

MELHORANDO A EXPERIÊNCIA DE COMPRA EM MERCADOS FÍSICOS  
COM O USO DE ALGORITMOS DE RECOMENDAÇÃO  
*Improving the on-site shopping experience of markets with the use of recommendation  
algorithms*

Camila Leão<sup>1</sup>

Jhonas Victor Mutton<sup>1</sup>

Fabio Andrijauskas<sup>2</sup>

Universidade São Francisco – *Campus Itatiba/SP*

[camila.leao@mail.usf.edu.br](mailto:camila.leao@mail.usf.edu.br) | [jhonas.mutton@mail.usf.edu.br](mailto:jhonas.mutton@mail.usf.edu.br) |

[fabio.andrijauskas@usf.edu.br](mailto:fabio.andrijauskas@usf.edu.br)

<sup>1</sup>Alunos do Curso de Engenharia da Computação, Universidade São Francisco;  
Campus Itatiba - SP

<sup>2</sup>Professor Orientador Fabio Andrijauskas, Curso de Engenharia da Computação,  
Universidade São Francisco; Campus Itatiba - SP.

**RESUMO.** Com a pandemia do vírus SARs-CoV-2 a forma de consumo mudou rapidamente, o que levou o consumidor a interagir ainda mais com o meio digital, e por este motivo os comércios de venda física precisam se conectar e gerar a experiência ao cliente. Este artigo tem como objetivo arquitetar e propor melhorias na experiência de compra de clientes em supermercados físicos através da interação dos usuários em plataforma mobile. Isso se dará através do gerenciamento de recomendação utilizando *field aware factorization machines (FFM)* e uma proposta básica de aplicativo que traga usabilidade ao usuário. Para a elaboração do trabalho, foi utilizada como base a metodologia de desenvolvimento de *software* Cascata. Foram criados protótipos básicos de tela, utilizando o programa Figma, de maneira a demonstrar como seriam as personalizações com base nas preferências do usuário, no aplicativo. Foi desenvolvido o menu de navegação inferior e a tela principal do aplicativo, possibilitando ao usuário visualizar promoções, *banners* de campanhas e uma visualização básica de itens em geral. Foi alcançado o desenvolvimento de um *back-end* funcional, onde é possível realizar transações nas principais entidades do sistema, utilizando *web APIs*. O modelo de recomendação alcançou um nível satisfatório de precisão com dados artificiais, no entanto, ainda não é possível validar a precisão e funcionamento das recomendações em um cenário real. Os próximos passos a serem tomados devem ser direcionados para a integração de todos esses módulos e completa utilização dos recursos desenvolvidos.

**PALAVRAS-CHAVE:** sistema de recomendação; *collaborative filtering*; *content-based filtering* ; supermercado físico; consumo presencial; inteligência artificial; experiência do cliente; *field aware factorization machines*.

**ABSTRACT.** With the SARs-CoV-2 pandemic quickly changing the way people consume and it makes the customer interact even more with the digital way, for this reason, physical shops must be connected to generate a better customer experience. This article has as its main objective to build and propose improvements for the customer buying experience at supermarkets through user interaction on mobile devices. It will be given through recommendation management using field aware factorization machines (FFM) and a simple proposal of an app that brings user usability. For the project design was used software development method called “Cascata”. Basic screen prototypes were created by using the program “Figma”, in order to demonstrate the customizations based on user preferences on the app. The lower navigation menu and the main application screen were developed,

allowing the user to view promotions, campaign banners and basic view of items in general. the development of a functional back-end was achieved, where it's possible to carry out transactions in the main entities of the system, using web APIs. The recommendation model reached a satisfactory level of accuracy with artificial data, however, it's still not possible to validate the accuracy and functioning of the recommendations in a real scenario. The next steps to be taken must be directed towards the integration of all these modules and full use of the developed resources.

**KEYWORDS:** recommendation system; collaborative filtering; content-based filtering; hybrid recommendation; supermarket; in person consumption; artificial intelligence; customer experience; field aware factorization machines.

## INTRODUÇÃO

O uso de aplicativos em dispositivos móveis tem aumentado a cada ano e segundo uma pesquisa feita pela AppsFlyer (2020), o Brasil teve um aumento de 55% no número de instalações nos últimos 2 anos. Isto demonstra que o consumidor está cada vez mais conectado e integrado às novas tecnologias que o mercado oferece. Os dados também mostram que 78% dos brasileiros preferem fazer pagamentos pelo celular, indicando que cada vez mais o uso de dispositivos móveis estará presente no cotidiano e em sua rotina de compras.

Com a pandemia do vírus SARs-CoV-2 em 2020, os dados demonstraram que houve um crescimento no uso à medida que os casos aumentaram no Brasil. A pandemia também gerou ao consumidor algumas mudanças de comportamentos, segundo uma pesquisa feita pela VEJA INSIGHTS e EY Parthenon (2020), o brasileiro agora tem uma preocupação maior pela economia, tanto do país quanto a doméstica, fazendo com que o consumidor passe a comprar mais produtos essenciais e menos superfluidades. O levantamento também indica que houve mudanças de rotinas devido a pandemia e demonstra que o consumidor as medidas que as restrições diminuem, se sente muito desconfortável em frequentar alguns ambientes menos essenciais, mas 70% ainda se sente confortável ou indiferente ao frequentar supermercados. Estas alterações de hábitos continuarão mesmo após a pandemia e fazem com que os supermercadistas necessitem se estruturar e estar um passo à frente quanto às tendências de consumo e de comportamento para atender ao novo consumidor. VABENE (2021). O conceito *Omnichannel* deixou de ser tendência e passou a ser realidade segundo VABENE (2021), em que integra loja física, virtual e compradores, sem que o usuário veja diferença entre o mundo *online* e *offline*, com o foco voltado ao consumidor. SEBRAE (2017)

Como visto, o uso de aplicativos tem se tornado maior e mais viável, ao mesmo tempo que o brasileiro, em meio a uma pandemia, ainda se sente confortável a frequentar supermercados e busca mais economia e produtos essenciais. Levantamos, então, um questionamento: como poderíamos tornar a ida ao supermercado mais confortável, fácil e mais econômica?

O objetivo deste trabalho é unir a tecnologia em telefones móveis com experiência de compra presencial, trazendo ganhos de comodidade e economia para o consumidor, principalmente em um cenário delicado, como uma pandemia. E, para os mercados, o ganho de grande volume de dados dos consumidores, de maneira a possibilitar mais eficiência logística, econômica e fidelização dos clientes.

Para isso, a proposta é arquitetar e propor melhorias na experiência de compra nos mercados físicos através da interação do usuário com uma plataforma *mobile*, por meio de gerenciamento de recomendação utilizando *field aware factorization machines (FFM)* e uma proposta básica de aplicativo que traga usabilidade ao usuário, permitindo ao consumidor ter

mais conforto em suas compras, pesquisar melhores preços e receber promoções e cupons exclusivos baseados em seus hábitos de compras ou buscas, com a ajuda de Sistemas de Recomendação. Além disso, dar possibilidade ao supermercado de ser capaz de direcionar suas promoções para um público mais específico, assim melhorando a eficiência de suas campanhas.

## REFERENCIAL TEÓRICO

Na revisão de literatura, foram abordados os conceitos que motivaram o desenvolvimento do trabalho, a base teórica de construção de *softwares* e análise de literaturas que referenciam o estudo de algoritmos de recomendação e suas variações.

Atualmente, aplicativos de supermercados precisam se aperfeiçoar para levar a experiência de compra em lojas físicas ao patamar das compras *online*, ou seja, toda personalização, praticidade e agilidade que procuram na palma da mão. Os *Apps* de supermercados físicos apresentam pontos fracos em relação a comunicação e interação com o consumidor, muitos entregam apenas a possibilidade de visualizar os *flyers*/jornais e promoções do dia sem qualquer ligação com o usuário. Em relação a estes pontos, nosso projeto visa gerar melhorias na recomendação de produtos para os usuários, levando a personificação, interação e otimização de compra.

### *Sistema de recomendação*

O sistema de recomendação é responsável por recomendar itens específicos para atender a preferência dos usuários a partir de técnicas computacionais. Para o sistema de recomendação foram levantados pontos-chaves que fazem diferença na aplicabilidade do sistema pretendido. Atualmente estes algoritmos funcionam com duas abordagens de alto nível: *content-based filtering* e *collaborative filtering*. De acordo com Motta et al.(2011), a *content-based filtering* refere-se às informações sobre o conteúdo dos itens, representando usuários e itens como vetores de atributos ou características como descrito por LUNDQUIST(2020). Como exemplo, o usuário A pesquisou filtro de café em um aplicativo de supermercado, e o sistema o recomendará tipos de cafés que podem ser usados com este filtro. Neste contexto, a classificação será uma matriz de item-conteúdo com itens nas linhas e atributos de itens nas colunas, como mencionado por BORODESCU (2021). No modelo de *collaborative filtering*, o sistema utiliza simultaneamente as similaridades das consultas e dos itens para que sejam geradas as recomendações. Baseia-se, basicamente, apenas no comportamento anterior do consumidor, conforme dito por LUNDQUIST (2020). Este modelo analisa padrões de concorrência para determinar a semelhança entre usuários ou itens e tenta inferir as preferências de um outro usuário para os itens não vistos. LUNDQUIST (2020) ainda expõe que a abordagem de *collaborative filtering* é baseada em *Matrix Factorization* (MF). Esta matriz de interação de usuário-item esparsa observada é “fatorada” no produto aproximado de duas matrizes de menor dimensão que contém *embedding* de usuário e item. No entanto, este processo encontra problemas quando precisa lidar com dados de *feedback* implícitos.

### *Coleta e análise dados*

A coleta de dados das preferências do usuário pode ocorrer de formas diversas, e a mais comum em contextos reais é por *feedback* implícito. LOY (2020) explica que os dados de preferência são coletados das interpretações do usuário e agem como um *proxy* para as preferências dele. Por exemplo, os itens em que o usuário clicou, buscou, visualizou ou

comprou, são usados como dados para *feedback* implícito, para que então itens semelhantes sejam sugeridos para este usuário. LUNDQUIST (2020) aponta que neste caso não é observado nenhuma classificação numérica explícita ou resposta positiva ou negativa, como ocorre nos casos de *feedback* explícito, mas apenas a apuração do comportamento bruto do usuário. Para adaptar a abordagem de *matrix factorization* aos dados de *feedback* implícito, HU et al. (2008) introduziu o conceito de classificação binária (o usuário observou o item ou não). Entretanto, apesar deste modelo de *matrix factorization* ter a vantagem de ser simples no âmbito de dados necessários para seu funcionamento, precisando apenas de uma matriz de *feedbacks*, a classificação binária (0 e 1) da preferência do usuário pode gerar erro de previsão de classificação dos interesses reais, e também é incapaz de incorporar recursos da *content-based filtering*, que de certa forma acabam sendo dados perdidos, não aproveitados. Além de não ser um bom método para casos em que necessite lidar com um usuário *cold-start*, SANTANA (2018) explica que são os usuários com pouco ou quase nenhum histórico de consumo ou mesmo aqueles que não realizaram o login, mas que estão navegando na plataforma, que nestas situações gera recomendações aleatórias.

### *Field Aware Factorization Machines (FFM)*

Para aprimorar e melhorar a classificação e minimizar os problemas encontrados no parágrafo anterior, foi utilizado o *field aware factorization machines* (FFM). No entanto, antes de explicar os motivos da sua escolha, é preciso entender um método intermediário entre MF e FFM, o *factorization machines* (FM), que como citado por CHONG (2020) “é formulado como um modelo linear, com interações entre recursos como parâmetros adicionais (recursos).” Em *factorization machines* cada recurso tem apenas um vetor latente para aprender com todos os outros recursos. Vamos exemplificar (1) como se tivéssemos 3 campos (colunas) Gênero, Setor e Item, e, analisando a linha 01 da tabela 1, a interação seria:

**Tabela 1** - Exemplo de tabela de dados com 3 campos

| Campos   | Genero    | Setor           | Item    | Avaliação |
|----------|-----------|-----------------|---------|-----------|
|          | masculino | hortifruti      | maça    | 3         |
|          | feminino  | açougue         | picanha | 5         |
| ↑        | feminino  | adega e bebidas | gim     | 3         |
| recursos | masculino | açougue         | picanha | 4         |
| ↓        | feminino  | adega e bebidas | vinho   | 5         |

Fonte: próprio autor

$$V_{\text{masculino}} * V_{\text{hortifruti}} + V_{\text{masculino}} * V_{\text{maça}} + V_{\text{hortifruti}} * V_{\text{maça}} \quad (1)$$

Assim, o vetor latente de Gênero é usado para aprender o efeito latente de Setor e Categoria. E para o caso de FFM, utilizando o mesmo exemplo acima, seria dividido a representação única em múltiplos vetores latentes, sendo a interação latente da seguinte forma para este caso:

$$V_{\text{masculino, setor}} * V_{\text{hortifruti, gênero}} + V_{\text{masculino, item}} * V_{\text{maça, gênero}} + V_{\text{setor, item}} * V_{\text{maça, setor}} \quad (2)$$

A comparação entre estes dois tipos de matrizes, o FFM aprende múltiplos vetores latentes para cada recurso (2), e com isso conseguimos uma precisão melhor na classificação para a recomendação, além de as características serem agrupadas em “campos”, explica ZHAG et al. (2017).

## METODOLOGIA

O presente trabalho se propõe a gerar conhecimentos e aplicação prática direcionado a solução do problema especificado no mesmo, tendo isto em vista, o trabalho se encaixa na metodologia de pesquisa de natureza aplicada. Os passos seguidos têm como base a metodologia de desenvolvimento de *software* Cascata, que é processo onde é tomado uma série de passos para a construção completa de um *software*, estes passos são conhecidos como: análise de requisitos; projeto; implementação; testes; integração; e manutenção de *software*. Como o projeto se trata de um desenvolvimento experimental, alguns passos não serão seguidos à risca e por tratar de funcionalidades com Algoritmos de Recomendação, é adicionado um passo direcionado ao assunto.

Uma das principais etapas para o desenvolvimento de um *Software* é a análise de requisitos. Para ser definido os requisitos do projeto a ser desenvolvido, foi feito uma análise de um aplicativo já existente, para identificar seus pontos fracos e sugerir melhorias para o que será desenvolvido. Dado a análise, foram desenvolvidos protótipos de tela do aplicativo para tornar a ideia mais palpável.

Com os protótipos feitos, foi possível definir as principais funcionalidades e entidades que seriam utilizadas pelo aplicativo. Com estas informações e o tempo disponível, foi decidido o escopo de desenvolvimento, que seria desenvolver um aplicativo para dispositivos móveis de maneira básica, construção de um *back-end* com as principais funcionalidades e por fim o desenvolvimento de um algoritmo de recomendação.

Baseado no protótipo foi desenvolvido um aplicativo básico, contendo a tela inicial contendo seções para exposição dos produtos escolhidos pelo algoritmo de recomendação, *banners* e listagem de produtos. Também foi programado o *back-end* da aplicação, onde foram criadas as principais operações de gerenciamento das entidades utilizadas no aplicativo móvel.

A última etapa de desenvolvimento, foi desenvolver o algoritmo de recomendação, que consistiu em:

- Separação dos dados
- Transformação dos dados
- Aplicação em um modelo *FFM*
- Treinamento do modelo
- Teste do modelo
- Criação de dados artificiais

## RESULTADOS E DISCUSSÃO

### *Requisitos e Escopo*

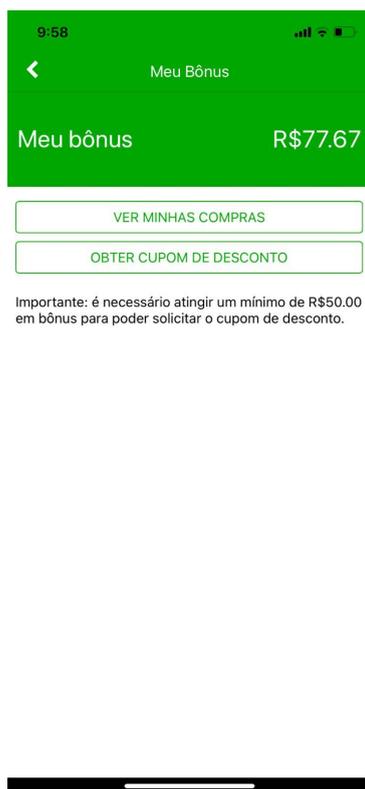
A interação do usuário com o aplicativo é um ponto importantíssimo. Assim como a disposição dos produtos nas gôndolas dos supermercados influenciam na decisão da compra, o *layout* do *App* está diretamente ligado à satisfação do usuário ao navegar, interagir e realizar compras. Abaixo, temos um exemplo de aplicativo com problemas de usabilidade de supermercado (foi retirado o nome da empresa por privacidade) para demonstrar como pequenos detalhes podem gerar perda de interesse do consumidor e conseqüentemente menos venda, mesmo que as ofertas de produtos sejam atrativos.

Figura 1 - Tela inicial



Fonte: Próprio autor

Figura 2 - Menu Bônus



Fonte: Próprio autor

Figura 3 - Tela "minhas compras"



Fonte: Próprio autor

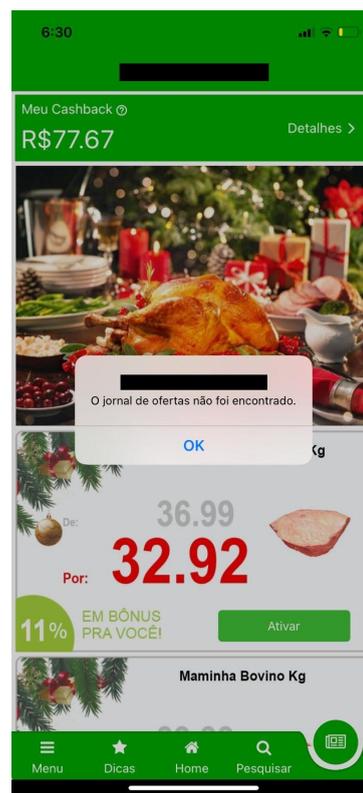
Na tela inicial do App (Figura 1), apesar dos valores do produto e do bônus estarem visíveis, as proporções de imagem e disposição não estão adequados, além de não gerar nenhum interesse de procura para os próximos itens abaixo na rolagem da tela. Já na Figura 2, é possível visualizar a página de "Meu Bônus" em que traz poucas informações de como utilizar ou conseguir mais bônus. Ao ir na opção do botão "ver minhas compras", o usuário consegue verificar as compras realizadas que geraram o valor do bônus disponível.

Figura 4 - Tela “Pesquisar”



Fonte: Próprio autor

Figura 5 - Tela “Jornal”



Fonte: Próprio autor

Na Figura 4, onde haveria a possibilidade de pesquisa de itens, não há nenhum retorno, ou seja, não há um banco de dados dos itens disponíveis, sejam com ou sem retorno de bônus. Outra falha no *App* se encontra na opção em que seria possível visualizar o “jornal” do supermercado, mas que também está indisponível (figura 5). Neste exemplo de aplicativo é possível notar que o supermercado também não apresenta ao usuário nenhuma recomendação ao perfil dele, todo o aplicativo é genérico, sem qualquer personalização.

A proposta criada é um aplicativo com mais interação com o usuário e com maior usabilidade mobile. Para entender melhor os requisitos, foram criados protótipos básicos de tela, utilizando o programa Figma, um editor gráfico utilizado para esse tipo de tarefa. Como pode ser observado nas figuras a seguir, foi criada uma ideia de interface que foca no controle econômico do usuário e nas promoções e cupons disponibilizados para ele.

Figura 6 - Tela inicial



Fonte: Próprio autor

Figura 7 - Tela de Cupons



Fonte: Próprio autor

Figura 8 - Tela de Finanças



Fonte: Próprio autor

Levando em conta os elementos presentes nos protótipos de tela e que o tempo hábil a ser consumido era limitado, foi definido um escopo reduzido de funcionalidades. Como o foco do trabalho é a melhor experiência do usuário baseada nas recomendações, foi definido como prioridade de desenvolvimento: a tela inicial com as recomendações e banners; o desenvolvimento do *back-end* com os recursos básicos de cadastro e manutenção de produtos, promoções, usuários e interações; e, por fim, o desenvolvimento do sistema de recomendação.

### *Desenvolvimento*

Para o desenvolvimento da parte funcional de cadastro e manutenção das informações utilizadas no app para dispositivos móveis, foi criado um módulo web *back-end*, em que foi utilizado Java como linguagem de programação padrão, juntamente com o *framework* de desenvolvimento Spring Boot e o Banco de Dados utilizado foi o MySQL.

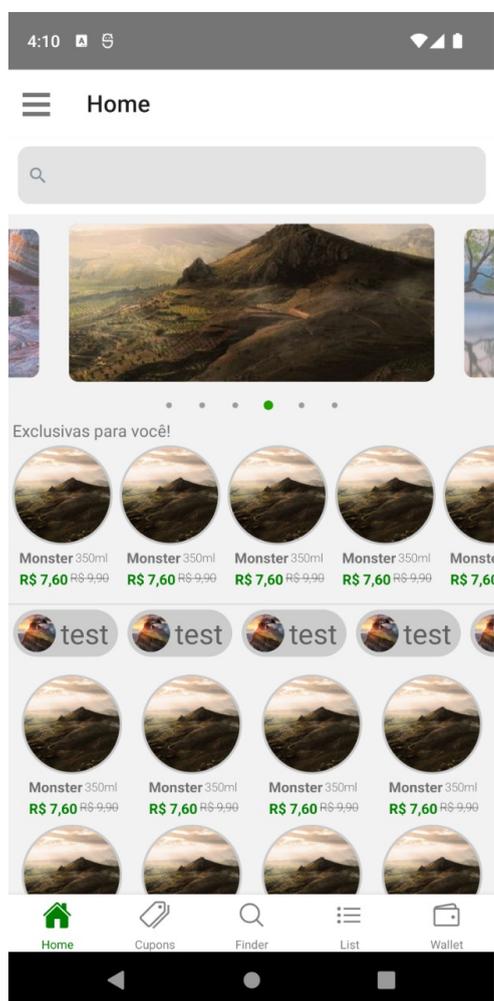
A base de *design* da Arquitetura utilizada foi o *Clean Architecture*, um modelo proposto por Robert C. Martin em seu livro *Clean Architecture: A Craftsman's Guide to Software Structure and Design* (2017), que tem como ideia facilitar o desenvolvimento de códigos, permitir uma melhor manutenção, atualização e menos dependências ZARELLI (2020).

Foram desenvolvidas as operações de criação, resgate, edição e deleção de produtos, usuários, interações dos usuários, promoções e descontos. Foram disponibilizados via *web API*, que é uma interface de programação de aplicações (API), que possibilita comunicação entre o servidor e o cliente.

Para o aplicativo para dispositivos móveis, foi utilizado *TypeScript* como linguagem de programação principal em conjunto com o *framework React Native*, em que, dado o escopo

do projeto, foi desenvolvido o menu de navegação inferior e a tela principal do aplicativo, possibilitando ao usuário visualizar promoções exclusivas, *banners* de campanhas personalizadas e uma visualização básica de itens em geral, conforme a figura a seguir.

**Figura 9 - Tela inicial**



Fonte: Próprio autor

### *Aplicação do Modelo FFM*

Durante o desenvolvimento do *back-end* do aplicativo, contemplamos a entrada de uma série de dados que o usuário poderia nos fornecer, como informações básicas do usuário, suas interações dentro do sistema, incluindo quais produtos e suas categorias. Com base nesses dados, podemos ter uma noção de seu hábito de compra e interesses, assim podendo cruzar informações com outros perfis similares e prever quais produtos ele possui afinidade e que seja mais provável de realizar a compra, desta forma podendo ser recomendado especificamente ao usuário caso haja promoções e cupons compatíveis.

Para o desenvolvimento do sistema de recomendação, devido a facilidade de transformação e tratativa de dados, foi escolhido Python como a linguagem de programação padrão do sistema de recomendações. Com isso, também utilizamos a biblioteca Pandas, para auxiliar na coleta e transformação dos dados. Em conjunto a essas tecnologias, foi utilizado a biblioteca xLearn, que nos permite facilmente gerar modelos de FFM e FM.

Os dados utilizados provêm de 3 tabelas principais, a tabela *user*, *interaction* e a *product\_categories*, que correspondem aos seguintes modelos:

Figura 10 - Modelagem Interação



Fonte: Próprio autor

Para os dados serem utilizados pelo sistema *FFM*, foram convertidos para apenas uma visualização, em que conseguimos agrupar os dados, que servirão como ancoragem para o modelo preditivo funcionar e relacioná-los. Como um usuário pode ter mais de uma interação e mais de um tipo por item, essas interações foram contadas e acumuladas em uma coluna que foi chamada de *affinity*, onde a contagem depende do peso de cada tipo de interação, por ser considerado que algumas ações são mais importantes que interações mais simples e que necessitam de menos interesse no item. Para interações de clique e procura, foi considerado peso 1, e para adição em lista de compras ou lista de favoritos o peso 2. Isso quer dizer que quanto maior o número de afinidade, mais interessado o usuário está com o item, aumentando suas chances de ser recomendado.

Outra transformação necessária realizada foi agregar as interações de compras em uma coluna separada, onde se houver a compra do item, essa coluna é verdadeira, assim podendo relacionar o número de afinidade com a possibilidade de compra e definir a compra como coluna alvo para a predição a ser feita. Para realizarmos a transformação dos dados, foi criado uma *query* SQL em um método de resgate de dados do *framework* Pandas, como pode ser conferido na Figura a seguir:

Figura 11 - Busca e transformação dos dados

```
rawData = pd.read_sql('''SELECT i.user_id,
    u.gender,
    i.product_id,
    p.categories as category,
    COUNT(CASE
        WHEN i.interaction = 'CLICK' || 'FIND' THEN 1
        WHEN i.interaction = 'LIST_ADD' || 'FAVORITE_ADD' THEN 2
        ELSE 0 END) as affinity,
    MAX(i.buy) as buy
#     IF(i.interaction = 'buy', 1, 0) AS buy
FROM (SELECT it.user_id, it.product_id, it.interaction, IF(it.interaction = 'buy', 1, 0) AS buy
    FROM interactiondto it) i, user u, product_categories p where u.id=i.user_id and p.product_id = i.product_id
GROUP BY user_id, product_id, p.categories''', con=db_connection)
return rawData
```

Fonte: Próprio autor

Para usar o campo *affinity* na modelagem, com a ajuda do *framework* Pandas, foi necessário classificar os intervalos numéricos e classificá-los como categorias, e então convertê-los para o formato libffm. Onde a escala de afinidade passou a ser separada em 7 categorias como visto na Figura 12.

**Figura 12** - Categorização de Afinidade

```
affinities = [0, 4, 8, 16, 32, 64, np.inf]
names = ['<4', '4-8', '8-16', '16-32', '32-64', '64+']

rawData['affinity_range'] = pd.cut(rawData['affinity'], affinities, labels=names)
```

Fonte: Próprio autor

Para que os dados sejam utilizados no aprendizado da FFM, é preciso convertê-los para o formato libffm CHOUDHARY(2018). Os dados devem ter uma configuração como a Figura 13.

**Figura 13** - Formato dos dados

```
<label> <field1>:<feature1>:<value1> <field2>:<feature2>:<value2> .....
```

Fonte: Factorization Machines | Applications On Huge Dataset, 2018

Para conversão dos dados para libffm, foram utilizados um método de utilidade encontrado em Factorization Machines | Applications On Huge Dataset, CHOUDHARY (2021).

Por fim, é criada uma instância de um modelo FFM e é configurado uma coleção de dados de treinamento, definido os parâmetros de treinamento, definindo onde o modelo será salvo e ativando o *cross-validation*. Assim, é possível inserir a uma coleção de dados e calcular uma predição baseada no modelo calculado e que será salva em um arquivo local, como pode ser observado na Figura 14.

**Figura 13** - Formato dos dados

```
ffm_model = xl.create_ffm()
ffm_model.setTrain("train_ffm.txt")
param = {'task': 'binary',
         'lr': 0.2,
         'lambda': 0.002,
         'metric': 'acc'}

ffm_model.fit(param, './model.out')
ffm_model.cv(param)

ffm_model.setTest(
    "test_ffm.txt")
ffm_model.setSigmoid()
ffm_model.predict("./model.out", "./output.txt")
```

Fonte: Próprio autor

Como não se teve acesso a dados reais, foram criados dados artificiais para verificar como é o funcionamento do modelo e seu desempenho. Para isso, foi desenvolvido um algoritmo que implantou produtos e usuários aleatórios no banco de dados utilizado. Para que os dados de interação não fiquem aleatórios demais e não haja padrões a serem encontrados na predição, para criar dados de interação foram usadas as seguintes condições: 100 perfis com 200 usuários cada, que consomem pelo menos uma vez (1 a 10) 70% dos produtos de uma faixa de produtos aleatórios de 100 itens e interage pelo menos uma vez (1 a 5) com 30% desses itens. Dessa maneira, se identificado produtos parecidos com o perfil gerado, é possível validar o funcionamento da predição.

Com os dados artificiais gerados e utilizado o algoritmo desenvolvido, foi possível alcançar uma acurácia média de 86%, o que é considerado bem preciso.

## CONCLUSÕES

Neste trabalho abordamos a importância de um sistema de recomendação nos aplicativos para supermercados físicos para personalizá-lo de acordo com cada usuário, principalmente no momento atual em que estamos passando devido às restrições causadas pelo COVID-19. O usuário ter acesso a um aplicativo pensado para ele traz maior conforto e segurança no momento da compra, além de levar vantagens para ambos os lados, usuário e supermercadista. Junto a isso, foi feito um desenvolvimento inicial de um sistema que busca suprir as necessidades do usuário na questão da comodidade ao frequentar supermercados.

Os 3 módulos principais que compõem o sistema total proposto, o *back-end*, aplicativo e modelo de recomendação não foram integrados, ou seja, por mais que o modelo de recomendação tenha alcançado um nível satisfatório de precisão com dados artificiais, ainda não é possível validar a precisão e funcionamento das recomendações em um cenário real. Ainda não é possível interagir com o aplicativo, gerar dados válidos e utilizar o algoritmo de recomendação desenvolvido. Dado as circunstâncias, não é possível validar a melhoria na experiência do usuário ao utilizar um aplicativo durante o consumo em um supermercado.

O algoritmo preditivo utilizado, a primeiro momento, se mostra satisfatório no resultado, mas o ideal seria sua validação em um cenário real e em comparativo com outros tipos de algoritmos de recomendação, para que seja possível avaliar sua real precisão e utilidade.

Os próximos passos a serem tomados devem ser direcionados para a integração de todos esses módulos e completa utilização dos recursos desenvolvidos. Caberia também uma pesquisa de campo, para identificar os pontos importantes e as dificuldades que o público identifica ao utilizar os aplicativos de supermercados atuais, aumentando assim a precisão dos recursos desenvolvidos e se aproximando de uma experiência satisfatória.

## REFERÊNCIAS BIBLIOGRÁFICAS

APPSFLYER. **The state of App Marketing in Latin America** - 2020 Edition. AppsFlyer, c2021. Disponível em: <<https://www.appsflyer.com/resources/state-of-app-marketing-latam/>>. Acesso em: 05 ago. 2021.

BORODESCU, Ciprian. **The anatomy of high-performance recommender systems** – Part 1. Algolia, 02 abr. 2021. Disponível em

<<https://www.algolia.com/blog/ai/the-anatomy-of-high-performance-recommender-systems-part-1/>>. Acesso em: 05 ago. 2021.

CHOUDHARY, Ankit. **Introductory Guide – Factorization Machines & their application on huge datasets (with codes in Python)**. Analytics Vidhya, 07 jan. 2018. Disponível em: <<https://www.analyticsvidhya.com/blog/2018/01/factorization-machines/>>. Acesso em: 1 dez. 2021.

CHONG, David. **An Intuitive Explanation of Field Aware Factorization Machines**. Towards Data Science, 02 maio 2020. Disponível em: <<https://towardsdatascience.com/an-intuitive-explanation-of-field-aware-factorization-machines-a8fee92ce29f>>. Acesso em: 25 nov. 2021.

**Clean Architecture: A Craftsman’s Guide to Software Structure and Design (Robert C. Martin Series) (English Edition) eBook : C., Martin Robert: Amazon.com.br: Livros, Amazon.com.br, disponível em: <[https://www.amazon.com.br/dp/B075LRM681/ref=dp-kindle-redirect?\\_encoding=UTF8&btkr=1](https://www.amazon.com.br/dp/B075LRM681/ref=dp-kindle-redirect?_encoding=UTF8&btkr=1)>. Acesso em: 01 dez. 2021.**

HU, Yifan, KOREN, Yehuda e VOLINSKY, Chris. **Collaborative Filtering for Implicit Feedback Datasets**. Eighth IEEE International Conference on Data Mining, 2008. Disponível em: <<https://ieeexplore.ieee.org/document/4781121>>. Acesso em: 15 nov. 2021.

**Integre seus canais de vendas a partir do conceito de omnichannel**. SEBRAE, 10 nov. 2017. Disponível em: <<https://www.sebrae.com.br/sites/PortalSebrae/artigos/integre-seus-canais-de-vendas-a-partir-do-conceito-de-omni-channel,87426f65a8f3a410VgnVCM2000003c74010aRCRD#:~:text=87426f65a8f3a410VgnVCM2000003c74010aRCRD-,O%20que%20%C3%A9%20omnichannel%3Flojas%20f%C3%ADsicas%2C%20virtuais%20e%20compradores>>. Acesso em: 18 nov. 2021.

LOY, James. **Deep Learning based Recommender Systems**. Towards Data Science, 18 out. 2020. Disponível em: <<https://towardsdatascience.com/deep-learning-based-recommender-systems-3d120201db7e>>. Acesso em: 15 set. 2021.

LUNDQUIST, Eric. **Factorization Machines for Item Recommendation with Implicit Feedback Data**. Towards Data Science, 28 jun. 2020. Disponível em: <<https://towardsdatascience.com/factorization-machines-for-item-recommendation-with-implicit-feedback-data-5655a7c749db>>. Acesso em: 20 nov. 2021.

MU, Ruihui, ZENG, Xiaoqin, LIXIN, Han. **A Survey of Recommender Systems Based on Deep Learning**, nov. 2018. Disponível em: <[https://www.researchgate.net/publication/328843931\\_A\\_Survey\\_of\\_Recommender\\_Systems\\_Based\\_on\\_Deep\\_Learning](https://www.researchgate.net/publication/328843931_A_Survey_of_Recommender_Systems_Based_on_Deep_Learning)>. Acesso em 05 ago. 2021.

**O seu supermercado está preparado para a pós pandemia?** Vabene, 22 abr. 2021. Disponível em

<<https://vabene.com.br/o-seu-supermercado-esta-preparado-para-a-pos-pandemia/>>. Acesso em: 18 nov. 2021.

SANTANA, Marlesson. **Deep Learning para Sistemas de Recomendação (Parte 1) — Introdução.** Medium, 06 dez. 2018. Disponível em: <<https://medium.com/data-hackers/deep-learning-para-sistemas-de-recomenda%C3%A7%C3%A3o-parte-1-introdu%C3%A7%C3%A3o-b19a896c471e>>. Acesso em: 18 nov. 2021.

ZARELLI, Guilherme. **Descomplicando a Clean Architecture - luizalabs - Medium.** Medium, 29 jun. 2020. Disponível em: <<https://medium.com/luizalabs/descomplicando-a-clean-architecture-cf4dfc4a1ac6>>. Acesso em: 01 dez. 2021.

ZHANG, Zhiyuan, LIU, Yun e ZHANG Zhenjiang. **Field-Aware Matrix Factorization for Recommender Systems**, 28 dez. 2017. Disponível em: <<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8241366>>. Acesso em: 25 nov. 2021.