g | Proteins | 5.9 g | 2.95 g | Salt | 0.7 g | 0.35 g | Sodium | 0.28 g | 0.14 g | Nutrition score - France | 2 | 2 | Nutri-Score | B | B | | | | | | | | | | | | | | | | | | | |
| Nutrition facts | As sold for 100 g / 100 ml | As sold per serving (50g) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy (kJ) | ? | ? | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy (kcal) | 385 kcal | 192 kcal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy | 1,611 kJ (385 kcal) | 806 kJ (192 kcal) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fat | 12 g | 6 g | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Saturated fat | 1 g | 0.5 g | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Carbohydrates | 63 g | 31.5 g | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Sugars | 18 g | 9 g | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Starch | 39 g | 19.5 g | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fibers | 6.7 g | 3.35 g | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Proteins | 5.9 g | 2.95 g | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Salt | 0.7 g | 0.35 g | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Sodium | 0.28 g | 0.14 g | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutrition score - France | 2 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutri-Score | B | B | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Manteiga Light, Neutral - Matinal</p> <p>Barcode: 560722493812 (EAN / EAN-13)</p> <p>Quantity: 200 g</p> <p>Brand: Matinal</p> <p>Categories: Fats</p> | <p>Nutrition facts</p> <p>NutriScore color nutrition grade D</p> <p>NUTRI-SCORE</p> <p>A B C D E</p> <p>Warning: the amounts of fiber and of fruits, vegetables and nuts are not specified, their possible positive contribution to the grade could not be taken into account.</p> <p>Details of the calculation of the Nutri-Score »</p> <p>Nutrient levels for 100 g</p> <ul style="list-style-type: none"> 41 g Fat in high quantity 26 g Saturated fat in high quantity 0 g Sugars in low quantity 1.2 g Salt in moderate quantity <p>Comparison to average values of products in the same category:</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Fats (17047 products) <p>• % of difference ○ value for 100 g / 100 ml</p> <p>– Please note: for each nutriment, the average is computed for products for which the nutriment quantity is known, not on all products of the category.</p> | <table border="1"> <thead> <tr> <th>Nutrition facts</th> <th>As sold for 100 g / 100 ml</th> <th>Fats</th> </tr> </thead> <tbody> <tr> <td>Energy (kJ)</td> <td>?</td> <td>3,170 kJ</td> </tr> <tr> <td>Energy (kcal)</td> <td>384 kcal</td> <td>-49%</td> </tr> <tr> <td>Energy</td> <td>1,607 kJ (384 kcal)</td> <td>-49%</td> </tr> <tr> <td>Fat</td> <td>41 g</td> <td>-50%</td> </tr> <tr> <td>- Saturated fat</td> <td>26 g</td> <td>+31%</td> </tr> <tr> <td>Carbohydrates</td> <td>0 g</td> <td>-100%</td> </tr> <tr> <td>- Sugars</td> <td>0 g</td> <td>-100%</td> </tr> <tr> <td>Proteins</td> <td>1.9 g</td> <td>+17%</td> </tr> <tr> <td>Salt</td> <td>1.2 g</td> <td>+351%</td> </tr> <tr> <td>Sodium</td> <td>0.48 g</td> <td>+351%</td> </tr> <tr> <td>Nutrition score - France</td> <td>18</td> <td>+61%</td> </tr> <tr> <td>Nutri-Score</td> <td>D</td> <td>D</td> </tr> </tbody> </table> | Nutrition facts | As sold for 100 g / 100 ml | Fats | Energy (kJ) | ? | 3,170 kJ | Energy (kcal) | 384 kcal | -49% | Energy | 1,607 kJ (384 kcal) | -49% | Fat | 41 g | -50% | - Saturated fat | 26 g | +31% | Carbohydrates | 0 g | -100% | - Sugars | 0 g | -100% | Proteins | 1.9 g | +17% | Salt | 1.2 g | +351% | Sodium | 0.48 g | +351% | Nutrition score - France | 18 | +61% | Nutri-Score | D | D | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutrition facts | As sold for 100 g / 100 ml | Fats | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy (kJ) | ? | 3,170 kJ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy (kcal) | 384 kcal | -49% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Energy | 1,607 kJ (384 kcal) | -49% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fat | 41 g | -50% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Saturated fat | 26 g | +31% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Carbohydrates | 0 g | -100% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Sugars | 0 g | -100% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Proteins | 1.9 g | +17% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Salt | 1.2 g | +351% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Sodium | 0.48 g | +351% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutrition score - France | 18 | +61% | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nutri-Score | D | D | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Fonte: O autor, 2019

4.4 Considerações sobre o Experimento

O projeto de desenvolvimento do aplicativo Lista Saudável foi instanciado no contexto da nutrição, mais precisamente, na seleção de alimentos que poderiam ajudar um indivíduo a manter uma dieta saudável.

Os testes com o *Dataset* contendo produtos brasileiros mostraram ser necessária a utilização de uma coluna a mais como dado categórico para melhorar os resultados, assim, totalizando 5 colunas ("product name", "nutrition grade fr", "categories" e "main category" e "nutrition score fr 100g code"). Também foi observado que, utilizando 70% dos dados do dataset para treino e 30% para o teste, conseguimos o melhor resultado com 65% de acurácia.

CONCLUSÕES E TRABALHOS FUTUROS

Em geral os produtos no Brasil não possuem uma classificação com base nos seus valores nutricionais. Portanto, especificamos aqui um sistema de recomendação baseado no valor nutricional de alimentos para o domínio de supermercado e/ou nutrição, com o propósito de consumidores comprarem os melhores produtos alimentícios. Ainda que os sistemas de recomendação em relação aos produtos de lazer (por ex. filmes, livros e música) tenham sido alvo de estudos, esse novo domínio oferece as vantagens de abordar o problema de oferecer boas recomendações com alto desempenho computacional e foco na qualidade dos produtos, além de evitar as más recomendações, que podem gerar decepção aos usuários.

Os resultados preliminares deste estudo e os algoritmos desenvolvidos, servirão de base para iniciar um experimento voltado ao desenvolvimento de um aplicativo para dispositivos móveis multiplataforma, para a criação de listas de supermercado com maior nível de especificidade.

Esperamos, dessa forma, contribuir para a prevenção de doenças coronarianas, sabendo que existe uma relação entre os hábitos alimentares, que contribuem para o surgimento de Síndromes Metabólicas, principalmente, na população diabética e hipertensa.

Os principais resultados deste trabalho foram os seguintes:

- Para realizar testes com o *dataset* brasileiro contendo dados sobre *nutriscore* faltante. Foi necessário a utilização de algoritmos de *machine learning* para seu treinamento, a partir do *dataset* contendo produtos franceses, onde o campo com *nutriscore* está completamente preenchido.
- Para a criação do algoritmo recomendador, utilizou-se o conceito de similaridade com os métodos da *CountVectorizer* e o *TF-IDFVectorizer* do pacote *Scikit-learn* da linguagem de programação *python*. Foi observado que ambos os métodos mostraram o mesmo resultado.
- Para possibilitar o uso do aplicativo por um maior número de pessoas, esperamos

refinar o aplicativo para possibilitar seu uso dentro dos supermercados, bastando apenas que o usuário possua internet para acessar o sistema de recomendação.

Ameaças a validade

A avaliação da confiabilidade do instrumento aplicado, se limita aos testes e treinamentos realizados com o *dataset*, que demonstraram um valor de acurácia de 65%. A ameaça a validade consiste em rotular nutriscore para produtos, que não correspondem aos seus efetivos valores. Os resultados deste estudo referem-se apenas aos números de respostas alcançadas com os algoritmos de *Machine learning* escolhidos.

Trabalhos futuros

Como trabalhos futuros, pretende-se criar uma lista de produtos brasileiros mais saudável utilizando um dispositivo móvel. Entretanto, ainda temos diversas questões em aberto. Podemos destacar algumas delas, a seguir.

- Estudar e desenvolver um módulo utilizando *machine learning* para reconhecimento de imagem. Neste caso, o usuário poderá obter uma foto do carrinho de compras e o sistema sugere uma outra lista com produtos mais saudáveis, facilitando a criação da lista de supermercado;
- Estudar uma variação de dietas para a criação do perfil de usuários por tipo específico de doença. Por exemplo, o usuário poderá selecionar a doença e o sistema listar apenas produtos que são adequados para ele;
- Estudar e implementar um módulo que, utilizando leitura de código de barras, possa mostrar as informações nutricionais do produto com a foto, para demonstrar os produtos similares a ele e/ou mais saudáveis;
- Estudar e desenvolver um módulo que utilizando *machine learning* para sugerir produtos mais saudáveis baseado no perfil e no histórico das listas de compras realizadas anteriormente.

REFERÊNCIAS

- AIOLFI, S. B. S. Using mobile applications: A model of technology adoption in the grocery setting. *International Journal of Business and Management*, INFORMS, p. 1–50, 2019.
- ALSEDRAH, M. *Artificial Intelligence*. 2017. 3–4 p.
- ANACONDA. *Anaconda*. 2019. Disponível em: <<https://anaconda.org/>>. Acesso em: 02 janeiro 2019.
- ANDONI, A.; INDYK, P.; RAZENSHTEYN, I. Approximate nearest neighbor search in high dimensions. *arXiv preprint arXiv:1806.09823*, World Scientific, v. 7, 2018.
- ANTON, O. et al. Vege application! using mobile application to promote vegetarian food. In: IEEE. *2018 International Conference on Applied Engineering (ICAE)*. [S.l.], 2018. p. 1–6.
- ANY-LIST. *AnyList*. 2019. Disponível em: <<https://www.anylist.com/>>. Acesso em: 31 agosto 2019.
- AWAD, M.; KHANNA, R. *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. [S.l.]: Apress, 2015.
- BANIK, R. *Hands-On Recommendation Systems with Python*. [S.l.]: Packt Publishing Ltd, 2018.
- BOOSTING-CLASSIFIER. *BoostingClassifier*. 2019. Disponível em: <<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html>>. Acesso em: 02 janeiro 2019.
- BURKE, R. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, Springer, v. 12, n. 4, p. 331–370, 2002.
- BUSCAPE. *Buscape*. 2019. Disponível em: <<https://www.buscapede.com.br/conheca-o-buscapede>>. Acesso em: 31 agosto 2019.
- BUY-ME-A-PIE. *Buy Me A Pie*. 2019. Disponível em: <<https://buymeapie.com/pt>>. Acesso em: 31 agosto 2019.
- CANTIERI, G. N.; BUENO, C. A. M.; MARTINEZ-ÁVILA, D. Efeitos do treinamento resistido em adultos com síndrome metabólica effects of resistance training in adults with metabolic syndrome. *Revista Brasileira de Fisiologia do Exercício*, v. 17, n. 3, p. 185–194, 2018.
- CATALTEPE, Z.; ULUYAĞMUR, M.; TAYFUR, E. Feature selection for movie recommendation. *Turkish Journal of Electrical Engineering & Computer Sciences*, The Scientific and Technological Research Council of Turkey, v. 24, n. 3, p. 833–848, 2016.

- CHEN, T. et al. Content recommendation system based on private dynamic user profile. In: IEEE. *Machine Learning and Cybernetics, 2007 International Conference on*. [S.l.], 2007. v. 4, p. 2112–2118.
- CLUBE-EXTRA. *Clube Extra*. 2019. Disponível em: <<https://www.clubeextra.com.br>>. Acesso em: 31 agosto 2019.
- COSINE-SIMILARITY. *Cosine Similarity*. 2019. Disponível em: <<https://www.sciencedirect.com/topics/computer-science/cosine-similarity>>. Acesso em: 16 dezembro 2019.
- COUNT-VECTORIZER. *Count Vectorizer*. 2019. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html>. Acesso em: 02 janeiro 2019.
- DANGETI, P. *Statistics for Machine Learning*. [S.l.]: Packt Publishing Ltd, 2017.
- DAVIS, F. D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, JSTOR, p. 319–340, 1989.
- DAVIS, F. D.; BAGOZZI, R. P.; WARSHAW, P. R. User acceptance of computer technology: a comparison of two theoretical models. *Management science*, INFORMS, v. 35, n. 8, p. 982–1003, 1989.
- DEERWESTER, S. et al. Indexing by latent semantic analysis. *Journal of the American society for information science*, Wiley Online Library, v. 41, n. 6, p. 391–407, 1990.
- DESROTULANDO. *Desrotulando*. 2019. Disponível em: <<https://desrotulando.com/>>. Acesso em: 31 agosto 2019.
- FACILISTA. *Facilista*. 2019. Disponível em: <<https://www.bonde.com.br/blogs/enter-x/facilista-o-app-de-sua-lista-de-compras-de-supermercado--340014.html>>. Acesso em: 31 agosto 2019.
- FIONDA, V.; PIRRÓ, G. Triple2vec: Learning triple embeddings from knowledge graphs. *arXiv preprint arXiv:1905.11691*, 2019.
- FRIEDMAN, J. et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, Institute of Mathematical Statistics, v. 28, n. 2, p. 337–407, 2000.
- GOLUB, G. H.; REINSCH, C. Singular value decomposition and least squares solutions. *Numerische mathematik*, Springer, v. 14, n. 5, p. 403–420, 1970.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016.
- GORAKALA, S. K. *Building Recommendation Engines*. [S.l.]: Packt Publishing Ltd, 2016.
- HAJJDIAB, H. et al. A food wastage reduction mobile application. In: IEEE. *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. [S.l.], 2018. p. 152–157.

HEART, L. N.; INSTITUTE, B. et al. Your guide to lowering your cholesterol with tlc (therapeutic lifestyle changes). *National Institutes of Health, US Department of Health and Human Services, Bethesda, MD*, 2006.

HOME-REFILL. *Home Refill*. 2019. Disponível em: <https://play.google.com/store/apps/details?id=br.com.homerefill.homerefill&hl=pt_BR>. Acesso em: 31 agosto 2019.

Hugo Honda, Matheus Facure, Peng Yaohao. *Os Três Tipos de Aprendizado de Máquina*. 2017. Disponível em: <<https://lamfo-unb.github.io/2017/07/27/tres-tipos-am/>>. Acesso em: 08 março 2020.

ILISTOUCH. *IListouch*. 2019. Disponível em: <<https://www.techtudo.com.br/tudo-sobre/ilist-touch.html>>. Acesso em: 31 agosto 2019.

ISINKAYE, F.; FOLAJIMI, Y.; OJOKOH, B. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, Elsevier, v. 16, n. 3, p. 261–273, 2015.

JUPYTER. *Jupyter*. 2019. Disponível em: <<https://jupyter.org/>>. Acesso em: 02 janeiro 2019.

KERNELS. *KERNELS*. 2019. Disponível em: <<https://scikit-learn.org/stable/modules/metrics.html#linear-kernel>>. Acesso em: 02 janeiro 2019.

KIM, D. et al. Multi-co-training for document classification using various document representations: Tf-idf, lda, and doc2vec. *Information Sciences*, Elsevier, v. 477, p. 15–29, 2019.

KODINARIYA, T. M.; MAKWANA, P. R. Review on determining number of cluster in k-means clustering. *International Journal*, v. 1, n. 6, p. 90–95, 2013.

KÜHL, N. et al. Machine learning in artificial intelligence: Towards a common understanding. In: *Proceedings of the 52nd Hawaii International Conference on System Sciences*. [S.l.: s.n.], 2019.

KUMARI, M.; JAIN, A.; BHATIA, A. Synonyms based term weighting scheme: An extension to tf. idf. *Procedia Computer Science*, Elsevier, v. 89, p. 555–561, 2016.

KUNKEL, J.; FARRELL, P. Predicting companies mentioned in news articles, a comparison of two approaches: Latent dirichlet allocation with k-nearest neighbor versus bag of words with k-nearest neighbor. 2018.

LAMFO. *Lamfo-UnB*. 2019. Disponível em: <<https://lamfo-unb.github.io/2017/07/27/tres-tipos-am>>. Acesso em: 13 setembro 2019.

LEIPOLD, N. et al. Nutrilize a personalized nutrition recommender system: an enable study. *HealthRecSys' 18*, 2018.

LISTONIC. *Listonic*. 2019. Disponível em: <https://play.google.com/store/apps/details?id=com.l&hl=pt_BR>. Acesso em: 31 agosto 2019.

LUGER, G. *Inteligência Artificial*. [S.l.]: Pearson, 2014.

- MACKENZIE, A. Personalization and probabilities: Impersonal propensities in online grocery shopping. *Big Data & Society*, SAGE Publications Sage UK: London, England, v. 5, n. 1, p. 2053951718778310, 2018.
- MANTHA, A. et al. A large-scale deep architecture for personalized grocery basket recommendations. *arXiv preprint arXiv:1910.12757*, 2019.
- METEREN, R. V.; SOMEREN, M. V. Using content-based filtering for recommendation. In: *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*. [S.l.: s.n.], 2000. p. 47–56.
- MEU-CARRINHO. *Meu Carrinho*. 2019. Disponível em: <<https://www.techtudo.com.br/tudo-sobre/meucarrinho-lista-de-compras.html>>. Acesso em: 31 agosto 2019.
- MURPHY, K. P. *Machine learning: a probabilistic perspective*. [S.l.]: MIT press, 2012.
- OKFN. *OKFN*. 2019. Disponível em: <<https://okfn.org/network/france/>>. Acesso em: 02 janeiro 2019.
- OLKIN, G. C. S. F. I. Springer texts in statistics. Springer, 2002.
- OMS. *OMS*. 2019. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>>. Acesso em: 02 janeiro 2019.
- OPENFOODFACTS. *Open Food Facts*. 2019. Disponível em: <<https://world.openfoodfacts.org/>>. Acesso em: 02 janeiro 2019.
- PAODEACUCAR. *Pão de açúcar*. 2019. Disponível em: <<https://www.paodeacucar.com/>>. Acesso em: 31 agosto 2019.
- PAZZANI, M. J. A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review*, Springer, v. 13, n. 5-6, p. 393–408, 1999.
- PAZZANI, M. J.; BILLSUS, D. Content-based recommendation systems. In: *The adaptive web*. [S.l.]: Springer, 2007. p. 325–341.
- PNNS. *PNNS*. 2019. Disponível em: <<http://www.mangerbouger.fr/PNNS/>>. Acesso em: 02 janeiro 2019.
- PYTHON. *Python*. 2019. Disponível em: <<https://www.python.org/>>. Acesso em: 02 janeiro 2019.
- QQFALTA. *QQFalta*. 2019. Disponível em: <<http://www.qqfalta.com.br/Sobre>>. Acesso em: 31 agosto 2019.
- RICCI, F.; ROKACH, L.; SHAPIRA, B. Recommender systems: introduction and challenges. In: *Recommender systems handbook*. [S.l.]: Springer, 2015. p. 1–34.
- SBC. *SBC*. 2019. Disponível em: <<https://www.cardiol.br/>>. Acesso em: 02 janeiro 2019.
- SCHAFER, J. B. et al. Collaborative filtering recommender systems. In: *The adaptive web*. [S.l.]: Springer, 2007. p. 291–324.

- SCHWAB, J. A. Multinomial logistic regression: Basic relationships and complete problems. *Austin, Texas: University of Texas*, 2002.
- SCIKIT-LEARN. *Scikit-Learn*. 2019. Disponível em: <<https://scikit-learn.org/stable/>>. Acesso em: 02 janeiro 2019.
- SHOPPER. *Shopper*. 2019. Disponível em: <<https://shopper.com.br/>>. Acesso em: 31 agosto 2019.
- SHOPPING-LIST. *ShoppingList*. 2019. Disponível em: <https://play.google.com/store/apps/details?id=ru.grocerylist.android&hl=en_US>. Acesso em: 31 agosto 2019.
- SHOPWELL. *Shopwell*. 2019. Disponível em: <<https://www.innit.com/shopwell/>>. Acesso em: 31 agosto 2019.
- SIELIS, G. A.; TZANAVARI, A.; PAPADOPOULOS, G. A. Recommender systems review of types, techniques, and applications. In: *Encyclopedia of Information Science and Technology, Third Edition*. [S.l.]: IGI Global, 2015. p. 7260–7270.
- SIRI. *Siri*. 2019. Disponível em: <<https://www.apple.com/br/siri/>>. Acesso em: 31 agosto 2019.
- SOFT-LIST. *SoftList*. 2019. Disponível em: <https://play.google.com/store/apps/details?id=br.com.ridsoftware.shoppinglist&hl=pt_BR>. Acesso em: 31 agosto 2019.
- STAT-SOFT. *Statsoft*. 2019. Disponível em: <<http://www.statsoft.com/Textbook/Boosting-Trees-Regression-Classification>>. Acesso em: 02 janeiro 2019.
- SU, X. et al. *Introduction to Big Data*. 2012.
- TF-IDF VECTORIZER. *TF-IDF Vectorizer*. 2019. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html>. Acesso em: 02 janeiro 2019.
- TORRES, J.; SANTOS, S. D. L. Malicious pdf documents detection using machine learning techniques-a practical approach with cloud computing applications. In: *ICISSP*. [S.l.: s.n.], 2018. p. 337–344.
- TRATAMENTO, D. E. I diretriz brasileira de diagnóstico e tratamento da síndrome metabólica. *Arquivos Brasileiros de Cardiologia*, SciELO Brasil, v. 84, n. Suplemento I, 2005.
- VERAKIS. *Verakis*. 2019. Disponível em: <<http://verakis.over-blog.com/2018/03/programa-nacional-nutricao-e-saude-franca.html>>. Acesso em: 29 abril 2019.
- VILLEGAS, J.; SAITO, S. Assisting system for grocery shopping navigation and product recommendation. In: IEEE. *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*. [S.l.], 2017. p. 1–4.
- ZUVA, T. et al. Image content in location-based shopping recommender systems for mobile users. *Advanced Computing*, Academy & Industry Research Collaboration Center (AIRCC), v. 3, n. 4, p. 1, 2012.

GLOSSÁRIO

| | |
|--------------|---|
| Acurácia | É a métrica para medir a precisão mais usada para avaliar o desempenho de um modelo de classificação. É a razão entre o número de previsões corretas e o número total de previsões feitas pelo modelo. |
| Algoritmo | Um algoritmo é qualquer procedimento computacional bem definido que toma algum valor ou conjunto de valores como entrada e produz algum valor ou conjunto de valores como saída. |
| Bag of Words | No processamento de texto, as palavras do texto representam recursos discretos e categóricos. Como codificamos esses dados de uma maneira que está pronta para ser usada pelos algoritmos? O mapeamento de dados textuais para vetores com valor real é chamado de extração de características. Uma das técnicas mais simples para representar numericamente o texto é o Bag of Words. |
| Big data | É um grande volume de informações, alta velocidade e / ou ativos de informações de alta variedade que exigem formas inovadoras e econômicas de processamento de informações que permitem uma melhor percepção, tomada de decisões e automação de processos. |
| Data mining | Também chamada de Mineração de dados pode ser definida como o processo de decifrar percepções significativas das existentes bases de dados e análise de resultados para consumo por usuários empresariais. Analisando dados de várias fontes e resumindo-o em informações significativas e insights é que parte da descoberta do conhecimento estatístico que ajuda não apenas usuários de negócios, mas também múltiplas comunidades como analistas estatísticos, consultores e cientistas de dados. A maior parte do tempo, o processo de descoberta de conhecimento de bancos de dados é inesperado e os resultados podem ser interpretado de várias maneiras. |
| Dataset | Um conjunto de dados que são publicados, mantidos ou agregados por um único provedor. |
| MAE | É técnica de avaliação popular para o modelo de mineração de dados. Essa métrica de avaliação é muito semelhante ao RMSE, ela é calculada como erro médio entre valores previstos e reais. |
| MSE | Nessa configuração de regressão é a medida mais usada. O MSE será pequeno se as respostas previstas estiverem muito próximas das respostas verdadeiras, e será grande se, para algumas das observações, as respostas previstas e verdadeiras diferem substancialmente. O MSE é calculado usando os dados de treinamento que foram usados para ajustar o modelo representado. |

| | |
|-------------|--|
| Matriz | É um arranjo bidimensional ou multidimensional de alocação estática e sequencial. A matriz é uma estrutura de dados que necessita de um índice para referenciar a linha e outro para referenciar a coluna para que seus elementos sejam endereçados. Da mesma forma que um vetor, uma matriz é definida com um tamanho fixo, todos os elementos são do mesmo tipo, cada célula contém somente um valor e os tamanhos dos valores são os mesmos. |
| Recall | É a proporção do número de casos positivos que foram identificados para todos os casos positivos presentes no conjunto de dados. |
| RMSE | É uma métrica amplamente usada para avaliar o desempenho dos regressores. |
| Overfitting | É um grande problema em redes neurais. Isso é especialmente verdadeiro em redes modernas, que geralmente têm um grande número de pesos e vieses. Para treinar de forma eficaz, precisamos de uma maneira de detectar quando o overfitting está acontecendo. E precisamos aplicar técnicas para reduzir os efeitos do overfitting (por todo esse trabalho e conhecimento necessário, Cientistas de Dados devem ser muito bem remunerados). |
| Vetor | É uma estrutura de dados linear que necessita de somente um índice para que seus elementos sejam endereçados. É utilizado para armazenar uma lista de valores do mesmo tipo, ou seja, o tipo vetor permite armazenar mais de um valor em uma mesma variável. Um dado vetor é definido como tendo um número fixo de células idênticas (seu conteúdo é dividido em posições). Cada célula armazena um e somente um dos valores de dados do vetor. Cada uma das células de um vetor possui seu próprio endereço, ou índice, através do qual pode ser referenciada. Nessa estrutura todos os elementos são do mesmo tipo, e cada um pode receber um valor diferente. |

APÊNDICE A - Algoritmo de Similaridade de produto baseado em Nutriscore

O Algoritmo 1 espera receber como parâmetros de entrada para utilizar o conceito de similaridade as colunas: "product name", "nome do produto", "nutrition grade fr", "categories" e "main category". Como a coluna de Nutriscore (nutrition grade fr) do Dataset brasileiro foi preenchida pelos valores resultante da predição, obtidos pelos resultados do treinamento da classificação. A criação do chamado Bag of Words citado na seção 1.3, que será formado por um vetor contendo o nome do produto, e assim aplicar o algoritmo com a chamada dos métodos de vetorização: CountVectorizer e o TF-IDFVectorizer.

Algoritmo 1 – Similaridade de produto baseado em Nutriscore.

```

1: função BUSCAPRODUTOSIMILAR
   ENTRADAS
2:   np: nome do produto
   idx: índice vetor do nome do produto
   vet-np: vetor de produtos
   indices: vetor com o índice do nome do produto
   sim-scores: vetor com o índice do nome do produto
   produto-índice: vetor com o índice do produto
   cosine-sim: vetor de similares de todos os produtos
3: SAÍDAS
4:   vet-np: produtos similares com menor Nutriscore que o np

5: OBSERVAÇÕES, RESTRIÇÕES, REQUISITOS
6:   Excluir nome do produto repetido do vetor de produtos - vet-np.

7: 1. declarar np caractere
8: 2. declarar i numéricos
9: 3. declarar vet – np, cosine – sim vetor
10: 4. cosine – sim  $\leftarrow$  vet – np
11: 5. idx  $\leftarrow$  indices[np]
12: 6. sim – scores[10]  $\leftarrow$  cosine – sim[idx]
13:
14: para sim – scores de 1 até sim – scores–1 faça
15:   se vet-np[sim – scores].nutrescore < np.nutrescore então
16:     vet – np  $\leftarrow$  np.nutrescore
17:     vet – np[sim-scores]
18:   senão
19:     np

```
