RAISSA POLYANNA PAPINI DE MELO SOUZA

RECOMENDAÇÃO DE APLICATIVOS MÓVEIS COM BASE EM INFORMAÇÕES DEMOGRÁFICAS E DE DISPOSITIVOS

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

Orientador: Fabrício Aguiar Silva

VIÇOSA - MINAS GERAIS
2021

Ficha catalográfica elaborada pela Biblioteca Central da Universidade Federal de Viçosa - Campus Viçosa

T
S729r
2021

Souza, Raissa Polyanna Papini de Melo, 1997-
Recomendação de aplicativos móveis com base em informações demográficas e de dispositivos / Raissa Polyanna Papini de Melo Souza. – Viçosa, MG, 2021.
84 f. : il. (algumas color.) ; 29 cm.

Inclui apêndice.
Orientador: Fabrício Aguilar Silva.
Dissertação (mestrado) - Universidade Federal de Viçosa.
Referências bibliográficas: f. 79-81.

1. Aplicativos móveis. 2. Demografia - Processamento de dados. I. Universidade Federal de Viçosa. Departamento de Informática. Programa de Pós-Graduação em Ciência da Computação. II. Título.

CDD 22. ed. 005.26

Bibliotecário(a) responsável: Alice Regina Pinto Pires CRB6 2523

RAISSA POLYANNA PAPINI DE MELO SOUZA

RECOMENDAÇÃO DE APLICATIVOS MÓVEIS COM BASE EM INFORMAÇÕES DEMOGRÁFICAS E DE DISPOSITIVOS

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 05 de maio de 2021.

Assentimento:

Raissa Polyanna Papini de M. Souza
Raissa Polyanna Papini de Melo Souza
Autora

Fabrício A. Silva
Fabrício Aguiar Silva
Orientador

Aos meus pais, orientador e meu noivo Leonardo.

AGRADECIMENTOS

Ao meu noivo Leonardo por ter me apoiado e incentivado nos momentos mais difíceis. Aos meus pais que sempre acreditaram em mim, sem eles não teria alcançado tudo que consegui. Ao meu orientador Fabrício pelo incentivo, paciência e por ter feito todo o possível para me direcionar, muito obrigada! Ao aluno Gabriel, que colaborou com tantas coisas nesse projeto, e à CAPES pelo financiamento. Por fim, à todas as outras pessoas que fizeram parte dessa caminhada.

RESUMO

SOUZA, Raissa Polyanna Papini de Melo, M.Sc., Universidade Federal de Viçosa, maio de 2021. **Recomendação de aplicativos móveis com base em informações demográficas e de dispositivos**. Orientador: Fabrício Aguiar Silva.

Nos últimos anos tem-se percebido um grande aumento no número de pessoas com acesso a dispositivo móveis. Com isso, o número de aplicativos para esses dispositivos tem crescido de tal forma que usuários precisam escolher entre aqueles que melhor os atendem. Porém, essa escolha não é uma tarefa trivial, visto o número cada vez maior de aplicativos se propondo a realizar a mesma função. Da mesma forma, as empresas por trás de tais aplicativos encontram dificuldades em atrair usuários através de campanhas comuns de *marketing*. Uma possível solução para este problema é a utilização de sistemas de recomendação, onde é possível avaliar a similaridade entre perfis de usuários. Entretanto, muitas vezes tais sistemas levam em consideração perfis de usuários que são construídos apenas com seus interesses, ou necessitam da utilização de dados sensíveis (e.g., *logs* de chamadas e de mensagens de texto). Apesar disso, a instalação de um aplicativo pode envolver outros fatores além do interesse intrínseco de cada usuário, como por exemplo a capacidade técnica do dispositivo móvel utilizado (e.g., memória, processamento), e as informações demográficas de sua área de residência. Assim, o trabalho desenvolvido nesta pesquisa investigou a motivação para instalação, e o impacto do uso de informações demográficas e de dispositivos na recomendação de aplicativos móveis. Para isso, foi criado um perfil de usuário que utiliza somente dados facilmente obtidos (i.e., aplicativos instalados, localização aproximada e o modelo do dispositivo móvel) para ser enriquecido com outras informações de contexto do usuário. Além disso, o uso de tal perfil de usuário foi avaliado com base em três abordagens de recomendação: *Latent Dirichlet Allocation* (LDA), Cadeias de *Markov* (MTM) e Filtro Colaborativo. Os resultados gerais mostraram uma maior eficácia com o uso da abordagem LDA, utilizando informação da renda média da região do usuário, atingindo aproximadamente 64% de melhora em precisão e 27.91% em revocação.

Palavras-chave: Recomendação de Aplicativos. Dados Demográficos. Enriquecimento de Dados.

ABSTRACT

SOUZA, Raissa Polyanna Papini de Melo, M.Sc., Universidade Federal de Viçosa, May, 2021. **Mobile app recommendation based on demographic and handset information**. Advisor: Fabrício Aguiar Silva.

Nowadays the number of people with access to mobile devices has been increasing significantly. Thus, users have to choose among a high number of apps, those that better serve them. However, this is not a trivial task, as we have seen an increasing number of apps proposing to do the same functions. In the same way, companies are facing difficulties to attract users through a usual marketing campaign. A possible solution for this problem is the use of recommendation systems, where it is possible to compare the similarities of user profiles. Meanwhile, these systems often consider only users' preferences to create a profile, or request sensitive data (e.g., call and message logs). However, to install an app may involve other factors like the capacity of the mobile device (e.g., memory and processing power) and the demographic information of the user's living area. So, this research investigated the users' motivation of installment and the impact of using demographic and handset information on app recommendation. To do that, we used this information to enrich the user profile that uses only easy-to-obtain data (i.e., installed apps, approximate location, and handset model). Besides, this profile was evaluated based on three different recommending approaches: Latent Dirichlet Allocation (LDA), Markov Chain (MTM), and Collaborative Filtering. The general results reveal that the LDA approach achieved the highest efficacy when added information about the user's region mean wage, in terms of precision (approximately 64%) and recall (approximately 27.91%).

Keywords: App Recommendation. Demographic Information. Data Enrichment.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

Capítulo 1

Introdução

Atualmente, tem-se percebido um grande aumento no número de pessoas com acesso a dispositivos móveis (GSMA, 2020). De fato, o uso destes já não se limita mais aos recursos básicos, como envio e recebimentos de chamadas e mensagens SMS. Com o advento da computação ubíqua e pervasiva, dispositivos móveis têm auxiliado pessoas desde tarefas complexas, como movimentações bancárias, até atividades rotineiras, como ajuste de alarmes. Com isso, o número de aplicativos (*apps*) para dispositivos móveis tem crescido de tal forma que usuários devem escolher entre aqueles que melhor os atendem. Porém, esta escolha não é tão simples, visto que o número de aplicativos disponíveis para *download* tem crescido significativamente, com a *Google Play Store* chegando a mais de 3.4 milhões de *apps* disponíveis (Matters, 2021).

Existem várias formas de se recomendar aplicativos a usuários. Uma recomendação mais simples pode utilizar somente dados relativos aos próprios aplicativos, recomendando para o usuário aqueles que possuem maior taxa de aceitação em termos de popularidade. Já outros sistemas de recomendação levam em consideração dados referentes a cada usuário, agregando tanto informações fáceis de se obter (e.g., localização aproximada do usuário), como informações mais sensíveis (e.g., *logs* de chamadas). Estas informações podem ser obtidas através de permissões concedidas por cada usuário, e podem abranger diversos sensores embutidos nos dispositivos, como GPS, acelerômetro, *WiFi*, entre outros.

Através da coleta de tais informações, é possível construir um perfil do usuário. Dessa forma, são levantadas e analisadas características e gostos do usuário, que poderão contribuir para o entendimento de quais aplicativos terão chance de serem escolhidos. Além disso, a utilização do perfil do usuário possibilita o entendimento de suas relações sociais, além de preferências pessoais e de mobilidade (Goel and Kumar, 2018).

Entretanto, a obtenção de tais dados não é uma tarefa trivial. Isso porque muitas dessas informações são sensíveis e, portanto, mais difíceis de serem obtidas. Dados sensíveis são aqueles que revelam informações pessoais de um indivíduo como convicções religiosas, filiações, informações médicas, entre outras. Além disso, muitos usuários não concordam com a utilização de seus dados pessoais, fazendo com que muitas abordagens não possam ser utilizadas em larga escala.

1.1 O problema e sua importância

Hoje em dia, existe um grande número de aplicativos móveis e, muitos destes, desenvolvidos para o mesmo propósito. Então, para fazer uma instalação, usuários têm a difícil missão de pesquisar entre os aplicativos para encontrar aquele que melhor se encaixa nas suas necessidades e preferências. Além disso, aplicativos não populares são raramente recomendados, já que não possuem boa visibilidade em termos de downloads ou avaliações recebidas. Entretanto, tais *apps* podem ser extremamente relevantes para um determinado grupo de nicho, que não recebe tal recomendação por não possuir um perfil como o da maioria dos usuários.

Por outro lado, as empresas por trás do desenvolvimento de aplicativos têm enfrentado uma concorrência cada vez maior. Com isso, muitos recursos são gastos com campanhas tradicionais de *marketing*, muitas vezes empregadas a usuários com pouca ou nenhuma chance de aderir ao produto. Da mesma forma, pequenas empresas têm dificuldade de atingir o público-alvo com potencial interesse, como aplicativos de transporte específicos para uma cidade.

Para mitigar esses problemas, o uso de um sistema de recomendação de aplicativos móveis pode trazer facilidade aos usuários, que podem não precisar mais fazer uma pesquisa profunda para encontrar as aplicações desejadas. Do mesmo modo, empresas responsáveis pelo desenvolvimento de aplicativos podem poupar recursos ao direcionar campanhas de *marketing* aos usuários corretos.

Para que esta tarefa seja possível, alguns trabalhos na literatura utilizam modelos baseados na estratégia de filtro colaborativo para recomendar aplicativos, normalmente adicionando a um perfil de usuário, informações de seus interesses. Já outras abordagens resolvem este problema através de técnicas de recuperação de informação ou de processamento de linguagem natural (PLN). Esses trabalhos utilizam de algoritmos como *Bag of Words*, *Word2vec*, entre outros, para identificar padrões de coocorrência entre os aplicativos analisados.

Entretanto, um problema comum a todas essas abordagens diz respeito ao fato de que os dados adicionados podem conter informações como preferências de privacidade, histórico de uso, *logs* de chamadas, entre outras que não são fáceis de se obter. Isto pode levar tais abordagens à falta de conhecimento sobre os usuários, causando uma impossibilidade de uso em larga escala. Além disso, a escolha de um aplicativo também pode ter relação com fatores demográficos e de dispositivos, já que nem todos os aplicativos estarão disponíveis para qualquer região ou dispositivo. Apesar disso, não foram encontradas soluções que levassem em consideração informações contextuais demográficas de usuários, ou de seus dispositivos.

Assim, diversos meios têm usado sistemas de recomendação de aplicativos móveis, mas para que tal recomendação seja mais precisa, é necessário construir um perfil de

usuário com informações relevantes. Por outro lado, alguns usuários não concordam com o uso de seus dados sensíveis, tornando impraticável o uso do sistema em larga escala. Por isso, esta pesquisa propõe resolver o problema de recomendação de aplicativos para usuários, melhorando os resultados alcançados por outras abordagens, porém usando somente dados facilmente obtidos.

1.2 Objetivo Geral

Nesse contexto, a hipótese deste trabalho é que o uso de informações demográficas e de dispositivos favorecem a recomendação de aplicativos móveis. Com base nisso, o objetivo desta pesquisa é validar essa hipótese, através da criação de um perfil de usuário que utilize apenas dados simples de se obter, que englobam informações demográficas da região de residência do usuário e o seu dispositivo móvel.

Os objetivos específicos são:

- Investigar se o uso de informações do tipo de dispositivo do usuário, além de dados demográficos sobre a renda e a população de sua cidade, favorece a recomendação de aplicativos móveis por meio de uma solução baseada em tópicos (i.e., *Latent Dirichlet Allocation* (LDA));
- Investigar se o uso de informações do tipo de dispositivo do usuário, além de dados demográficos sobre a renda e a população de sua cidade, favorece a recomendação de aplicativos móveis por meio de uma solução baseada em filtro colaborativo;
- Investigar se o uso de informações do tipo de dispositivo do usuário, além de dados demográficos sobre a renda e a população de sua cidade, favorece a recomendação de aplicativos móveis por meio de uma solução baseada em cadeias de *Markov*;

1.3 Contribuições

As principais contribuições desta pesquisa são:

- A avaliação da utilização de dados demográficos, através da aplicação destes em três tipos de abordagens de recomendação e em dados reais de milhares de usuários, tendo alcançado uma melhora de até 210% para revocação.
- A criação e a utilização de um perfil de usuário com o uso de dados demográficos e dispositivos, utilizando somente dados fáceis de se obter acerca dos usuários.

- A análise de um questionário respondido por 270 pessoas, buscando entender as motivações de instalação de aplicativos.

1.4 Estrutura do Trabalho

Esta dissertação está estruturada na forma de capítulos e, a partir deste ponto, a organização do texto se dará da seguinte forma: No Capítulo 2 é apresentado o referencial teórico utilizado para desenvolvimento desta pesquisa, assim como a análise das informações obtidas através do questionário de motivação de instalação, e por fim os trabalhos relacionados com esta pesquisa. Já o Capítulo 3, mostra a metodologia e resultados obtidos com a realização de uma análise preliminar, acerca do impacto do uso de informações demográficas. A metodologia utilizada para uma análise mais elaborada, assim como os resultados obtidos através desta, são mostrados no Capítulo 4. Já no Capítulo 5 são discutidas as conclusões e possíveis trabalhos futuros. Por fim, no Apêndice A está contido o questionário elaborado.

Capítulo 2

Referencial Teórico

A identificação dos aplicativos que serão recomendados para um usuário requer que sejam analisadas informações referentes aos próprios aplicativos e/ou ao usuário a quem serão recomendados. Para que tal recomendação seja precisa, deve-se analisar a fundo não só as características inerentes a cada aplicativo, como também os interesses dos usuários que os consomem, ou não. Do mesmo modo, é importante ter conhecimento acerca das características de sistemas de recomendação em geral, para que seja possível a escolha do método mais relevante com base nos dados coletados. Tais dados, também devem ser selecionados de maneira a se adequarem ao método escolhido, possibilitando melhor aproveitamento de ambos.

Sendo assim, neste capítulo são abordados temas relevantes para o desenvolvimento e análise de um sistema de recomendação de aplicativos móveis.

2.1 Sistemas de Recomendação

Ser capaz de prever atos de adesão a produtos e serviços tem se tornado um passo importante para diversas empresas. Com a facilidade de acesso cada vez maior da tecnologia, pessoas são bombardeadas todos os dias por diversas notícias, além de ofertas de produtos e serviços. Com isso, só em 2019, foram movimentados cerca de US$3,53 trilhões derivados do comércio eletrônico (*e-commerce*) (Sabanoglu, 2021). Assim, para que tais pessoas sejam capazes de encontrar mais facilmente produtos de seu interesse, foi-se criado o conceito de Sistemas de Recomendação.

Sistemas de Recomendação são serviços desenvolvidos para se recomendar itens de um conjunto a um determinado usuário, baseando-se nos itens que tal usuário geralmente utiliza (Jacobi et al., 2006). A adoção de sistemas de recomendação pode abranger diversos meios. Aqueles implantados em *sites* de comércio *online*, indica produtos baseando-se em históricos de consumo e pesquisas realizadas por um determinado usuário. Esta abordagem pode ser utilizada para se recomendar tanto produtos específicos, como categorias de produtos, ou itens similares àqueles anteriormente relacionados com o usuário. Outro meio que utiliza este tipo de sistema são os catálogos de *streaming*. Estes meios, por outro lado, podem se basear em avaliações realizadas por outros usuários para se recomendar um item, já que várias pessoas

se interessam pelos mesmos estilos musicais, por exemplo. Assim, músicas, livros, filmes e séries, entre outros itens, podem ser apresentados ao usuário baseando-se em títulos avaliados positivamente por outros usuários com perfil semelhante ao de quem se deseja recomendar.

Assim como as aplicações apresentadas anteriormente, a recomendação de aplicativos pode não depender unicamente dos *apps* instalados anteriormente pelo usuário. Sistemas de recomendação encontrados em lojas de aplicativos, como *Google Play Store* e *Apple App Store*, sugerem também ao usuário a instalação de aplicativos populares.

2.2 Dificuldades Inerentes ao Problema

Apesar de sistemas de recomendação muitas vezes obterem bons resultados, a recomendação de aplicativos apresenta ainda algumas outras dificuldades. Uma delas é caracterizada pela grande quantidade de itens passíveis de serem apresentados ao usuário. Isso acontece porque o número de aplicativos com o mesmo propósito é grande e vem crescendo cada vez mais. Além disso, ao contrário da recomendação de filmes, onde é interessante recomendar títulos semelhantes a outro que obteve boa avaliação por parte do usuário, não é provável que um indivíduo irá aderir a um aplicativo que atende ao mesmo propósito que outro anteriormente instalado. Esta característica faz com que bons sistemas de recomendação de aplicativos tenham de recorrer a outras características dos *apps*, para que seja possível estabelecer uma maior afinidade entre o usuário e um determinado aplicativo dentre todos os pré-selecionados. Além disso, com o número de aplicativos crescendo cada vez mais, é de suma importância o desenvolvimento de sistemas otimizados e com boa escalabilidade.

O número de aplicativos disponíveis não dificulta somente o desempenho computacional dos sistemas. Uma instalação de aplicativos nem sempre segue um padrão, já que é possível que ocorra a instalação baseando-se em motivações abstratas. Por este motivo, algoritmos de Aprendizado de Máquina têm dificuldade em captar quando tais eventos ocorrerão, se atendo à recomendação de aplicativos relacionados ao perfil do usuário e causando uma baixa taxa de acerto.

Outra dificuldade inerente ao problema diz respeito à quantidade de dados necessária para se construir um perfil de usuário. Como muitas abordagens utilizam de dados históricos para se construir tal perfil, o sistema necessita de certo tempo para que possa compreender as características de um novo indivíduo, o chamado *cold start* (Schafer et al., 2007). Apesar de existirem abordagens que não sofrem com este problema, estas podem não alcançar resultados satisfatórios, fazendo com que seja necessária uma análise para se decidir qual abordagem utilizar.

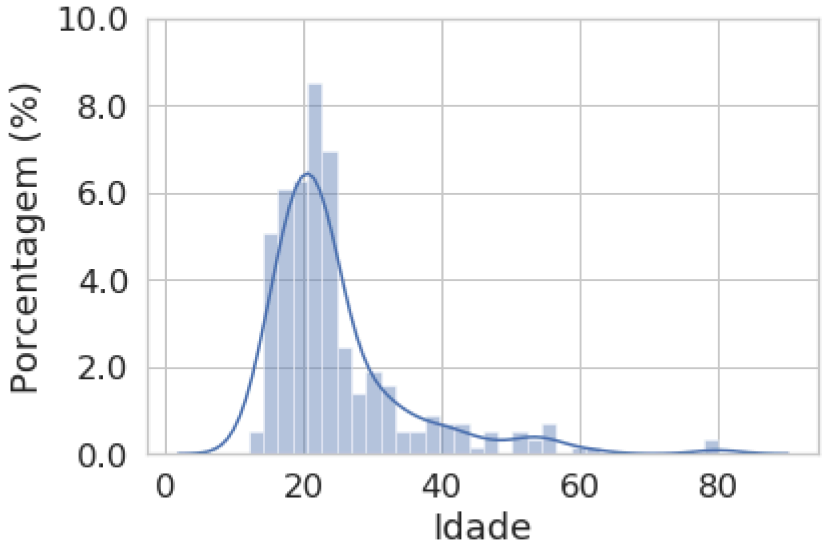


Figura 2.1: Distribuição de idade dos participantes.

2.3 Motivos de Instalação

A fim de melhor entender as causas que levam um usuário a instalar determinado aplicativo, foi elaborado e aplicado um questionário (Apêndice A), que coletou respostas durante o período de um mês, e contou com 270 participantes. Vale ressaltar que não há como estabelecer relação entre as pessoas que responderam o questionário, e os usuários utilizados nos capítulos seguintes desta pesquisa.

Dentre os participantes, 155 se identificavam com o gênero feminino e 115 com o gênero masculino. Além disso, 229 pessoas afirmaram utilizar sistema operacional *Android*, enquanto outras 41 delas utilizam sistema *iOS*. Quanto à idade, apenas 25% dos indivíduos possuem, pelo menos, 26 anos. Além disso, o participante mais novo afirmou ter 12 anos de idade, enquanto o mais velho 80 anos. O histograma da idade pode ser visto na Figura 2.1. Para que fosse possível entender padrões por trás da idade, os participantes foram categorizados como Adolescente (até 19 anos), Jovem (de 20 a 29 anos de idade), Adulto (de 30 a 59 anos) ou Idoso (pelo menos 60 anos).

Além disso, os participantes foram perguntados sobre a sua satisfação com recomendações de aplicativos provenientes de três fontes: redes sociais, lojas de aplicativos e pessoas próximas. Em uma escala de 1 (pouco satisfeito) a 5 (muito satisfeito), redes sociais foram avaliadas com uma satisfação de, em média, 2,95; recomendações de lojas de aplicativos obtiveram 3,12; e recomendações de pessoas próximas foram avaliadas com pontuação de, em média, 4.

Analisando as respostas acerca de redes sociais, vemos na Figura 2.2, que participantes do gênero feminino são mais propensos a instalar *apps* recomendados por redes sociais. Além disso, este canal de recomendação é mais adotado por adoles-

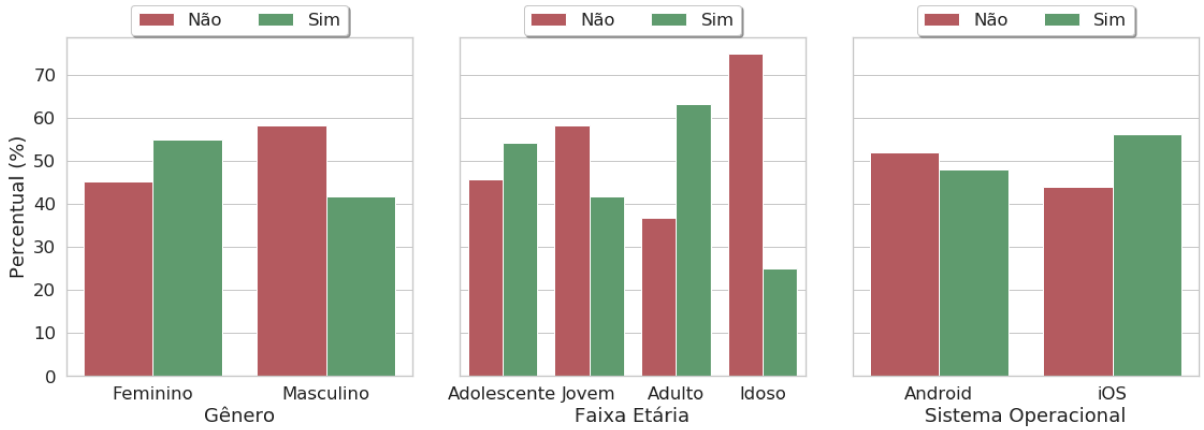


Figura 2.2: Padrão de instalação de aplicativos provenientes de redes sociais.

centes, adultos e pessoas que utilizam *iOS*. Vale ressaltar que, segundo as respostas obtidas, pessoas idosas têm grande chance de não instalar aplicativos recomendados por esta fonte.

Por outro lado, segundo as respostas do questionário, pessoas do gênero masculino têm mais chance de aderirem a *apps* recomendados por lojas de aplicativos, do que pessoas do gênero feminino (Figura 2.3). Além disso, assim como em redes sociais, adultos e adolescentes costumam instalar aplicativos recomendados por lojas, enquanto idosos parecem nunca instalá-los. Quanto ao sistema operacional, parece não haver tendência de instalação.

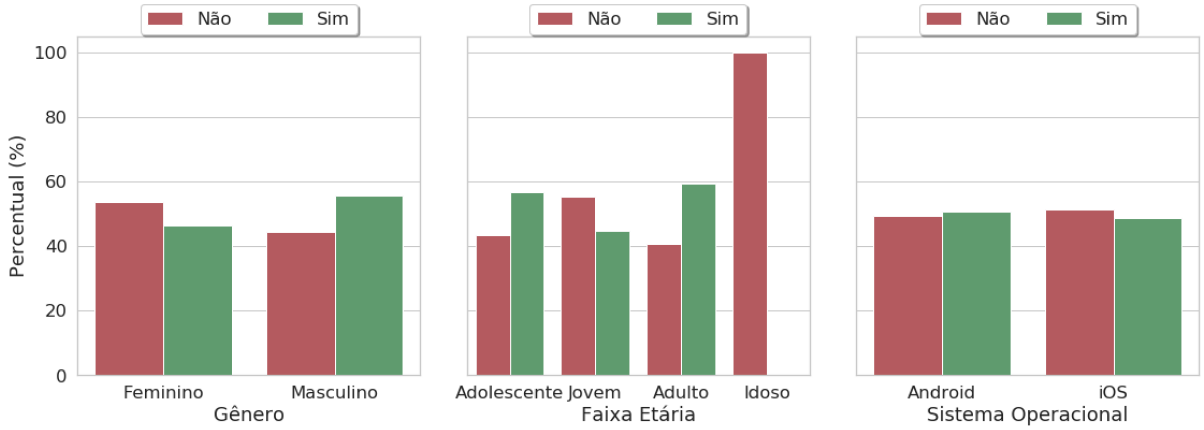


Figura 2.3: Instalação de aplicativos recomendados por lojas de aplicativos.

Entretanto, parece haver consenso quanto à aderência de aplicativos recomendados por pessoas próximas (Figura 2.4). Isso porque a grande maioria dos participantes declarou instalar os *apps* recomendados.

Já no que consta à satisfação com as recomendações, vemos que a maioria dos participantes considera que recomendações de redes sociais são razoáveis (3) ou boas (4) (Figura 2.5). Entretanto, todos os idosos que responderam o questionário, e instalam *apps* recomendados por essa fonte, disseram se sentir satisfeitos com tal recomenda-

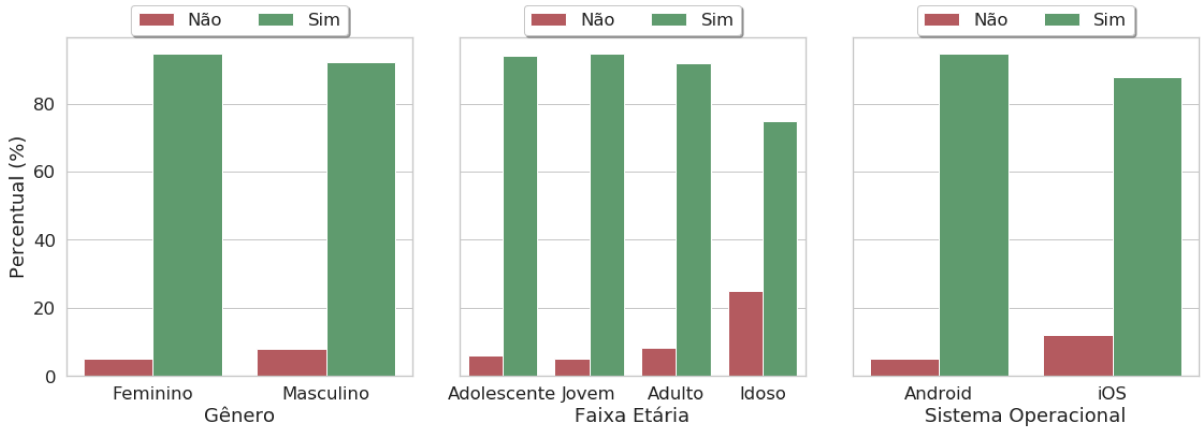


Figura 2.4: Análise de instalação de *apps* recomendados por pessoas próximas.

ção. Por outro lado, o grupo que se sentiu mais satisfeito com tais recomendações foi o de adultos, com quase 20%.

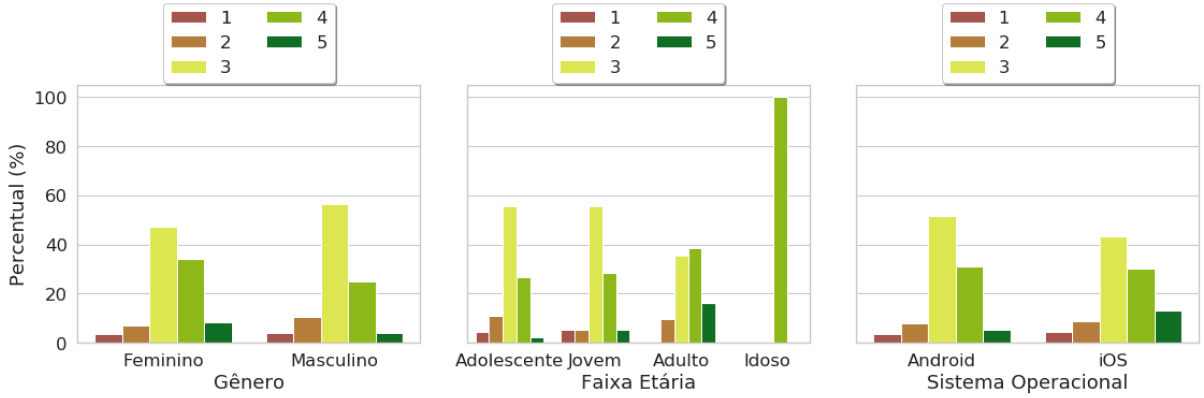


Figura 2.5: Satisfação quanto à recomendação de aplicativos recomendados em redes sociais.

A satisfação com recomendações de lojas de aplicativos, entretanto, não mostrou grandes tendências (Figura 2.6). Vale destacar, porém, o aumento no número de participantes que consideram as recomendações desse meio como satisfatórias (4).

Ao contrário do observado em outros meios, a recomendação de aplicativos por pessoas próximas foi considerada satisfatória (4), ou muito satisfatória (5) (Figura 2.7). De fato, mais de 65% das pessoas idosas, disseram considerar essa fonte de recomendação como muito satisfatória. Além disso, o restante desse grupo considera a recomendação satisfatória. Da mesma forma, essa fonte de recomendação obteve a menor quantidade de votos abaixo de razoável, sendo que ninguém a considerou pouco satisfatória.

Além disso, os participantes foram ainda perguntados acerca da quantidade de aplicativos que eles instalavam durante o período de seis meses, provenientes de recomendações de cada fonte. No que se refere às recomendações de redes sociais, a maior parte dos usuários relataram instalar cerca de 1 a 5 aplicativos no período, como

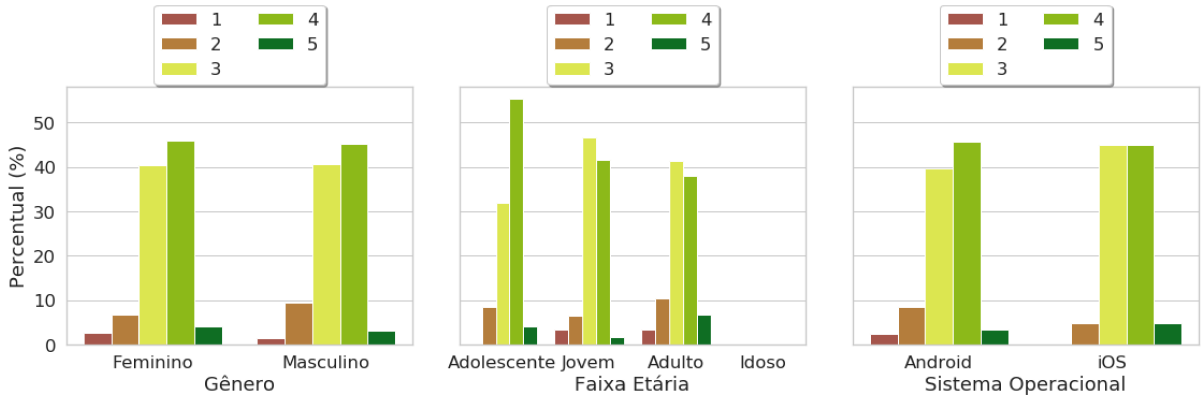


Figura 2.6: Satisfação percebida com a recomendação de *apps* em lojas de aplicativos.

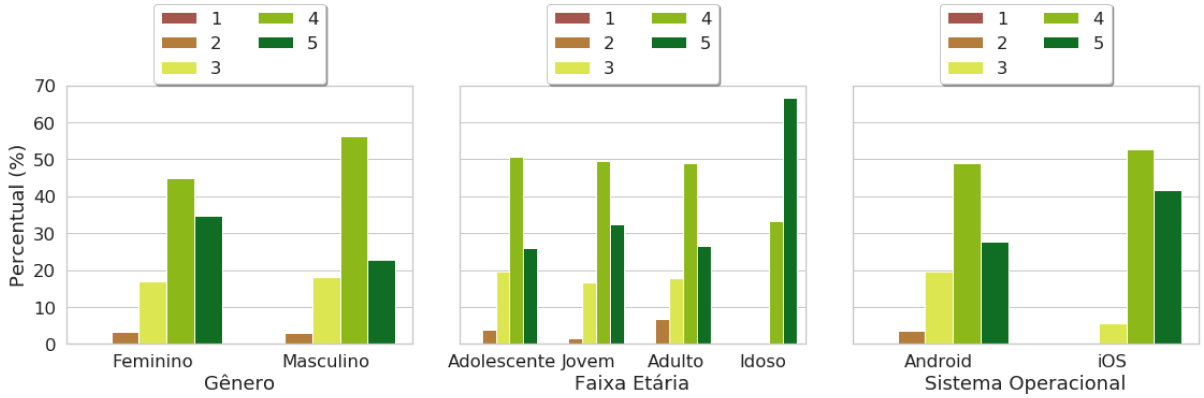


Figura 2.7: Satisfação quanto à recomendação de *apps* por pessoas próximas.

pode ser visto na Figura 2.8. Entretanto, pouco mais de 20% do público masculino relatou instalar de 6 a 10 aplicativos recomendados por esta fonte, fato que só se manifestou novamente ao analisar usuários *iOS*, onde cerca de 25% instalam de 6 a 10 aplicativos.

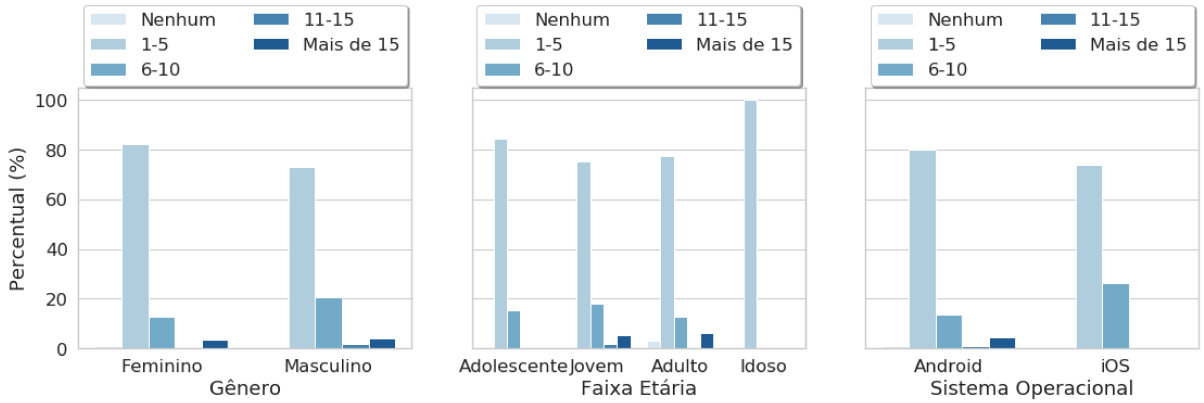


Figura 2.8: Número de aplicativos instalados a partir de redes sociais.

Da mesma forma, no que consta à quantidade de aplicativos instalados por recomendação de lojas de *apps* (Figura 2.9), vemos que a grande maioria dos participantes

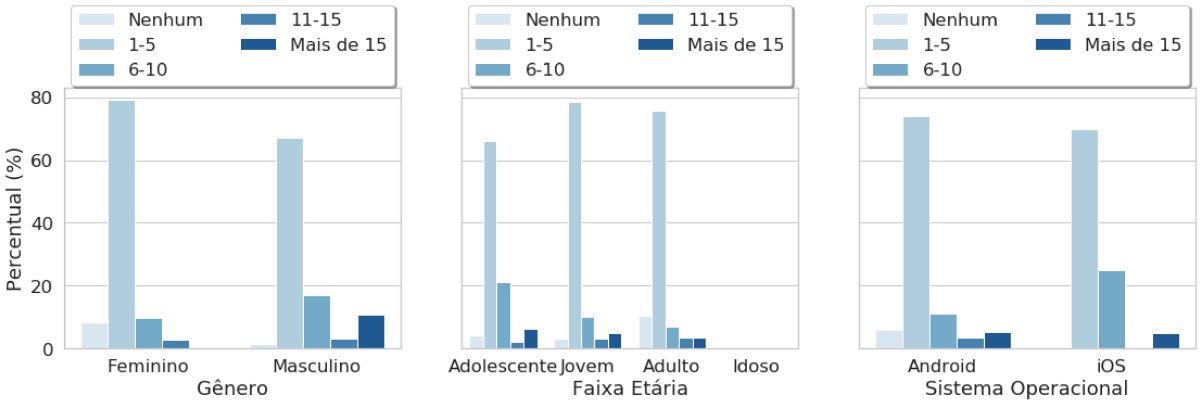


Figura 2.9: Número de aplicativos instalados por recomendações de lojas.

relatou instalar de 1 a 5 aplicativos. Entretanto, 10% das do gênero masculino declararam instalar mais de 15 aplicativos em seis meses, se recomendados por lojas. Isso pode indicar um maior direcionamento desse tipo de público para recomendações de lojas de *apps*. Além disso, somente participantes adolescentes e aqueles do grupo *iOS*, declararam instalar de 6 a 10 *apps* no período avaliado.

Já com relação ao número de *apps* instalados através de pessoas próximas, vemos que a grande maioria dos participantes declararam instalar apenas até 5 aplicativos (Figura 2.10). Além disso, dentre os grupos analisados, apenas cerca de 20% de adultos e pessoas do gênero masculino disseram instalar de 6 a 10 aplicativos. Vale ressaltar também que, pouquíssimas pessoas disseram instalar mais de 15 aplicativos.

Assim, os resultados mostrados destacam a importância de uma análise mais profunda acerca da motivação de instalação de usuários. Isso porque, com tal análise, é possível identificar perfis relacionados à fonte de recomendação. Além disso, a adição de informações de laços sociais parecem beneficiar a recomendação de aplicativos, já que o número de pessoas que instalam por essa fonte é grande, além de a satisfação possuir a maior média observada. Porém, parecem ser instalados poucos aplicativos provenientes de pessoas próximas, não sendo possível identificar se tal fato ocorreu

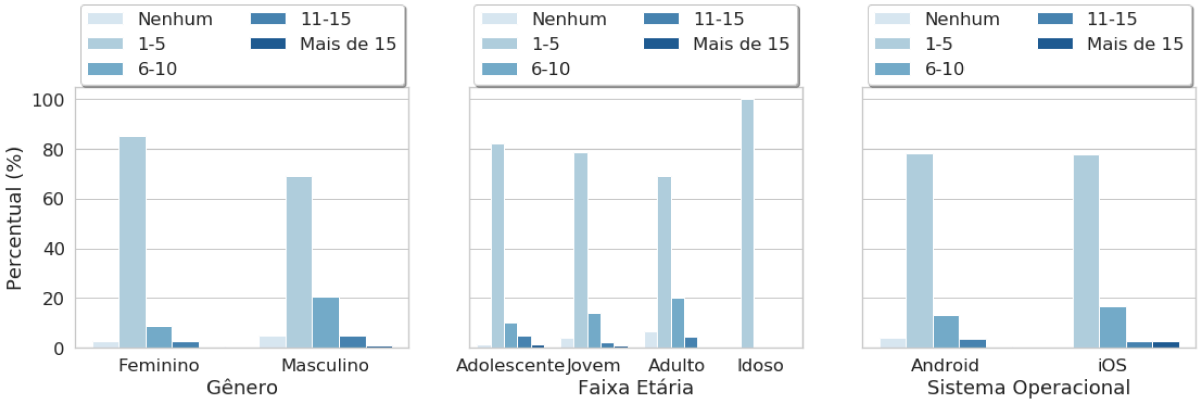


Figura 2.10: Número de *apps* instalados por recomendações de pessoas próximas.

por falta de interesse na instalação, ou simplesmente por serem recomendados poucos *apps*. Além disso, a adição de informações de gênero, faixa etária e sistema operacional parecem possibilitar uma melhor recomendação, sendo necessário maiores estudos acerca disso.

Além dos tópicos citados acima, os participantes também foram perguntados acerca das principais motivações que os levam a instalar algum aplicativo. As respostas obtidas podem ser vistas no gráfico da Figura 2.11. Dentre as motivações citadas, podemos destacar a aparente baixa relevância de tópicos como Interesse, Popularidade e Necessidade, abordados em muitos trabalhos (Frey et al., 2017; Xu et al., 2018; Peng et al., 2018). Por outro lado, motivações como Anúncios e Recomendações de outros usuários parecem ter grande impacto na escolha do aplicativo a ser instalado.

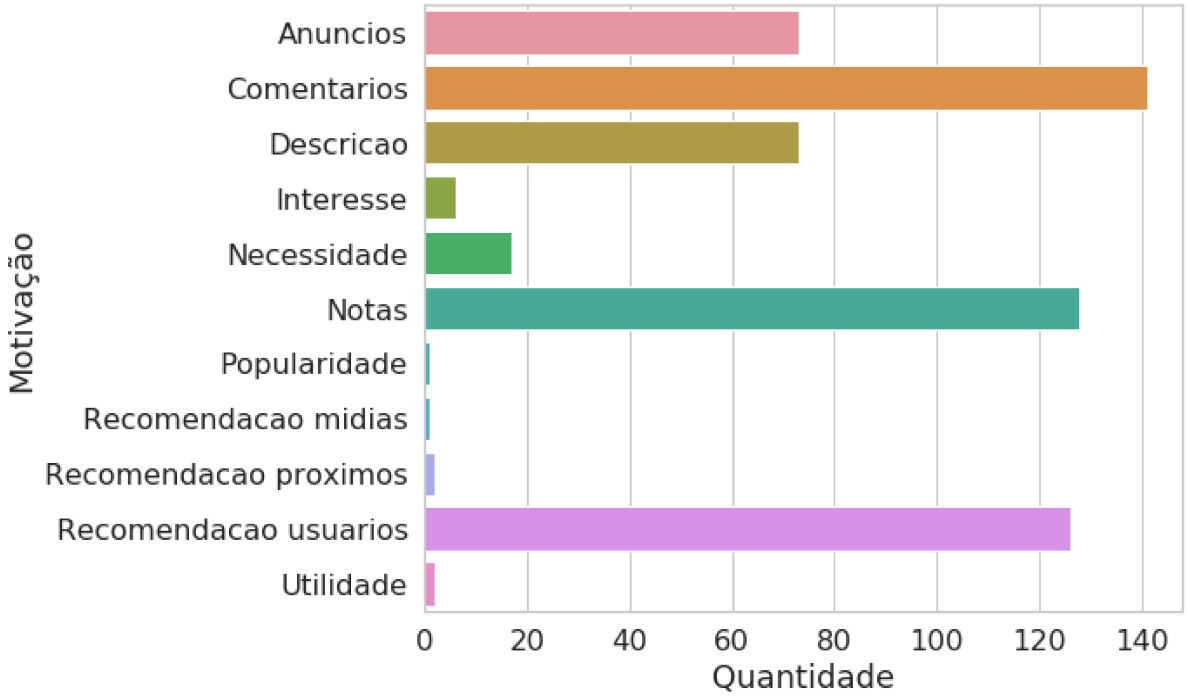


Figura 2.11: Motivações declaradas pelos usuários para instalar um aplicativo.

Quando perguntados sobre os principais fatores que os levavam à escolha de algum aplicativo, as principais respostas foram Segurança e Finalidade (Figura 2.12). Já dentre os fatores menos levados em consideração estão Preferência de permissões requisitadas e de anúncios presentes no *app*.

Portanto, com base nas respostas obtidas com a enquete, pode-se observar que anúncios de aplicativos com a finalidade correta podem gerar um impacto positivo na instalação. Isso porque recomendações corretas podem levar o usuário a perceber os benefícios de se utilizar um *app*, construindo uma relação de confiança com o sistema de recomendação. Entretanto, é preciso conhecer as finalidades esperadas por cada tipo de usuário, e mais ainda, se o aplicativo adequado está disponível e faz

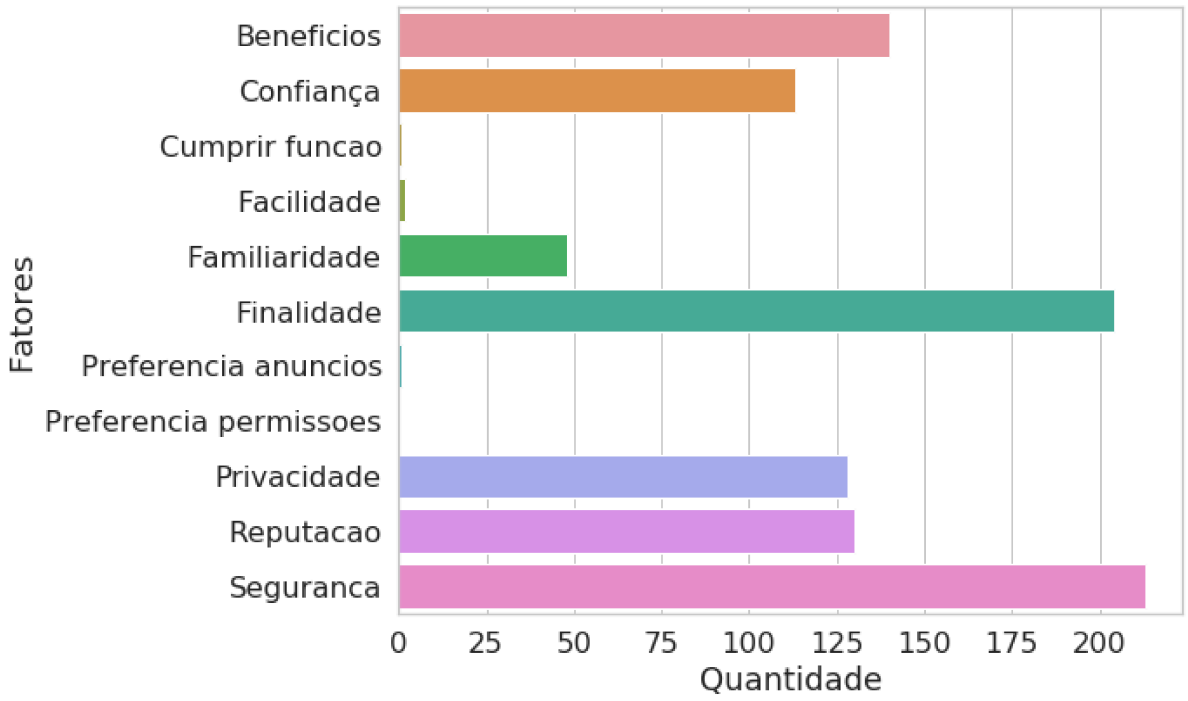


Figura 2.12: Outros fatores que levam à instalação de *apps*.

sentido com as características de tal usuário, como região de residência, renda e tipo de dispositivo.

2.4 Dados Demográficos

Com a utilização de dados demográficos nas mais diversas áreas, suas informações têm sido incorporadas em aplicações e rotinas anteriormente estabelecidas. Porém, primeiramente, é importante entender como e em quais situações tais dados são coletados, visto que o uso de tais características podem impactar nos resultados obtidos com seu uso.

Segundo a Associação Brasileira de Estudos Populacionais (ABEP), a demografia em si é uma ciência empírica que busca desenvolver análises para que se compreenda os fenômenos sofridos por uma população. Estes fenômenos podem ser entendidos como eventos que irão modificar a forma e comportamento de um determinado povo.

O censo é uma pesquisa realizada em subáreas geográficas, buscando coletar dados demográficos acerca dos moradores de cada domicílio ali localizados. As informações coletadas representam fonte de informação sobre as condições de vida características daquela região e são utilizadas para definir orçamentos e as ações governamentais que serão aplicadas àquela área, segundo a ABEP. No Brasil, o censo é realizado decenalmente pelo Instituto Brasileiro de Geografia e Estatística (IBGE), sendo que o último aconteceu em 2010. As informações coletadas são analisadas e é

disponibilizado seu resumo de acordo com as unidades territoriais: setores censitários (a menor delas), subdistritos, distritos, municípios, regiões metropolitanas, Microrregiões, Mesorregiões, Unidades da Federação, Grandes Regiões e para o Brasil como um todo (IBGE, 2021).

Dentre as informações coletadas no censo podemos citar: distribuição dos habitantes por sexo, raça e idade; rendimento médio de moradores (e.g., com renda, com ou sem renda); variância do número de moradores por residência, entre outras. Além disso, é importante ressaltar que tais dados podem ser obtidos levando em consideração cada tipo de unidade territorial. Assim, de posse de tais dados, é possível inferir um perfil médio de usuário, pois as características socioeconômicas do local são conhecidas. Entretanto, apesar do censo prover informações precisas, os dados levantados por ele podem se tornar obsoletos em um curto período de tempo.

O IBGE também fornece uma base de dados mais recente, o IBGE Cidades (IBGE, 2020), que contém informações socioeconômicas acerca de cada um dos 5.570 municípios do território brasileiro. As informações contidas nesta base foram levantadas através de um estudo realizado em 2018, disponibilizando indicadores como: número de habitantes, PIB *per capita*, taxa de concentração urbana, indicativos das atividades que mais contribuem com a economia local, dentre outras. Apesar de ser uma base com informações mais recentes, os dados disponíveis dizem respeito somente a cidades como um todo, não sendo possível obter informações específicas acerca de unidades territoriais menores (setores censitários, subdistritos e distritos).

2.5 Trabalhos Relacionados

Nesta seção são apresentados os principais trabalhos da literatura que envolvem a recomendação de aplicativos móveis a usuários.

Com a necessidade de se recomendar aplicativos aos usuários, muitos trabalhos foram desenvolvidos para tal fim, fazendo com que surjam diversas abordagens e estratégias. A fim de melhor recomendar aplicativos aos usuários, os trabalhos dos autores Frey et al. (2017) e Cheng et al. (2018) utilizam LDA (*Latent Dirichlet Allocation*) para fazer as recomendações. O primeiro utilizou LDA para selecionar os tópicos principais dentre as descrições de aplicativos, utilizando a probabilidade de o usuário gostar de cada tópico como insumo para um modelo baseado no algoritmo *Floresta Aleatória*. Já os autores do segundo trabalho utilizam a ordem de instalação dos aplicativos para observar três aspectos, sendo eles: contextos de curto-prazo, onde é estimada a probabilidade de um usuário instalar um aplicativo, dado outros aplicativos que ele possui; padrões de co-instalação, onde analisou-se quais aplicativos normalmente são instalados em conjunto, aplicando LDA; e instalações aleatórias onde são indicados aplicativos populares com uma grande chance de serem aceitos

Tabela 2.1: Categorização dos trabalhos relacionados

Trabalho	**Filtro Colaborativo**	**Outras Técnicas**	**Detalhes**
Frey et al. (2017)		X	LDA e Floresta Aleatória
Cheng et al. (2018)		X	LDA
Pan et al. (2011)		X	Grafo
Xu et al. (2018)		X	Grafo e *PageRank*
Ma et al. (2016)		X	Modificação de *Word2Vec*
Yin et al. (2017)	X	X	*Bag of Words* e Tópicos Latentes
Peng et al. (2018)	X		Matriz de Fatorização
Liu et al. (2015)	X		Fatores Latentes
Liu et al. (2016)	X		Fatores Latentes

pelo usuário.

Outros dois trabalhos utilizam grafos para representar associações entre usuários. No trabalho de Pan et al. (2011) são utilizadas informações de sensores de *smartphones* para construir um grafo que representa ligações entre usuários, possibilitando o cálculo do potencial de instalação de um aplicativo com base nos vizinhos de um determinado usuário. Neste caso, é necessário que seja obtido acesso a dados de requisitos muitas vezes bloqueados por usuários comuns, impossibilitando sua utilização em larga-escala. Já o trabalho elaborado por Xu et al. (2018) considera a funcionalidade de cada aplicativo, fazendo com que seja possível predizer quais as próximas necessidades do usuário através de um grafo de coocorrência.

O trabalho desenvolvido por Ma et al. (2016) realiza uma alteração no algoritmo *Word2Vec*, visando predizer a instalação de aplicativos com base na utilização recente de outros. Já o trabalho desenvolvido por Yin et al. (2017) utiliza preferências de permissões para indicar aplicativos, por meio da descrição e permissões de cada aplicativo. Para isso, são utilizados *Tópicos Latentes* para caracterizar os interesses de um usuário e relacioná-los às preferências de permissões para cada categoria de aplicativo.

Outros trabalhos levam em consideração as preferências de privacidade dos usuários. O trabalho de Peng et al. (2018) identifica os aplicativos que requisitam muitas permissões e possuem classificação inferior, dando a eles baixa prioridade. Os aplicativos que passam por este filtro são combinados aos interesses dos usuários através de uma matriz de fatoração. Da mesma forma, o trabalho de Liu et al. (2015) também relaciona as preferências de privacidade e de comportamento do usuário. Entretanto, estes fatores são relacionados em um perfil latente. Posteriormente, os autores modificaram este trabalho adotando uma nova estratégia, onde é analisada a categoria e funcionalidade de cada aplicativo Liu et al. (2016). Mais uma vez, são utilizados vetores latentes, agora relacionando as funcionalidades de cada aplicativo aos interesses

do usuário através da lista de instalações deste.

A Tabela 2.1 apresenta uma comparação entre os trabalhos existentes. O grande dificultador dos trabalhos apresentados é o acesso em larga escala às informações muitas vezes utilizadas, como dados de preferências de privacidade e acesso aos *logs* de ligações. Por outro lado, esta pesquisa busca utilizar uma base de dados reais contendo somente os aplicativos instalados pelo usuário, assim como dados de algumas de suas localizações e a informação de qual dispositivo móvel o usuário utiliza. Tais dados, mais simples de serem obtidas, podem prover uma série de outras informações através do enriquecimento de dados. Tal enriquecimento será dado através da utilização da localização, para incorporar informações contextuais demográficas acerca do usuário. Além disso, saber qual o dispositivo utilizado, pode nos levar a conhecer seu preço (Maia et al., 2020), e portanto qual a categoria de tal dispositivo (e.g., entrada, *premium*, entre outras).

Por fim, como não foram encontrados trabalhos que utilizam dados demográficos para melhorar a recomendação de aplicativos, foram utilizadas as abordagens LDA, Filtro Colaborativo e Cadeias de *Markov* como soluções base. O Filtro Colaborativo foi escolhido por ser uma abordagem bastante utilizada quando se trata de sistemas de recomendação, no geral. Já as abordagens LDA e Cadeia de *Markov* foram escolhidas por apresentarem bons resultados na recomendação de aplicativos, como mostrado por Cheng et al. (2018).

Capítulo 3

Análise Preliminar

Para que se pudesse identificar previamente a eficácia do uso de dados demográficos durante a recomendação de aplicativos móveis, foi realizada a análise preliminar descrita neste capítulo. Com caráter exploratório, esta análise utilizou dados e abordagens iniciais no que compete à estruturação de um sistema de recomendação, tornando possível a escolha mais direcionada do caminho a ser seguido futuramente.

3.1 Dados

Para realizar o estudo preliminar descrito nesta seção, foi utilizado um conjunto de dados reais de usuários *Android*, obtidos sob confidencialidade de uma empresa provedora de serviços móveis. Estes dados foram coletados do dia 04 de Junho de 2019 ao dia 08 de Agosto de 2019. Para garantir a coerência dos resultados obtidos, foram utilizados 7.406 usuários que participaram da coleta em todo o período, retirando usuários que deixaram a coleta em algum momento, e aqueles que ingressaram durante o período.

Durante o período da coleta, a cada novo dia foi gerado um registro por usuário, contendo uma lista de todos os aplicativos que o mesmo possuía instalado. Entretanto, por se tratar de uma análise experimental, foram utilizados somente os 249 aplicativos mais relevantes em termos de popularidade dentre o conjunto de dados obtido. Estes aplicativos puderam ser agrupados em 28 categorias, tais como Bancos Digitais, Esportes, E-Commerce, dentre outros. As categorias foram elencadas através de critérios técnicos, uma vez que as categorias obtidas pela *Play Store* possuíam inconsistências. Isto porque, como a categoria atribuída a cada aplicativo depende do desenvolvedor, seu critério de escolha é subjetivo, fazendo com que possa haver aplicativos de mesmo propósito mas alocados em categorias distintas. Ao mesmo tempo, aplicativos de mesma categoria podem ter propósitos completamente diferentes (e.g., *apps* de jogos de tratamento facial categorizados como Beleza e não como Jogo).

Para a criação do modelo preliminar, os dados referentes ao primeiro dia da coleta (4 de Junho) foram utilizados para o treinamento, atribuindo os dados dos demais dias ao conjunto de teste. Na Figura 3.1, é possível ver a quantidade de usuários que cada categoria possui no dia de treino e nos dias de teste. É possível ver que o número

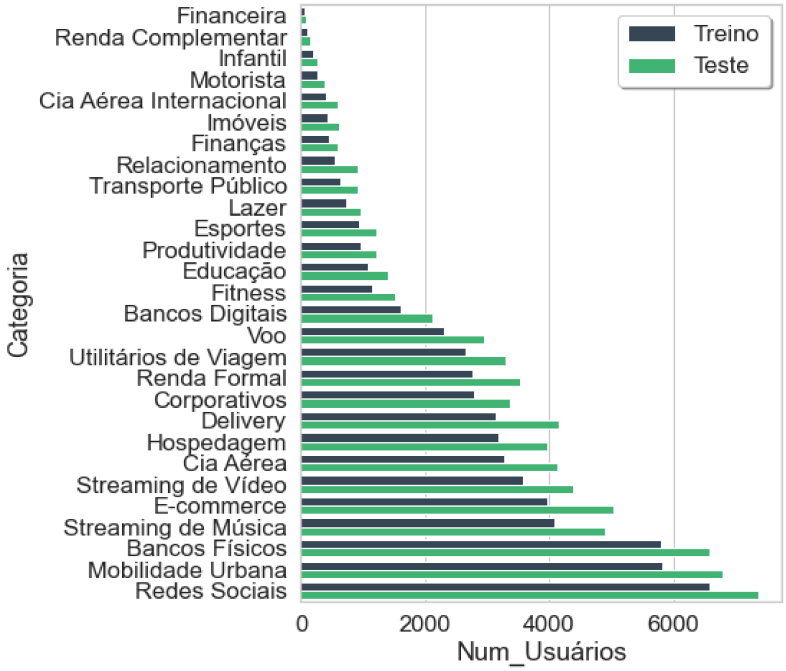


Figura 3.1: Número de usuários por categoria nas fases de treino e teste.

de usuários aumentou em todas as categorias durante o período avaliado, indicando a instalação destas por outros usuários. Com uma solução de recomendação, espera-se que uma parcela de tais usuários seja detectada antes da instalação.

Além dos aplicativos existentes no dispositivo móvel, também foi fornecida uma localização aproximada da residência dos usuários. A localização aproximada foi utilizada por se tratar de um dado mais fácil de ser obtido, em comparação com localizações exatas, protegendo a privacidade do usuário em questão.

3.1.1 Dados Demográficos

Para realização deste estudo, os dados foram enriquecidos com as informações dos setores censitários coletados da base de dados do IBGE (IBGE, 2021), referentes ao local de residência dos usuários. A base de dados demográficos foi formatada, preparada e organizada para que métricas relevantes fossem extraídas. Ao todo, foram selecionadas as 24 (Tabela 3.1) informações que apresentavam potencial de estarem relacionadas à decisão de se instalar um aplicativo ou não.

3.2 Métricas

Para avaliar a qualidade da proposta, foram calculadas algumas métricas que fazem referência aos conjuntos listados a seguir. Sendo U o conjunto de todos os usuários, seja $U_{i,a}$ o conjunto de todos os usuários que realmente instalaram o aplicativo a

Tabela 3.1: Características demográficas utilizadas para construção dos modelos.

Demografia	**Categoria**
Número de domicílios	Social
Número de moradores	Social
Média de moradores por domicílio	Social
Número de mulheres	Sexo
Número de mulheres alfabetizadas	Educação
Número de homens	Sexo
Número de homens alfabetizados	Educação
Número de pessoas brancas	Raça
Número de pessoas pretas	Raça
Número de pessoas pardas	Raça
Número de pessoas amarelas	Raça
Número de pessoas indígenas	Raça
Número de pessoas com Idade <= 10	Idade
Número de pessoas com 11 <= Idade <=20	Idade
Número de pessoas com 21 <= Idade <= 30	Idade
Número de pessoas com 31 <= Idade <= 40	Idade
Número de pessoas com 41 <= Idade <= 50	Idade
Número de pessoas com 51 <= Idade <= 60	Idade
Número de pessoas com 61 <= Idade <= 70	Idade
Número de pessoas com 71 <= Idade <= 100	Idade
Média de rendimento médio por morador com ou sem renda	Rendimento
Média de rendimento médio por morador com renda	Rendimento
Média de rendimento médio por morador responsável com renda	Rendimento
Média de rendimento médio por morador responsável com ou sem renda	Rendimento

durante o período de teste e $U_{r,a}$ o conjunto de todos os usuários que o modelo recomendou que iriam instalar o aplicativo a.

- **Precisão:** porcentagem do total de recomendações assertivas em relação ao total de instalações previstas do aplicativo a.

$$Precisao_a = \frac{|U_{i,a} \cap U_{r,a}|}{|U_{r,a}|} \tag{3.1}$$

- **Revocação:** do total de instalações que realmente ocorreram do aplicativo a, quantas foram previstas corretamente.

$$Revocacao_a = \frac{|U_{i,a} \cap U_{r,a}|}{|U_{i,a}|} \tag{3.2}$$

- ***F-Score*:** média harmônica da precisão e revocação do aplicativo a.

$$F\_Score_a = 2 * \frac{Precisao_a * Revocacao_a}{Precisao_a + Revocacao_a} \tag{3.3}$$

Por ser um problema com taxas de precisão e revocação naturalmente baixas, foi utilizada também a métrica de *Lift*, que indica a taxa de levantamento de um modelo em relação a outro. Se o valor do *Lift* for positivo, o modelo descreve melhor os dados do que o modelo de base; se negativo, o modelo não atingiu os mesmos resultados que o modelo de base. Nesta pesquisa, utilizamos esta técnica para avaliar tanto a solução base, quanto a proposta, em relação à estratégia de selecionar usuários aleatórios, verificando quantas vezes as soluções se saíram melhor em relação à uma escolha arbitrária. Assim:

- ***Lift*:** representa o ganho (ou perda) de precisão ao se utilizar determinado modelo, se comparado ao uso de amostras aleatórias de tamanho r.

$$Lift_a = \frac{Precisao_a - \overline{Precisao_s}}{\overline{Precisao_s}} \tag{3.4}$$

  onde $\overline{Precisao_s}$ é a precisão média de diferentes amostras aleatórias.

Para o cálculo do Lift médio de um aplicativo, foi utilizada a precisão média de 30 amostras geradas aleatoriamente. Este processo foi repetido para diferentes tamanhos de amostra (8%, 9%, 10%, 12%, 15%, 20%, 30%, 40%, 50% e 60% do total de usuários), buscando assim averiguar o comportamento geral dos modelos.

3.3 Modelos

Nesta seção serão descritos os detalhes da solução proposta, denominada ANCESTOR (*Application aNd CEnsus baSed recommendaTion algORithm*), assim como aqueles referentes ao modelo utilizado como solução base, chamado ALBERTA (*AppLication BasEd RecommendaTion Algorithm*). ALBERTA, utiliza uma abordagem tradicional de filtro colaborativo para esse problema, e considera como itens apenas os aplicativos que os usuários têm instalado. Já o ANCESTOR, utiliza, além dos aplicativos, 24 características demográficas, obtidas através do IBGE, como itens. Dessa forma, a hipótese de que os dados demográficos representam informações relevantes para a previsão de instalação de aplicativos pode ser validada ou refutada.

Filtro Colaborativo é uma abordagem presente em vários sistemas de recomendação e que utiliza da semelhança entre usuários ou itens para escolher os melhores candidatos à recomendação. Nesta estratégia, é construído um banco de dados com todos os usuários e suas respectivas preferências por itens diversos e, quando se deseja recomendar um item a algum usuário, seus interesses são comparados com os de todos os outros usuários. Dessa forma, o algoritmo irá recomendar o item que obtiver melhor avaliação dentre todos os usuários semelhantes ao usuário alvo, desde que este não o tenha avaliado anteriormente. Dentre as ramificações do Filtro Colaborativo, podemos citar dois tipos de abordagens possíveis: *Model-based* e *Memory-based* (Su and Khoshgoftaar, 2009).

Filtros colaborativos *Memory-based* são aqueles que utilizam de cálculos de similaridade entre usuários e/ou itens para realizar suas recomendações. Esta estratégia tem se mostrado efetiva em vários cenários e é mais fácil de ser implementada do que filtros colaborativos *Model-based*. Entretanto, como lida com dados esparsos, deve-se atentar à escalabilidade tanto do número de usuários, como da quantidade de itens (Su and Khoshgoftaar, 2009). Esta estratégia também pode ser utilizada para tratar de abordagens estocásticas, onde acredita-se que somente o estado atual do usuário é suficiente para indicar ações futuras.

Já a abordagem *Model-based* é indicada a situações onde o desempenho é um fator decisivo, pois utiliza de modelos de aprendizado de máquina, agrupamento, matrizes de fatoração, entre outras técnicas para entender os padrões comuns a usuários.

Como esta análise inicial contará com uma quantidade reduzida de dados tanto de usuários como de itens (*apps*), utilizaremos a abordagem *Memory-based*. Esta também pode ser subdividida entre duas técnicas: *Usuário-Item* e *Item-Item*. A técnica *Usuário-Item* utiliza uma matriz de $Usuarios \times Itens$ como base de dados para encontrar os vizinhos. Entretanto, tal método não apresenta boa escalabilidade em relação ao crescimento de usuários, principalmente caso haja interesse em conhecer alguns milhares de vizinhos de algum usuário a fim de fazer recomendações mais precisas (Sarwar

et al., 2001).

Já na abordagem *Item-Item*, a similaridade é primeiramente observada no que se refere aos itens, criando uma matriz simétrica de $Itens \times Itens$ como base de dados. Como o relacionamento entre itens tende a não crescer de forma tão rápida, filtros colaborativos *Item-Item* podem ser capazes de prover a mesma qualidade da abordagem *Usuário-Item* com menor custo computacional e melhor escalabilidade.

3.3.1 Modelo ALBERTA

Nesta fase da pesquisa, foi utilizada a abordagem *Item-Item* pois seu tempo de execução não cresce assintoticamente com o aumento da quantidade de usuários, já que é esperado que se tenha um número significativamente maior destes, em comparação com o número de aplicativos. Outro motivo da escolha deste método é que assumimos ser mais relevante saber que dois aplicativos possuem alta correlação e, portanto, se um usuário possuir um deles, o outro deverá ser recomendado, do que a informação de que dois usuários são correlacionados. Isso porque, sabendo que um aplicativo é altamente correlacionado a outro, podemos identificar padrões relativos à própria natureza de tais *apps*, podendo agrupá-los de certa forma. Além disso, com a adição de dados demográficos, será possível identificar *apps* que sejam mais relacionados a certos contextos demográficos. Assim, para a solução base ALBERTA, cada item representará um aplicativo.

Logo, assumindo U como o conjunto de usuários, e A como o conjunto de todos os aplicativos avaliados, primeiramente é criada uma matriz $M = |U| \times |A|$ preenchida com 1 caso o usuário tenha determinado aplicativo instalado, e 0 caso contrário. A Tabela 3.2 ilustra um exemplo simplificado da matriz de entrada M, onde os usuários estão representados nas linhas, e os aplicativos nas colunas. Neste exemplo podemos ver que o usuário u_1 possui os aplicativos a_1, a_2 e a_4.

Tabela 3.2: Exemplo de matriz M

	a_1	a_2	a_3	a_4
u_1	1	1	0	1
u_2	1	1	1	0
u_3	0	1	0	1

Em seguida, é utilizado o cálculo de similaridade de cossenos na matriz M, resultando na matriz simétrica $S = |A| \times |A|$, em que valores próximos a 1 indicam aplicativos correlacionados; próximos a 0, *apps* não correlacionados; e próximos a -1, que tais aplicativos possuem alta correlação inversa. A similaridade de cossenos foi utilizada por ser uma métrica que lida bem com estruturas de dados esparsas, uma vez que desconsidera se os dois usuários avaliados são "semelhantes" em não terem

determinado aplicativo instalado, caso que resultaria na similaridade de vários usuários (Tan et al., 2006). A fórmula abaixo representa o cálculo da similaridade entre dois aplicativos (i e j) quaisquer, onde $\left\|\vec{i}\right\|$ e $\left\|\vec{j}\right\|$ representam o módulo dos vetores $\vec{i}$ e $\vec{j}$, respectivamente.

$$S = \cos(\vec{i}, \vec{j}) = \frac{\vec{i}.\vec{j}}{\left\|\vec{i}\right\| . \left\|\vec{j}\right\|}$$

A Tabela 3.3 ilustra o resultado da similaridade com base na matriz M de exemplo da Tabela 3.2. Nota-se que os valores da diagonal principal assumem sempre o valor 1, pois um aplicativo é sempre altamente similar a si próprio. Além disso, podemos perceber que os pares de aplicativos (a_1, a_2) e (a_2, a_4), são os que possuem maior correlação.

Tabela 3.3: Matriz de similaridade de cosseno obtida

	a_1	a_2	a_3	a_4
a_1	1	0,82	0,71	0,5
a_2	0,82	1	0,58	0,82
a_3	0,71	0,58	1	0
a_4	0,5	0,82	0	1

Para se identificar quais aplicativos serão passíveis de recomendação ao usuário, é feita a multiplicação da matriz de similaridade S por uma matriz M de $Usuarios \times Aplicativos$, de acordo com a fórmula abaixo.

$$R = S \times M \tag{3.5}$$

A matriz R, também de formato $Usuarios \times Aplicativos$, indica os *scores* de recomendação de cada aplicativo para cada usuário. Nela, valores próximos a 1 indicam uma possível instalação pelo usuário correspondente. A Tabela 3.4 ilustra a matriz resultante R do exemplo apresentado.

Tabela 3.4: Matriz de previsão das instalações conforme exemplo

	a_1	a_2	a_3	a_4
u_1	0,76	0,81	0,56	0,99
u_2	0,83	0,74	0,99	0,57
u_3	0,43	0,56	0,25	0,78

Com base na matriz R, é escolhido um ponto de corte que será utilizado para separar os aplicativos que serão efetivamente recomendados. Tal ponto de corte pode representar tanto situações onde são escolhidos os N aplicativos com maior *score*, quanto valores numéricos onde são recomendados todos os *apps* com *score* de insta-

lação maior que ele. Nesta análise foi utilizado o valor correspondente a três vezes a média da tabela de previsões de instalação, escolhido empiricamente.

Neste trabalho, a solução base ALBERTA utiliza apenas os indicadores dos 239 aplicativos instalados como itens de entrada da matriz M.

3.3.2 Modelo ANCESTOR

A solução proposta ANCESTOR utiliza, além dos dados do modelo ALBERTA, os 24 valores demográficos do IBGE (Tabela 3.1), normalizados entre 0 e 1 para ficarem coerentes com a base de instalação, acrescidos na tabela M da primeira etapa. Com isso, são acrescentados os níveis de similaridade entre dados demográficos e os aplicativos. Para as duas soluções, os passos do algoritmo são os mesmos, sendo que a diferença entre elas é a quantidade e as características dos itens utilizados.

3.4 Resultados

De acordo com as métricas utilizadas, os valores são calculados para cada aplicativo individualmente. A Figura 3.2 mostra a distribuição da precisão, revocação e f-score considerando todos os aplicativos. Como pode-se ver, o ANCESTOR, no geral, se sobressai ao modelo ALBERTA. Isso mostra que a adição das informações demográficas do IBGE contribuíram para um maior acerto da predição das instalações da maioria dos aplicativos, tanto do ponto de vista de acerto dos aplicativos recomendados, quanto do ponto de vista dos aplicativos que realmente foram instalados. Inclusive, observamos que a precisão máxima alcançada pelo modelo ANCESTOR é quase o dobro da alcançada pelo ALBERTA. Mesmo que este comportamento se dê apenas para alguns aplicativos e não represente a maioria, a vantagem obtida para estes pode ser

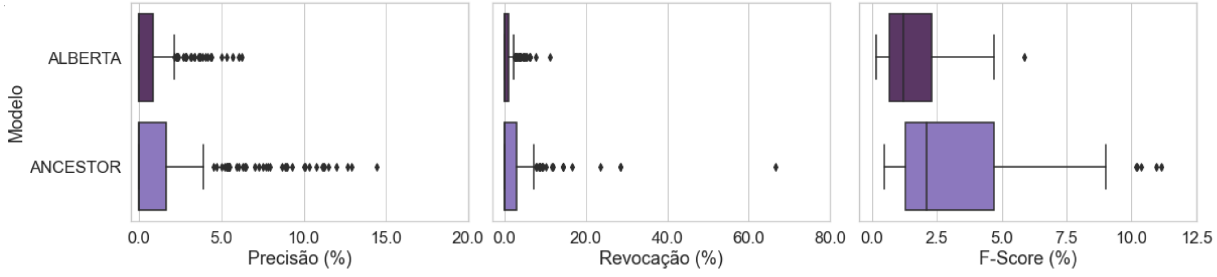


Figura 3.2: Precisão, revocação e f-score dos aplicativos.

Apesar de a precisão ser baixa em se tratando de modelos preditivos, algumas observações são importantes para o contexto de recomendação de instalação de aplicativos. Primeiramente, pode-se perceber que ao se acrescentar as informações demográficas dos usuários, a precisão aumentou para a maioria dos aplicativos. Isso

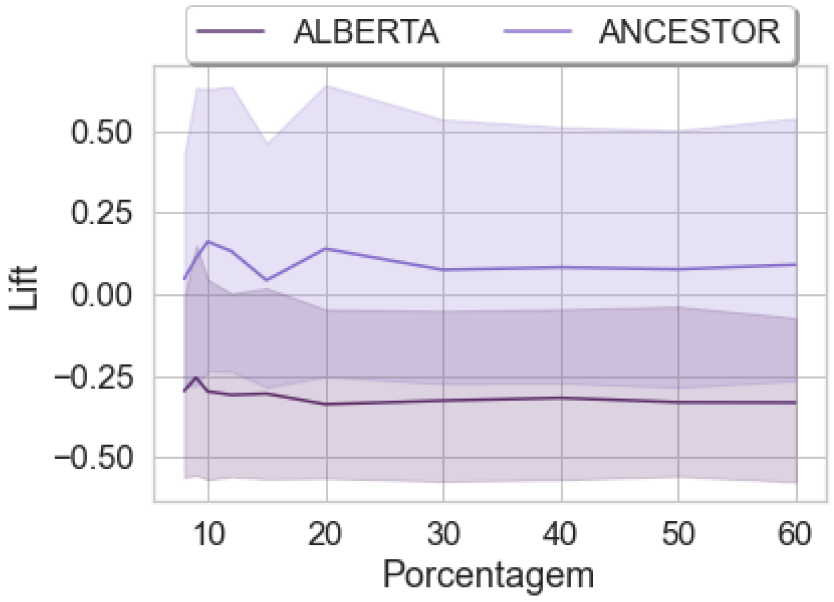


Figura 3.3: Curva do lift médio para todos os tamanhos de amostra, com intervalo de confiança de 95%.

mostra que a similaridade dos usuários em termos das características demográficas é um fator relevante.

Outra consideração diz respeito à taxa de conversão ou retorno quando uma campanha é feita para atrair novos usuários para determinado aplicativo. Pode ser que, mesmo com uma precisão baixa, a taxa de aumento (i.e., *Lift*) ao ter como alvo usuários recomendados pelo modelo seja maior do que ao se utilizar amostras aleatórias de usuários. Ou seja, o retorno do investimento por usuário alcançado será maior. O resultado de tal análise pode ser visto na Figura 3.3. Nela é possível perceber que o *Lift* não possui grande variação para as mudanças de tamanho da amostra (eixo-x), além de que o ANCESTOR possui, no geral, os melhores resultados. Além disso, vemos que mesmo com o intervalo de confiança, o ANCESTOR sempre possui uma taxa de levantamento maior que o ALBERTA.

Esses resultados mostram que, em geral, ao se utilizar o modelo de recomendação ANCESTOR, a taxa de retorno tende a ser mais efetiva do que ao se utilizar amostras aleatórias de usuários como alvo. Ou seja, a chance de um usuário recomendado pelo modelo ANCESTOR instalar determinado aplicativo após uma campanha de marketing direcionada a ele é maior do que se a campanha fosse direcionada a usuários escolhidos aleatoriamente.

Para melhor visualizar o comportamento dos modelos em relação aos aplicativos, calculamos a precisão média para cada categoria de aplicativo. Com isso, esperamos poder identificar o motivo de a precisão de alguns aplicativos aumentarem relativamente com a adição de dados demográficos. Na Figura 3.4, pode ser observado que a maior parte das categorias com maior precisão média dependem de fatores demográficos, como por exemplo aplicativos de mobilidade urbana e *delivery*, que estão

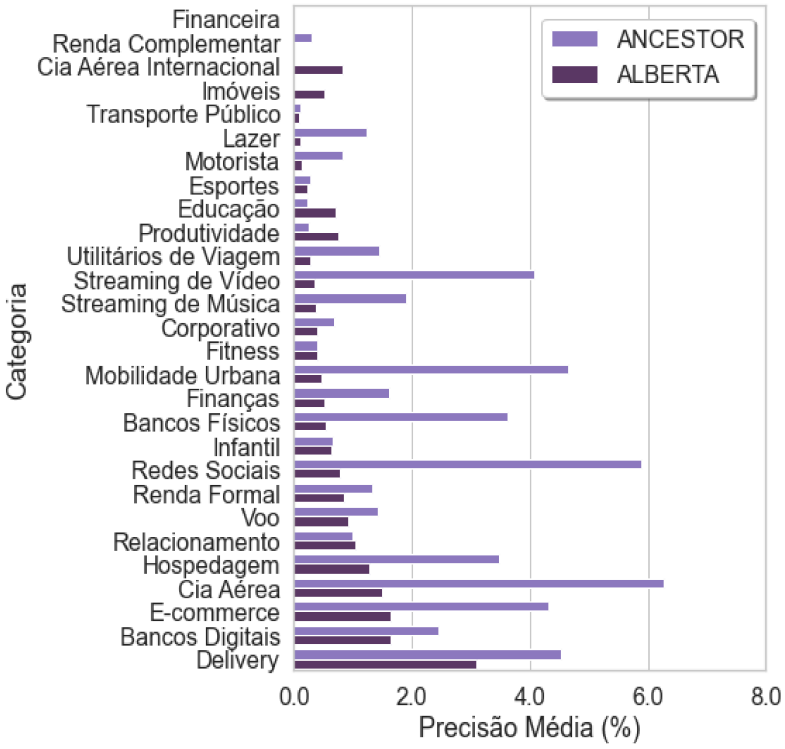


Figura 3.4: Precisão média por categoria.

mais presentes em cidades ou locais de alto índice populacional. Já aplicativos referentes a companhias aéreas, *streaming* de vídeo ou música, hospedagem e *e-commerce* podemdependem de outros fatores, como a renda, por exemplo. Outras categorias de aplicativos, como aplicativos infantis e de relacionamento, não dependem tanto dos fatores demográficos do local e, portanto, o ANCESTOR não apresenta uma melhora significativa em relação ao ALBERTA.

Já com relação à revocação, vemos que o ANCESTOR também se sobressai se com-

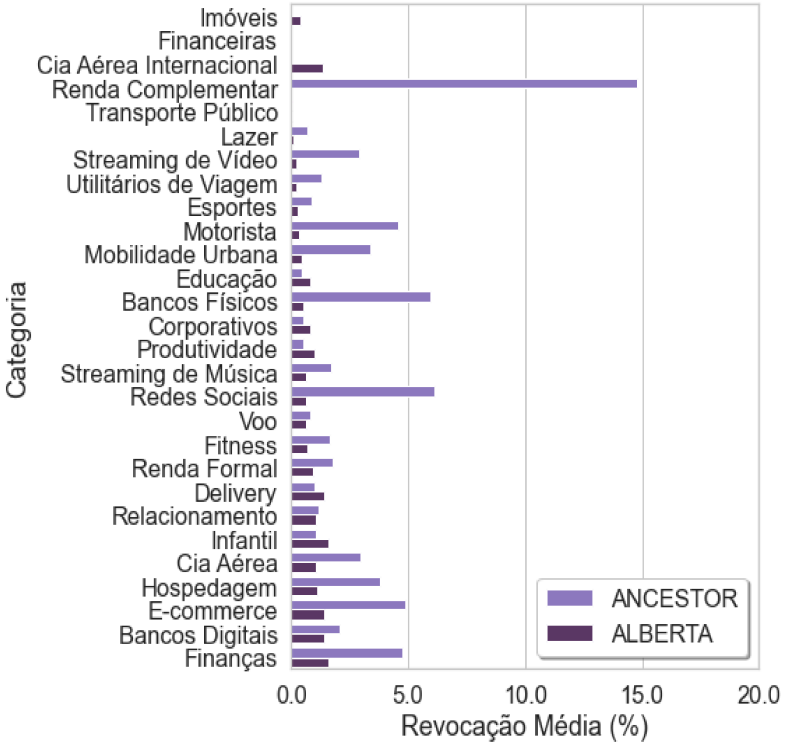


Figura 3.5: Revocação média por categoria.

parado ao ALBERTA, como vemos na Figura 3.5, indicando que esse tende a recomendar mais *apps* que serão efetivamente instalados. Aqui reforçamos também algumas categorias como mobilidade urbana, bancos físicos e redes sociais, que também apresentaram bons índices para precisão. Por outro lado, outras categorias obtiveram destaque somente na revocação, como *apps* de renda complementar e de motoristas, ambos que podem ser relacionados a fatores como a renda, por exemplo. O fato de que o ANCESTOR alcançou melhores resultados na revocação, se comparado à precisão, talvez possa ser explicado pela quantidade de pessoas que instalaram aplicativos dessas categorias, uma vez que, se poucos usuários efetivamente instalaram um *app*, boa parte destes podem ter sido alcançados, mesmo que o ANCESTOR tenha recomendado tal aplicativo a mais usuários.

Para apresentar mais detalhes, a Tabela 3.5 mostra os 20 aplicativos com melhor precisão do modelo ALBERTA. Pode-se perceber que a maior parte dos melhores resultados são referentes a aplicativos que obtiveram relativamente poucas instalações ou são concorrentes diretos de outros aplicativos considerados os principais de seus segmentos, como por exemplo *James*, *Uber Eats* e *Rappi* (concorrentes do *iFood*), e *Happn* (concorrente do *Tinder*).

Tabela 3.5: Tabela das estatísticas dos 20 aplicativos que obtiveram melhor precisão para o modelo ALBERTA.

Aplicativo	Total Instalações	Precisão ALBERTA	Precisão ANCESTOR	Revocação ALBERTA	Revocação ANCESTOR	Lift ALBERTA	Lift ANCESTOR
Udemy	54	6,25%	0,00%	3,70%	0,00%	7,65	-1,00
James	52	6,06%	0,00%	3,85%	0,00%	7,48	-1,00
Happn	131	5,67%	1,41%	6,11%	1,53%	2,21	-0,20
AliExpress	449	5,31%	7,05%	2,90%	3,56%	-0,13	0,16
Uber Eats	759	5,00%	6,45%	1,45%	1,05%	-0,51	-0,37
KayaK	150	4,40%	0,00%	2,67%	0,00%	1,14	-1,00
Next	65	4,35%	0,99%	4,62%	1,54%	3,91	0,13
FGTS	332	4,17%	3,05%	2,11%	1,51%	-0,07	-0,32
KLM	20	4,00%	0,00%	0,05%	0,00%	14,71	-1,00
Slack	23	3,85%	0,00%	4,35%	0,00%	11,18	-1,00
Rappi	420	3,70%	5,00%	1,43%	0,71%	-0,34	-0,12
Netshoes	207	3,69%	5,46%	3,86%	6,28%	0,34	0,98
Air France	27	3,57%	0,00%	3,70%	0,00%	9,45	-1,00
Centauro	95	3,36%	0,46%	5,26%	1,05%	1,67	-0,65
DAZN	100	3,33%	1,78%	4,00%	3,00%	1,46	0,32
Wish	443	3,16%	5,38%	1,81%	4,51%	-0,47	-0,11
Google Analytics	13	3,12%	0,00%	7,69%	0,00%	17,68	-1,00
Copa Airlines	37	2,86%	0,00%	5,41%	0,00%	4,76	-1,00
Trivago	235	2,82%	1,96%	2,55%	0,85%	-0,11	-0,39
Alelo	325	2,80%	6,28%	2,15%	4,62%	-0,36	0,43

Por outro lado, mesmo entre os aplicativos com melhor precisão para o ALBERTA, alguns (como AliExpress, Alelo, Uber Eats e Netshoes), possuem uma precisão menor do que ANCESTOR. Dentre os piores desempenhos do ANCESTOR, estão os aplicativos de companhias aéreas internacionais. Isso pode ser explicado pelo fato de que não haviam muitos dados acerca deste tipo de aplicativo, sendo somente 413 usuários no

período utilizado para treino, em comparação aos 3281 usuários que possuíam aplicativos de companhias aéreas nacionais no mesmo período. Esta explicação casa com o fato de que não há aplicativos de companhias aéreas nacionais dentre os melhores para o modelo ALBERTA.

Diferentemente do ALBERTA, o ANCESTOR se sai melhor dentre os aplicativos que possuem alto índice de instalação, chegando a 14,45% de precisão, como vemos na Tabela 3.6. Também pode-se perceber que a menor precisão indicada nesta tabela ainda é maior que a maior alcançada pelo ALBERTA. Vale também ressaltar que são mais frequentes os aplicativos pioneiros em suas áreas, como Uber, iFood e Netflix, além dos aplicativos de redes sociais. Outro ponto de destaque é o Lift alcançado pelo ANCESTOR, uma vez que todos os aplicativos obtêm melhora, ou se mantêm, no alcance de usuários em relação a amostras aleatórias. Considerando também o Lift, vemos que dentre os aplicativos que aparecem nesta tabela, alguns foram instalados por poucos usuários, levando a uma taxa de levantamento bem alta nestes casos, chegando a ser quase 31 vezes melhor que as amostras aleatórias.

Tabela 3.6: Tabela das estatísticas dos 20 aplicativos que obtiveram melhor precisão para o modelo ANCESTOR.

Aplicativo	Total Instalações	Precisão ALBERTA	Precisão ANCESTOR	Revocação ALBERTA	Revocação ANCESTOR	Lift ALBERTA	Lift ANCESTOR
Uber	961	0,23%	14,45%	0,10%	9,05%	-0,98	0,11
Messenger	937	2,12%	12,92%	1,07%	8,64%	-0,83	0,02
Instagram	896	0,69%	12,64%	0,33%	8,59%	-0,94	0,05
Whatsapp	894	0,22%	12,02%	0,11%	10,07%	-0,98	0,01
Facebook	855	0,63%	11,50%	0,35%	9,12%	-0,95	0,00
iFood	889	0,69%	11,24%	0,22%	3,15%	-0,94	-0,06
OLX	651	1,65%	11,16%	0,92%	7,53%	-0,81	0,27
Guiabolso	70	1,89%	11,11%	1,43%	5,71%	1,02	10,83
MercadoLivre	692	1,33%	10,79%	0,58%	5,92%	-0,86	0,16
Netflix	780	0,62%	10,33%	0,26%	5,26%	-0,94	-0,02
99 Taxis	767	0,00%	10,10%	0,00%	3,91%	-1,00	-0,03
Steam	24	0,00%	10,00%	0,00%	4,17%	-1,00	30,09
Booking	597	1,12%	9,28%	0,67%	3,69%	-0,86	0,15
LinkedIn	523	0,84%	8,96%	0,38%	3,63%	-0,88	0,27
Spotify	686	0,62%	8,95%	0,29%	4,23%	-0,93	-0,03
LATAM	479	1,58%	8,85%	1,04%	3,55%	-0,76	0,36
Prime Video	96	1,75%	8,70%	1,04%	2,08%	0,39	5,70
GOL	776	1,55%	7,97%	0,64%	2,84%	-0,85	-0,24
Pinterest	359	0,71%	7,79%	0,56%	6,69%	-0,85	0,61
Twitch	18	0,00%	7,69%	0,00%	5,56%	-1,00	30,73

Considerando agora o Lift, a Tabela 3.7 mostra os 20 aplicativos que se saíram melhor de acordo com o modelo ANCESTOR. Nela vemos que todos os aplicativos possuíram poucas instalações, se comparados aos outros aplicativos, apesar de três dos quatro aplicativos com maior taxa de levantamento também fazerem parte dos melhores aplicativos em termos de precisão para o ANCESTOR. Além disso, apesar de muitas das precisões alcançadas serem relativamente baixas, elas superam as obtidas pelo ALBERTA e também por amostras aleatórias, indicando que mesmo assim foram

alcançados mais usuários. Esse comportamento pode ser explicado pelo fato de que, entre aplicativos que possuem pouco engajamento, é mais difícil acertar uma pequena parcela dentre toda a população, sendo necessário conhecer mais do usuário para fazer uma melhor recomendação.

Tabela 3.7: Tabela das estatísticas dos 20 aplicativos que obtiveram melhor lift para o modelo ANCESTOR.

Aplicativo	Total Instalações	Precisão ALBERTA	Precisão ANCESTOR	Revocação ALBERTA	Revocação ANCESTOR	Lift ALBERTA	Lift ANCESTOR
Twitch	18	0,00%	7,69%	0,00%	5,56%	-1,00	30,73
Steam	24	0,00%	10,00%	0,00%	4,17%	-1,00	30,09
Mary Kay	3	0,00%	0,49%	0,00%	66,67%	-1,00	12,16
Guiabolso	70	1,89%	11,11%	1,43%	5,71%	1,02	10,83
V.O. Hinode	7	0,00%	0,81%	0,00%	14,29%	-1,00	7,32
Twitter Lite	7	0,00%	0,70%	0,00%	14,29%	-1,00	6,87
Montreal	7	0,00%	0,66%	0,00%	14,29%	-1,00	6,18
Prime Video	96	1,75%	8,70%	1,04%	2,08%	0,39	5,70
Bolsa Família	7	0,00%	0,47%	0,00%	28,57%	-1,00	4,32
Wisecash	7	0,00%	0,41%	0,00%	28,57%	-1,00	3,80
Discord	38	0,00%	2,27%	0,00%	2,63%	-1,00	3,43
Itaú	32	0,00%	1,82%	0,00%	3,12%	-1,00	3,28
Amazon	6	0,00%	0,32%	0,00%	16,67%	-1,00	3,11
Zattini	68	1,57%	3,88%	2,94%	5,88%	0,74	3,10
Natura	17	0,00%	0,97%	0,00%	23,53%	-1,00	3,08
Conversor de Moedas	20	0,00%	1,02%	0,00%	5,00%	-1,00	2,98
Mobills	26	0,00%	1,35%	0,00%	3,85%	-1,00	2,88
Buscapé	93	0,00%	4,55%	0,00%	3,23%	-1,00	2,63
Dieta e Saúde	12	0,00%	0,52%	0,00%	8,33%	-1,00	2,32
MyFitnessPal	37	0,00%	1,54%	0,00%	2,70%	-1,00	2,14

3.5 Conclusões

Este capítulo apresentou uma solução para recomendação de aplicativos a usuários com base nos seus aplicativos instalados e informações demográficas. Os resultados apresentados mostram que a solução proposta supera a solução de base em vários aspectos, atingindo bons resultados em termos de precisão, revocação e *lift*. Com isso, a hipótese de que dados demográficos são importantes para ajudar na recomendação de instalação de aplicativos é reforçada. Além disso, o fato de a solução não necessitar de dados históricos, tanto dos aplicativos quanto dos usuários, faz com que a construção do modelo seja mais simples.

Portanto, no próximo capítulo será feita uma análise mais avançada, com um número maior de usuários e o uso de três técnicas diferentes, visando alcançar resultados mais concretos.

Capítulo 4

Análise Avançada

Uma vez que a análise preliminar, apresentada no Capítulo 3, mostrou resultados promissores, foi realizada uma análise mais elaborada acerca da utilização de dados demográficos para a recomendação de aplicativos. Além disso, também foi avaliado o impacto da utilização de dados relativos à faixa de preço do dispositivo móvel utilizado pelo usuário. Assim, neste capítulo será mostrada a metodologia e os resultados obtidos com tal análise.

4.1 Os dados

Para que o impacto da incorporação de dados demográficos no perfil do usuário fosse notado, foi utilizada uma base de dados reais mais completa, porém semelhante à utilizada no Capítulo 3. Tal base foi coletada no período de 01 de Novembro de 2019 a 31 de Janeiro de 2020 e contou inicialmente com 18.560 aplicativos, distribuídos em conjuntos de instalação de 87.903 usuários. Da mesma forma que os modelos elaborados durante a Análise Preliminar, eliminamos os usuários que possuíam apenas 1 dia de registro, resultando em 16.159 aplicativos, divididos em registros de 47.305 usuários.

Entretanto, com a hipótese de que possíveis melhorias poderiam ser alcançadas com a adição de um histórico de instalações do usuário, foi necessário estabelecer uma possível ordem em que os aplicativos presentes nas coletas foram instalados. Porém, como somente foi realizada uma coleta por dia, não é possível saber ao certo a ordem em que os *apps* instalados em um mesmo dia ocorreram. Neste contexto, a cada novo dia de coleta, são identificados os aplicativos que foram instalados ou desinstalados por um usuário. Caso ocorra a reinstalação de um *app*, este será adicionado normalmente à ordem de instalação, mesmo que já esteja nela. Além disso, como não é possível identificar a sequência de instalação dos aplicativos presentes na primeira captura, estes serão mantidos separadamente à sequência em si. Assim, um exemplo de sequência de instalação pode ser visto na Tabela 4.1. Para poupar constante reprocessamento dos dados, também armazenamos a quantidade de dias em que houveram modificações nos aplicativos do usuário (instalações e desinstalações).

Como podemos ver, os aplicativos existentes na data da primeira coleta são armazenados na coluna *primeiros_apps*, enquanto os instalados posteriormente são inseri-

Tabela 4.1: Exemplo dos dados obtidos após as etapas de limpeza e pré-processamento.

cod_usuario	primeiros_apps	sequencia_instalacao	dias_com_instalacao
0	[Instagram, Placa_Fipe, LinkedIn]	[Video_Editor, i=8, Jogo_Bolhas, i=2, Wish, i=1, Aliexpress]	4
1	[Sicoob, 99Taxi, Imovelweb]	[Uber, Netshoes, i=2, Uber]	2
2	[Gmail, Rentcars, Netshoes, Imovelweb]	[99Taxi, Placa_Fipe, Bradesco]	1

dos na lista mantida na coluna *sequencia_instalacao*. Tal coluna também tem armazenados separadores na forma $i = X$, onde X representa o número de dias entre as instalações. Por exemplo, no momento da primeira coleta, o usuário 0 possuía três aplicativos instalados (*Instagram*, *Placa_Fipe*, *LinkedIn*). Como tais aplicativos já estavam instalados no primeiro dia, não há como saber a ordem em que foram instalados. Já no próximo dia em que houve modificação no conjunto de aplicativos do usuário, este instalou o *app Video_Editor*. Este foi seguido do aplicativo *Jogo_Bolhas*, com um intervalo de 8 dias entre as instalações ($i = 8$). Depois de dois dias, o usuário instalou também o aplicativo *Wish*, seguido do *app Aliexpress* no dia seguinte. No exemplo acima, é possível estabelecer a ordem em que cada aplicativo foi instalado, partindo da segunda data de instalação. Porém, também é possível que mais de um *app* seja instalado no mesmo dia, como podemos ver no exemplo do usuário 1. Como ambos os aplicativos, *Uber* e *Netshoes*, foram instalados na mesma data, não conseguimos estabelecer a ordem de instalação entre eles. Porém, houve a reinstalação do *app Uber* após o intervalo de 2 dias, uma vez que este volta a aparecer na lista de instalações. Casos como o do usuário de código *2* representam um impasse pois, apesar de haverem instalações, o fato de todas serem referentes ao mesmo dia, faz com que não seja possível inferir a ordem em que ocorreram.

Uma vez que deseja-se utilizar históricos de instalações de usuários, não foram considerados aqueles que possuíam menos de 7 aplicativos instalados após o primeiro dia, sendo que estes deveriam estar distribuídos entre instalações de 4 ou mais datas de coleta. Estes valores foram escolhidos por representarem a faixa de corte do primeiro quartil dentre os dados coletados. Ou seja, 75% dos usuários possuíam 7 ou mais aplicativos na sequência de instalações e, destes, outros 75% dos usuários haviam obtidos tais *apps* em um intervalo maior ou igual a 4 dias. Tendo realizado tais filtros, obtivemos uma base com 13.977 *apps*, dentre os 18.054 usuários.

Também foram avaliados casos onde ocorreram possíveis erros de coleta, como aquelas que registravam um número de *apps* muito discrepante em relação a coletas

de outras datas no mesmo período. Por isso, utilizou-se o *Z-Score* para filtrar tais ocorrências, fazendo com que o número de aplicativos distintos fosse reduzido a 14.667, distribuídos entre coletas de 13.343 usuários. Além disso, como os modelos também seriam avaliados em termos de categorias de aplicativos, e alguns dos *apps* identificados não possuíam categoria definida, estes também foram removidos. Logo, a base de dados final contou com um conjunto de 14.660 usuários e 13.329 *apps*, que podem ser agrupados em 48 categorias.

Diferentemente do que foi feito no Capítulo 3, foram utilizadas as categorias da *Play Store*. Isso foi feito pelo fato de que categorizar manualmente todos os aplicativos seria inviável, assim como desconsiderar boa parte de tais *apps* para utilizar as categorias levantadas anteriormente.

4.1.1 Dados Demográficos e de Dispositivos

Para escolha de quais dados demográficos seriam utilizados, cada indicador foi analisado semanticamente, de forma a levantar aqueles que poderiam ter maior impacto na instalação de aplicativos. Sendo assim, foram utilizados apenas dados referentes à renda e tamanho da população da cidade onde o usuário reside. Além disso, também foram utilizadas informações acerca do tipo de dispositivo móvel que o usuário possui. As informações demográficas aqui utilizadas foram retiradas das bases de dados do IBGE mencionadas anteriormente na Seção 3.1.1 (Censo de 2010 e IBGE Cidades).

Renda Média

Diferentemente da abordagem adotada na análise preliminar, utilizamos os dados brutos acerca da renda média por morador dos setores censitários para categorizar a renda dos usuários de acordo com seu local de residência. Vale ressaltar que, como os dados do IBGE mais atualizados são referentes ao censo de 2010, utilizamos o valor do salário mínimo aplicado na época. As categorias foram definidas em **Baixa**, **Intermediária Baixa**, **Média**, **Intermediária Alta** e **Alta**, conforme valores abaixo:

- **Baixa:** Menos de meio salário mínimo (Renda < R$255);
- **Intermediária Baixa:** De meio salário mínimo a um salário mínimo (R$255 $\leq$ Renda < R$510);
- **Média:** De um até dois salários mínimos (R$510 $\leq$ Renda < R$1.020);
- **Intermediária Alta:** De dois a quatro salários mínimos (R$1.020 $\leq$ Renda < R$2.040);
- **Alta:** Mais de quatro salários mínimos (Renda $\geq$ R$2.040).

O gráfico com a distribuição de usuários por categoria de renda pode ser visto na Figura 4.1.

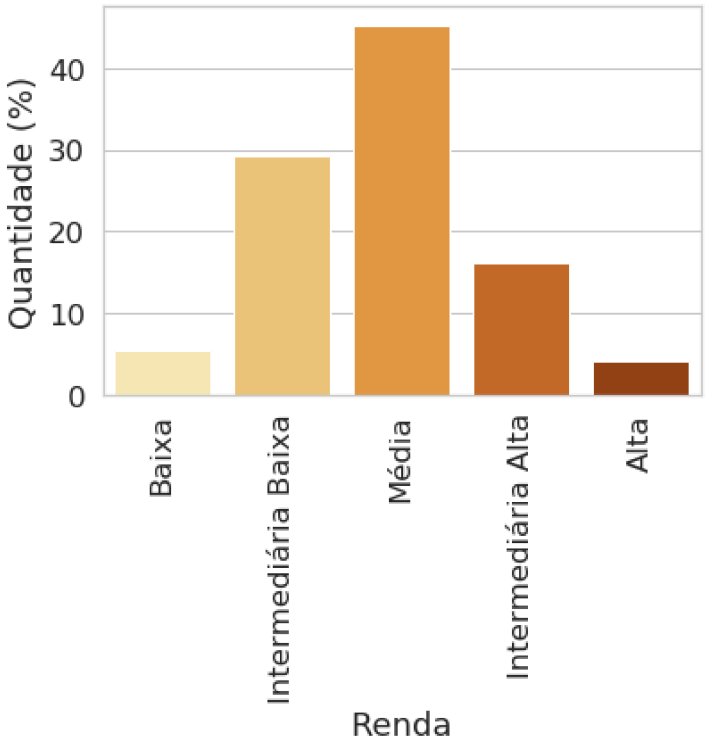


Figura 4.1: Distribuição dos usuários por renda.

Tamanho da População

Assim como foi feito com os dados de renda, também foram utilizadas categorias para identificar o tamanho da população da cidade de residência do usuário. A forma de divisão utilizada se baseou nos conceitos de cidades pequenas, médias e grandes, conforme Ipea (2020), e se deu da seguinte maneira:

- **Cidades com População Pequena:** São aquelas que seguem Tamanho Populacional < 100.000;
- **Cidades com População Média:** Equivale às cidades cujo número de habitantes segue $100.000 \leq$ Tamanho Populacional < 500.000;
- **Cidades com População Grande:** São as cidades com Tamanho Populacional ≥ 500.000.

Na Figura 4.2 é possível ver a porcentagem de usuários presentes em cada uma das categorias apresentadas acima. Nela, vemos que quase metade (46,44%) dos usuários residem em cidades de tamanho médio, possuindo aproximadamente 128% mais usuários que cidades consideradas grandes, por exemplo.

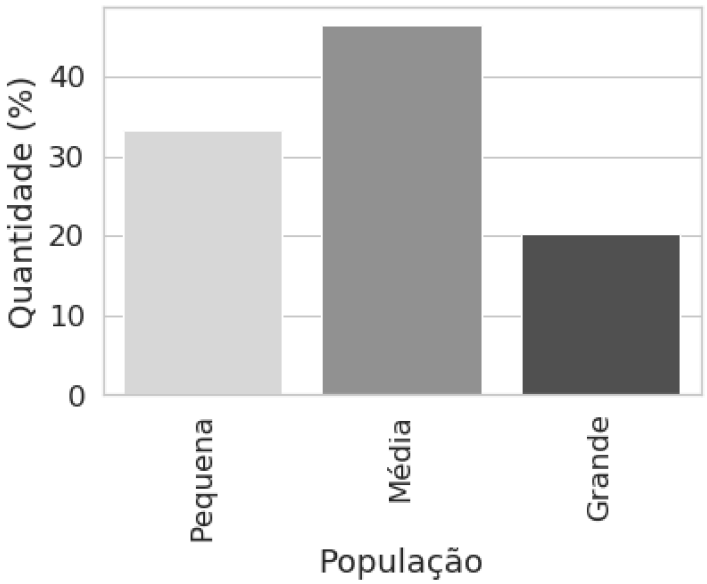


Figura 4.2: Distribuição dos usuários por tamanho da cidade de residência.

Preço do Dispositivo

Ao contrário das informações agregadas anteriormente, o preço do dispositivo utilizado pelo usuário não foi obtido através de uma base de dados do IBGE, e sim através do trabalho de Maia et al. (2020). A indicação de qual o dispositivo utilizado foi coletada juntamente ao conjunto de aplicativos que o usuário possuía instalado na referida data.

Uma vez obtido, o preço do dispositivo do usuário também foi classificado, de acordo com a divisão obtida em Medeiros (2019), em:

- **Entrada:** Dispositivos com Preço $< R\$700$;
- **Intermediário:** Aparelhos na faixa de $R\$700 \leq$ Preço $< R\$1.000$;
- ***Mid-high*:** Dispositivos na faixa de $R\$1.000 \leq$ Preço $< R\$2.000$;
- ***High-end*:** $R\$2.000 \leq$ Preço $< R\$3.000$;
- ***Premium*:** Aparelhos cujo Preço $\geq R\$3.000$.

Com o uso de tais informações, esperamos poder identificar padrões mais específicos de instalação relacionados à renda e consumo dos usuários. Na Figura 4.3 é possível ver a distribuição de usuários de acordo com as categorias de preço estabelecidas acima. Nela vemos que o número de usuários com dispositivos considerados Intermediários é consideravelmente superior às outras.

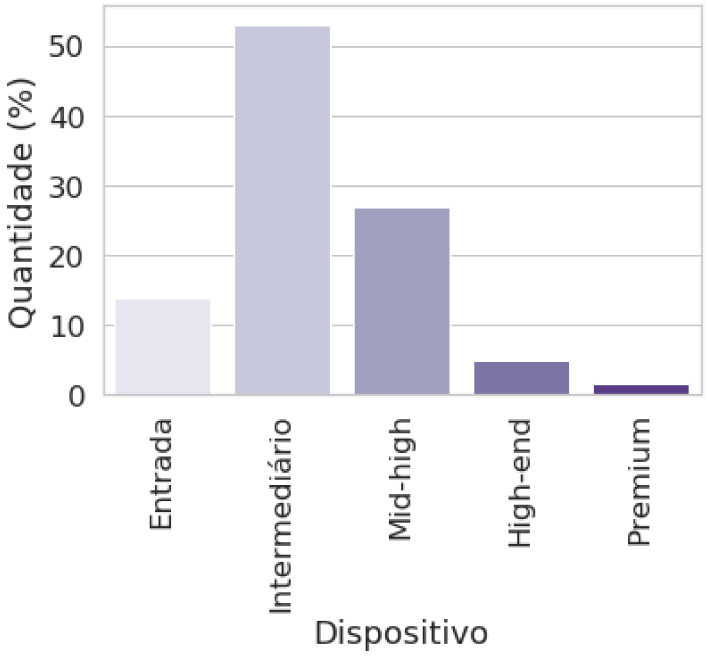


Figura 4.3: Distribuição dos usuários por preço do dispositivo.

4.2 Métricas

A utilização de diferentes abordagens e novos dados em larga escala fez com que fosse necessário adaptar as métricas anteriormente adotadas. Neste contexto, a forma como os conjuntos de treino e teste são obtidos também precisou ser revista, uma vez que estabelecer alguns dias como período de teste e outros para treino poderia reduzir significantemente o número de usuários. Assim, foi estabelecido que seriam utilizados os últimos s aplicativos instalados como conjunto de teste, com $s \in S = \{1, 3, 5\}$, mantendo os demais aplicativos para treino, assim como em Cheng et al. (2018). Além disso, para melhor avaliar o comportamento das soluções no que tange à percepção do perfil do usuário, realizamos recomendações de tamanho $N = \{5, 10\}$.

Com a modificação dos conjuntos de teste, treino e do formato utilizado, as especificações das métricas utilizadas também foram modificadas. Assim, sendo U o conjunto de todos os usuários, admitimos A_u^* como o conjunto de todos os aplicativos instalados pelo usuário u, sendo A_u^s o conjunto de todos os s últimos aplicativos instalados por tal usuário, e R_u^n como o conjunto de n *apps* recomendados para o mesmo usuário u. Levando isto em consideração, obteremos valores de precisão e revocação em função das variáveis s e n. Sendo assim, temos:

- **Precisão:** para que seja possível obter o valor de 100% caso todos os aplicativos recomendados sejam instalados, a precisão deverá ser calculada apenas para valores de $s = n$. Vale ressaltar também que somente para esta métrica, foi utilizado $N = \{1, 3, 5\}$, para que ficasse coerente com o conjunto S. Portanto,

a precisão definirá a porcentagem do total das recomendações assertivas, analisando recomendações de tamanho $n = s$, em relação ao total de aplicativos instalados.

$$Precisao_{s,n} = \frac{1}{|U|} \sum_{u \in U} \frac{|A_u^s \cap R_u^n|}{n} \tag{4.1}$$

onde $s \in S$ e $n = s$.

- **Revocação:** definida como a porcentagem, dentre os *s apps* instalados pelo usuário, que estavam no conjunto recomendado.

$$Revocacao_{s,n} = \frac{1}{|U|} \sum_{u \in U} \frac{|A_u^s \cap R_u^n|}{s} \tag{4.2}$$

onde $s \in S$ e $n \in N$.

Como as métricas de precisão e revocação precisaram ser calculadas para um número de recomendações distintos, não foi possível calcular a métrica de *F-Score*.

Além disso, já que para cálculo da precisão é necessário ter $n = s$, podemos considerar que a ideia de tal métrica é prever os s próximos *apps* que serão instalados pelo usuário. Por outro lado, por ser possível recomendar um número maior de aplicativos do que os presentes no conjunto de teste e ainda obter 100% de revocação, pode-se inferir que tal métrica melhor representa o desempenho das soluções, se compararmos com o que se espera de um sistema de recomendação na vida real.

Além das métricas levantadas acima, a análise realizada neste capítulo também levou em consideração o desempenho dos modelos em relação às categorias de aplicativos. Sendo assim, calculamos as métricas de Precisão e Revocação para recomendações tanto dos aplicativos quanto para suas categorias. Vale ressaltar que o cálculo das métricas referentes às categorias foi feito com base nos aplicativos recomendados para cada usuário.

Para que tais categorias pudessem ser obtidas, foram levantados os *apps* recomendados para cada usuário e estes foram substituídos por suas respectivas categorias. Dessa forma, uma categoria tem a chance de ser recomendada mais de uma vez para o mesmo usuário. A Figura 4.4 mostra um exemplo acerca da forma com que foram obtidas as categorias para cada usuário.

Como a forma com que as categorias foram obtidas permite que haja a repetição destas, foi necessário alterar a definição das métricas anteriores. Assim sendo, seja $R_{u,cat}^n$ o multiconjunto das n categorias recomendadas ao usuário u, e $A_{u,cat}^s$ o multiconjunto com todas as categorias dos s últimos aplicativos instalados pelo referido usuário u. Então, definimos:

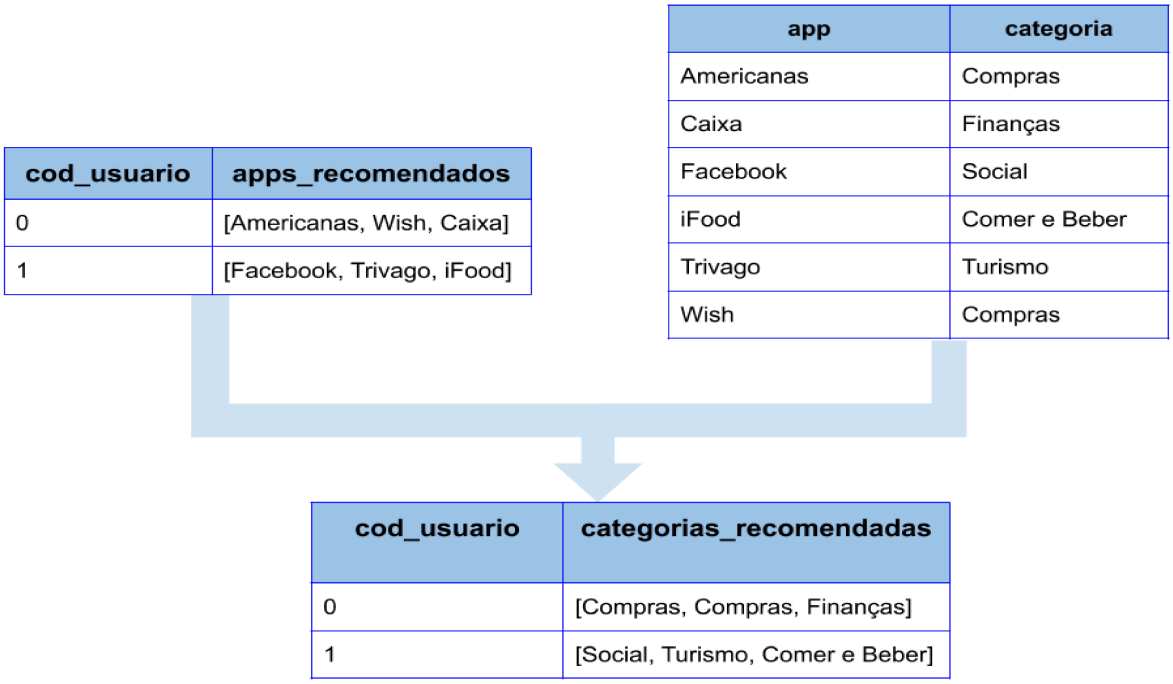


Figura 4.4: Exemplo de obtenção das categorias recomendadas.

$$R^n_{u,cat} = \left\{ cat_1^{m_r(cat_1)}, \ldots, cat_n^{m_r(cat_n)} \right\} \tag{4.3}$$

Da mesma forma:

$$A^s_{u,cat} = \left\{ cat_1^{m_a(cat_1)}, \ldots, cat_s^{m_a(cat_s)} \right\} \tag{4.4}$$

Sabendo que $m_r(x)$ e $m_a(x)$ representam as funções de multiplicidade dos multiconjuntos $R^n_{u,cat}$ e $A^s_{u,cat}$, respectivamente, definidas como:

$$m_r(x) = \sum_{c \in R^n_{u,cat}} \mathbb{1}(c = x) \tag{4.5}$$

e

$$m_a(x) = \sum_{c \in A^s_{u,cat}} \mathbb{1}(c = x) \tag{4.6}$$

Ou seja, a respectiva função de multiplicidade indica a quantidade de vezes que uma categoria aparece no multiconjunto analisado.

Além disso, a cardinalidade da interseção entre os dois multiconjuntos $A^s_{u,cat}$ e $R^n_{u,cat}$ (também chamada de *divisor comum mínimo*) pode ser dada por:

$$T_u = \sum_{c \in A^s_{u,cat}} \min(m_a(c), m_r(c)) \tag{4.7}$$

Com base nas equações acima, a precisão e revocação de categorias podem ser

definidas como:

- **Precisão de Categorias:** Definida como a porcentagem do total das categorias dos *apps* recomendados, quantas foram instaladas pelo usuário. Vale ressaltar que, assim como a métrica de Precisão para aplicativos, foi utilizado $N = \{1, 3, 5\}$ para o cálculo desta métrica.

$$Precisao_{cat\,s,n} = \frac{1}{|U|} \sum_{u \in U} \frac{T_u}{n} \tag{4.8}$$

  onde $s = n$ e $s \in S$.

- **Revocação de Categorias:** definida como a porcentagem de categorias que o usuário u instalou e que algum de seus *apps* foram recomendados para tal usuário.

$$Revocacao_{cat\,s,n} = \frac{1}{|U|} \sum_{u \in U} \frac{T_u}{s} \tag{4.9}$$

  onde $s \in S$ e $n \in N$.

4.3 Modelos

Apesar de a análise preliminar ter mostrado que a adição de dados demográficos torna a recomendação de aplicativos móveis mais assertiva, os resultados obtidos ainda possuíram margem para melhoria. Por este motivo, foram adotadas estratégias que visaram uma melhor utilização dos dados históricos do usuário. Para isso, os dados foram aplicados a três modelos diferentes: *Latent Dirichlet Allocation* (LDA), Filtro Colaborativo (FC) e às Matrizes de Transição de Markov (MTM).

4.3.1 *Latent Dirichlet Allocation* (LDA)

Latent Dirichlet Allocation (LDA) é um modelo probabilístico inicialmente desenvolvido para processamento de textos e bastante utilizado em atividades na área de Recuperação de Informação. O funcionamento do modelo trabalha através do pressuposto de que cada documento possui um texto e que, tais textos podem ser descritos como um conjunto de tópicos. Por sua vez, cada tópico possui termos que são mais frequentemente associados a ele (Blei et al., 2003). Por exemplo, em um tópico referente a arte seria muito provável encontrar termos como pintura, desenho, lápis, luz, sombra, entre outros.

Para que tais tópicos sejam encontrados, pressupõe-se que os textos sejam sujeitos ao princípio da permutabilidade, ou seja, a ordem em que termos aparecem dentro

do texto não tem impacto. Da mesma forma, subtende-se que a ordem dos textos de um usuário também não alterará o resultado final. A adoção de tal princípio fará com que seja possível identificar padrões de presença, relacionando termos que na maioria das vezes aparecem juntos, mesmo que sua relação não seja clara a princípio.

Entretanto, para que seja possível identificar os tópicos de forma mais adequada, é importante compreender a natureza dos documentos analisados. Tal natureza refere-se à abrangência de temas presentes em cada documento, e é definida pelo hiperparâmetro α. Caso o *corpus* dos documentos analisados indique que os termos presentes em cada um abrangerão poucos temas, ou seja, que cada documento irá possuir poucos tópicos relevantes que condensam a maioria dos termos, deverá ser utilizado um valor de α menor. Por outro lado, se cada documento possui termos relevantes em vários temas, e é importantes manter e compreender tal diversidade, a escolha de uma valor de α maior é mais adequada. Além disso, α pode assumir valores escalares ou representado por uma matriz de valores, quando espera-se que a distribuição dos termos seja simétrica ou não, respectivamente.

Assim sendo, a estrutura final de um modelo LDA é composta por um conjunto de k tópicos, sendo k um hiperparâmetro definido pelo usuário. Tais tópicos possuem uma lista com todos os termos presentes em todos os documentos, ordenados por sua relevância (distribuição) dentro do contexto abordado em cada tópico. Vale ressaltar que mesmo que um termo não possua relação alguma com um tópico em questão, esse ainda estará presente na lista de termos de tal tópico, porém com relevância igual a zero. É também possível avaliar a distribuição dos tópicos para cada documento, tornando possível elencar aqueles mais relevantes para o documento escolhido.

Uma vez criado, é necessário verificar se o modelo é adequado ao que foi proposto. Uma forma de realizar tal avaliação é através da análise da coerência de tópicos (Röder et al., 2015), que diz repeito à avaliação de modelos não-supervisionados de processamento de texto. A coerência de tópicos busca analisar os termos com maior relevância (distribuição) em cada tópico, e verificar se as semânticas destes podem ser relacionadas.

Solução Base

Para a criação do modelo LDA base, foi utilizado como referência o trabalho apresentado por Cheng et al. (2018). Assim, foi considerado que os documentos e termos seriam representados pelos usuários e os nomes de seus aplicativos instalados, respectivamente. Além disso, considerando o princípio da permutabilidade, seria possível a utilização também dos aplicativos presentes no primeiro dia de coleta de cada usuário (coluna *primeiros_apps* da Tabela 4.1). Entretanto, tais dados não foram utilizados, já que como não temos informação da ordem em que foram instalados, estes poderiam

representar um perfil de usuário que não mais condiz com a atualidade. Isso porque o usuário poderia ter mantido aplicativos de fábrica, o que geraria ruídos, ou ainda instalado um *app* há muito tempo atrás, e o mantido mesmo sem o utilizar.

Assim sendo, considerando que α será representado por um escalar, um modelo LDA é criado seguindo os seguintes passos, descritos por Cheng et al. (2018):

1. Estabeleça k tópicos $\varphi_1, ..., \varphi_K$ derivados da distribuição *Dirichlet* $Dir(\beta)$.
2. Para cada usuário u, estabeleça sua distribuição de tópicos $\theta_u \sim Dir(\alpha)$.
3. Para cada aplicativo instalado por u (A_u^*):
   (a) Escolha um tópico $\varphi_j \sim multinomial(\theta_u)$.
   (b) Escolha $app_i \sim multinomial(\varphi_j)$.

Sendo que o hiperparâmetro β foi definido como $\frac{1}{K}$. Já o número de tópicos e o valor do hiperparâmetro α foram escolhidos com base em testes realizados através da sequência de passos:

1. Estabeleça A como o conjunto de valores passíveis de serem boas escolhas para α.
2. Defina Ψ como o conjunto de valores considerados aceitáveis para representar o número de tópicos.
3. Para cada combinação de valores de $a \in A$ e $\psi \in \Psi$, repita η vezes os passos:
   (a) Crie o modelo LDA com $\alpha = a$ e $K = \psi$.
   (b) Calcule a coerência de tópicos do modelo criado.
4. Escolha um dentre todos os modelos elaborados.

A necessidade de se criar η modelos com os mesmos hiperparâmetros, se dá pelo fato de que a criação do modelo em si é probabilística. Isto faz com que modelos desenvolvidos a partir do mesmo *corpus* e com mesmos hiperparâmetros resultem em modelos diferentes. Sendo assim, foi adotado $\eta = 5$. Na Figura 4.5, são mostrados os valores de coerência dos vinte melhores modelos, e seus respectivos hiperparâmetros α e K. Vale ressaltar que η não foi mostrado por ser constante em relação ao número de tópicos ($\frac{1}{K}$).

No que diz respeito à escolha do modelo, foi considerado que seria utilizado aquele que obtivesse melhores resultados em termos de coerência. Sendo assim, utilizamos o modelo de 7 tópicos e $\alpha = 31$, gerado na primeira iteração.

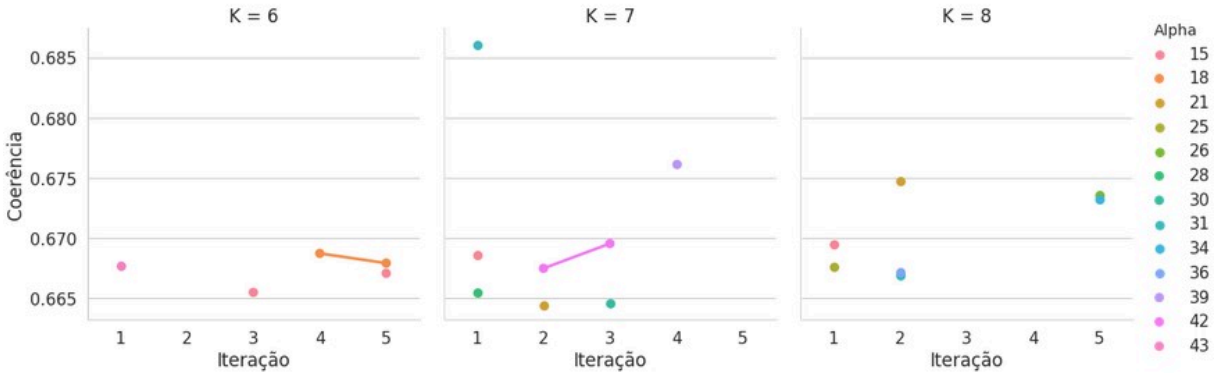


Figura 4.5: Vinte melhores modelos em termos de coerência de tópicos.

Uma vez que o modelo escolhido já havia aprendido os padrões, foi possível recomendar propriamente os aplicativos aos usuários. Para isso, foram utilizados os passos descritos a seguir, como discutido por Cheng et al. (2018):

1. Para cada tópico inferido, ordene de forma decrescente seus aplicativos, de acordo com a distribuição apresentada naquele tópico.

2. Para cada usuário u, obtenha sua distribuição θ_u através do modelo treinado.

3. Repita os passos abaixo n vezes, para recomendar n aplicativos:

   (a) Escolha um tópico x através da distribuição de tópicos do usuário.

   (b) Recomende o melhor *app* do tópico ao usuário, desde que este ainda não o possua ou que tal *app* ainda não tenha sido recomendado.

Solução Proposta

Para construir uma solução que levasse em consideração os dados demográficos levantados, não seria possível utilizar do mesmo modelo gerado anteriormente. Isso porque tal modelo não foi treinado de forma a se relacionar com a inserção dos dados enriquecidos. Sendo assim, foram gerados novos modelos de acordo com os passos descritos anteriormente, sob as mesmas condições.

Além disso, visando analisar e avaliar o impacto de cada tipo de dado demográfico separadamente, foram criadas diferentes soluções. Cada uma destas utilizou um subconjunto dos dados demográficos considerados, tornando possível que também fosse avaliado seu desempenho em função das três abordagens levantadas (LDA, Filtro Colaborativo e MTM). Assim, como foram analisadas diferentes situações e uso dos dados enriquecidos, também foram gerados novos modelos para cada uma de tais situações. Estas serão chamadas a partir de agora de *Soluções*, e seguem como discutido abaixo:

Tabela 4.2: Hiperparâmetros dos melhores modelos obtidos para cada solução.

Solução	**Hiperparâmetros**	
	Número de Tópicos	**Alpha**
População	8	28
Renda	8	40
Dispositivo	8	30
Demográfica	7	26
Completa	7	25

- ***População*:** São utilizadas somente as informações referentes ao tamanho da população da cidade de residência do usuário.
- ***Renda*:** Somente os dados acerca da renda média da região de residência do usuário serão incorporados.
- ***Dispositivo*:** Será utilizada somente a informação de faixa de preço do dispositivo do usuário.
- ***Demográfica*:** Utiliza somente as informações demográficas do usuário, ou seja, informações de renda e população.
- ***Completa*:** Utiliza todas as informações disponíveis do usuário, ou seja, aquelas referentes a renda, população e dispositivo.

Para cada solução especificada acima, iremos incrementar a lista de aplicativos do usuário com as respectivas informações a serem utilizadas. Dessa forma, cada documento segue representado como um usuário, porém os termos presentes em tais documentos agora se referem tanto aos aplicativos que o usuário possui, quanto às suas informações extras.

Além disso, como foram criados novos modelos, os hiperparâmetros também tiveram que ser revistos. Para isso, foram seguidos os mesmos passos descritos anteriormente para criação do modelo base. Sendo assim, os hiperparâmetros escolhidos para cada solução podem ser vistos na Tabela 4.2.

Finalmente, no que se refere ao processo de recomendação, somente foi necessário certificar-se de que nenhuma informação adicional seria recomendada ao usuário. Assim, os seguintes passos são aplicados:

1. Para cada tópico inferido, faça:
   (a) Exclua os itens que representam as informações demográficas e de dispositivos adicionadas.
   (b) Ordene de forma decrescente seus aplicativos, de acordo com a distribuição apresentada naquele tópico.

2. Para cada usuário u, obtenha sua distribuição θ_u através do modelo treinado.

3. Repita os passos abaixo n vezes, para gerar n recomendações:

   (a) Escolha um tópico x através da distribuição de tópicos do usuário.

   (b) Recomende o melhor *app* do tópico ao usuário, desde que este ainda não o possua ou que tal *app* ainda não tenha sido recomendado.

Resultados

Como explicitado nas seções anteriores, foram utilizadas as métricas de precisão e revocação para avaliar o desempenho das soluções propostas. Quanto aos dados de teste, foram utilizados os S últimos aplicativos instalados pelo usuário, sendo $S = \{1, 3, 5\}$. Vale ressaltar que as métricas utilizadas foram aplicadas a ambos os contextos de aplicativos e categorias, como discutido na Seção 4.2.

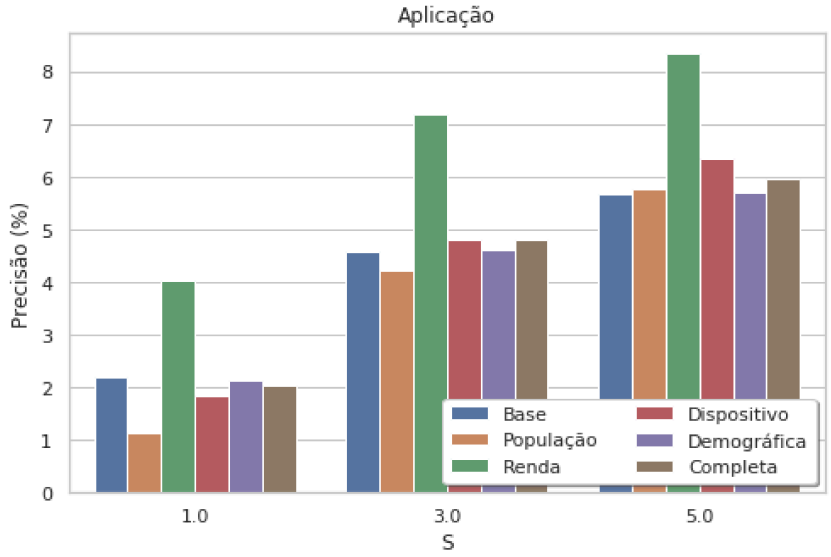


Figura 4.6: Resultados obtidos com os modelos LDA em termos de precisão, no contexto de aplicativos.

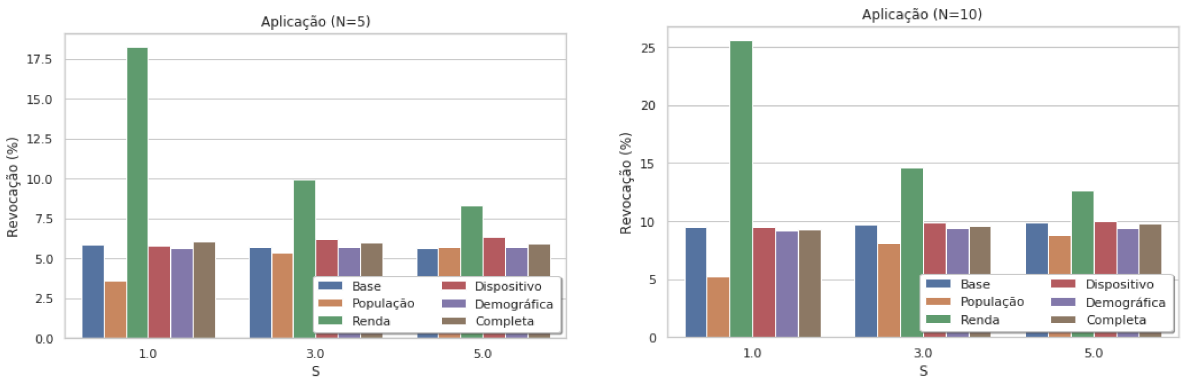
Como podemos ver na Figura 4.6, em geral os resultados dos modelos adaptados se saíram melhor que a solução base quando foram analisados os 3 e 5 últimos aplicativos instalados ($s = 3$ e $s = 5$). De fato, é percebida uma melhora nos resultados quando se aumenta o valor de s. Tal comportamento condiz com o esperado, uma vez que se são recomendados mais aplicativos, é mais provável acertar os que foram instalados pelo usuário. Já a inclusão do tamanho da população da cidade de residência do usuário não obteve bons resultados, superando a solução *Base* apenas quando recomendados 5 aplicativos. Isto pode indicar que as classes de usuários obtidas, além de não serem relevantes para este fim, ainda podem ter confundido o modelo.

Tal situação pode ter sido ocasionada pela disparidade do percentual de usuários residentes em cada tipo de cidade (i.e., cidades pequenas, médias e grandes). Assim,

com uma grande número de usuários agrupados em uma mesma categoria, é possível haver também uma grande diversidade de interesses, tornando mais difícil encontrar padrões. Além disso, a informação do tamanho da cidade é por si só generalista, uma vez que naturalmente agrupa um grande número de usuários com interesses e estilo de vida distintos. Por outro lado, informações como o dispositivo do usuário e a renda média do setor censitário em que o mesmo reside, dizem mais a respeito do contexto socioeconômico em que o indivíduo está inserido, sendo assim mais específicas.

Já quanto à informação da categoria do dispositivo, foi percebida uma melhora, mesmo que discreta, para recomendações de 3 e 5 aplicativos (4,51% e 11,71%, respectivamente). Tal melhora pode indicar uma leve polarização de determinados tipos de aplicativos no que compete às classes de dispositivos.

Por outro lado, vemos que a adição somente da informação de renda tem impacto positivo, o que pode indicar tanto uma maior relevância com relação ao tipo de aplicativo consumido por pessoas de diferentes classes sociais, quanto uma determinação mais assertiva das características de cada usuário.



(a) Recomendação de 5 aplicativos.

(b) Recomendação de 10 aplicativos.

Figura 4.7: Resultados de revocação obtidos no contexto de aplicativos, com os modelos LDA.

Já quanto aos resultados referentes a revocação, estes mostram uma melhora discreta para quase todas as soluções, como mostrado na Figura 4.7. As exceções são representadas mais uma vez pela adição da informação de tamanho populacional, onde houve melhoria discreta apenas em $s = 5$ e $n = 5$ (1,44%, visto na Figura 4.7a), e pela incorporação de informações referentes a renda, que mais uma vez apresentou melhorias relevantes. Assim, conhecer a renda média da região do usuário leva a melhoria de até aproximadamente 12 pontos percentuais (210,22%), se recomendados 5 aplicativos (Figura 4.7a), e de 16 pontos percentuais (167,45%), com a recomendação de 10 aplicativos (Figura 4.7b).

Já no que consta às métricas de categorias, a maior parte das soluções propostas não se mostrou eficaz, apesar de alcançarem níveis mais altos de precisão do que a

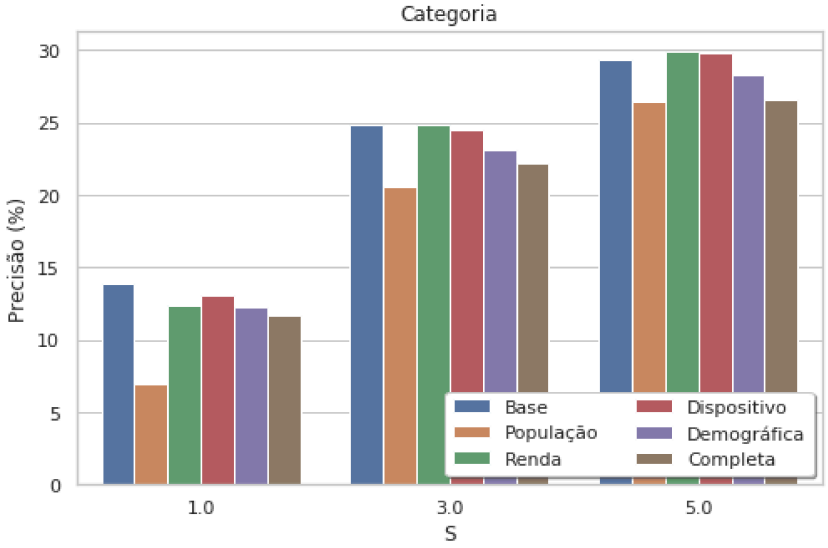
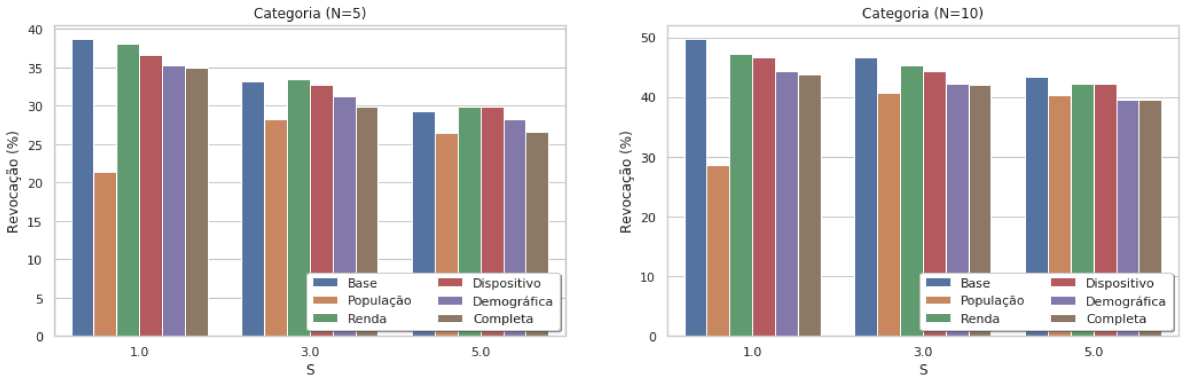


Figura 4.8: Resultados dos modelos LDA, obtidos em termos de precisão, no contexto de categorias.



(a) Recomendação de 5 aplicativos.

(b) Recomendação de 10 aplicativos.

Figura 4.9: Resultados de revocação obtidos no contexto de categorias, para os modelos LDA.

mesma métrica para aplicativos (Figura 4.8). As únicas exceções dizem respeito à adição das classes de renda, com uma melhora discreta de 1,92% em $s = 5$; além da incorporação de dispositivos, que se saiu melhor para $s = 5$ (1,77%). Esta situação pode ser explicada pelo fato de que a análise de categorias é feita com base na conversão dos aplicativos recomendados a suas respectivas categorias (Seção 4.2). Assim, pode ser que haja a recomendação de mais de um aplicativo da mesma categoria, fato que pode ser alavancado com o uso de informações mais específicas acerca do perfil do usuário, como é o caso de informações da classe de dispositivos. Sendo assim, a recomendação de mais de um *app* de categorias errôneas teria um impacto negativo maior do que a solução base.

Por fim, a análise de categorias em termos de revocação confirmou a tendência de

piora no que compete às soluções *População*, *Demográfica* e *Completa*, como visto na Figura 4.9. Entretanto, mais uma vez é possível notar que as soluções *Renda* e *Dispositivo* se mostraram eficazes ao se recomendar 5 *apps* e compará-los com os últimos 3 ou 5 aplicativos do usuário. Tal situação pode indicar que a adição de informações de contexto mais específicos pode ajudar na recomendação de aplicativos.

Vale também destacar que, para a métrica de revocação, espera-se que os resultados de cada solução se mantenham ou diminuam com o aumento da quantidade de *apps* de teste (i.e., valores de S). Entretanto, os resultados referentes à solução *População* mostram aumento entre $s = 1$ e $s = 3$, com diminuição entre $s = 3$ e $s = 5$. Para compreender tal resultado, primeiramente é preciso entender que considerando $s = 1$, um *app* predito corretamente leva a uma taxa de acerto de 100% para o usuário (i.e., ou o aplicativo recomendado foi instalado, ou não). Entretanto, quando analisamos $s = 3$, cada *app* recomendado corretamente, adiciona apenas cerca de 33% à taxa de acerto. O mesmo pode ser dito de $s = 5$, onde cada acerto conta apenas 20% para a revocação daquele usuário.

Assim, através de análises dos resultados obtidos, percebeu-se que quando verificados os 3 últimos aplicativos instalados, houve certa quantidade de acertos de apenas 1 aplicativo, e um número inferior de usuários onde acertou-se 2 *apps*. Entretanto, ao se considerar $s = 5$, houve aumento do número de acertos de 1 e 2 aplicativos, porém o número de vezes onde foi possível acertar 3, 4 ou 5 aplicativos foi considerada ínfima. Sendo assim, ao analisarmos *População*, esta obteve maior chance de se acertar apenas 1 ou 2 aplicativos, atingindo melhores resultados quando considerados apenas os 3 últimos *apps* instalados. Tal situação pode ter sido desencadeada pelo fato de o tamanho da população ser uma informação generalista.

Conclusões

Com base nos resultados obtidos, vemos que a aplicação da solução LDA se mostrou interessante em situações onde se deseja recomendar aplicativos, podendo ser adicionadas somente informações referentes à renda média da região onde o usuário habita. Se, por outro lado, se deseja recomendar uma categoria de aplicativos ao usuário, a informação de classe de dispositivo pode também ser empregada.

Quanto às outras soluções, uma possível melhoria seria a adição de pesos às informações mais generalistas, como população. Isso porque o uso de tais dados com pesos iguais ao de aplicativos, faz com que seu uso seja mínimo, e podendo até mesmo confundir o modelo.

4.3.2 Filtro Colaborativo (FC)

Como detalhado anteriormente na Seção 3.3, o Filtro Colaborativo é uma estratégia bastante utilizada em Sistemas de Recomendação em geral. Essa abordagem é dividida entre filtros colaborativos *Model-based*, focada na adoção de soluções principalmente de *machine learning*, e *Memory-based*, que é direcionada para a análise de similaridade entre usuários e/ou itens.

Apesar de possuir muitas vantagens, a abordagem *Memory-based* não é indicada em situações onde será utilizado um grande volume de dados. Assim, nesta análise será utilizada como base, a abordagem proposta por Kula (2015). Tal abordagem segue uma proposta híbrida de técnicas *Content-based* (Detalhada abaixo) e *Model-based*, a última através do uso de Fatoração Matricial.

A ideia principal por trás do uso da Fatoração Matricial em sistemas de recomendação é que supõe-se que, quando um usuário avalia um item, o porquê da escolha deste item em especial se dá através de características latentes. Tais características podem ser algo que aquele item em específico possui, mas que podem relacioná-lo a outros itens. Quando um usuário avalia uma música, por exemplo, o motivo da avaliação pode ter relação com o gênero de tal música, artista, sonoridade, entre outros. Assim, uma vez que um perfil de usuário é levantado e as características latentes de suas preferências são entendidas, é possível recomendar itens com fatores latentes semelhantes. Da mesma forma, entende-se que cada usuário é único, e portanto possui seus próprios fatores latentes. Entretanto, por se tratar de uma matriz de grande esparsidade, não é recomendado o uso de algoritmos como SVD (*Singular Value Decomposition*) para identificação de tais fatores latentes (Koren et al., 2009).

Assim, para encontrar tais fatores iremos assumir que um item i pode ser representado através de um vetor $q_i \in \mathbb{R}^f$, onde q_i representará o quanto i possui um determinado fator latente f. Do mesmo modo, o usuário u também será representado pelo vetor $p_u \in \mathbb{R}^f$, sendo que p_u determinará o quão adepto u é de um fator. Assim, a recomendação se dará através da estimativa de qual avaliação u dará a qualquer item i. Tal estimativa pode ser representada por:

$$\hat{r}_{ui} = q_i^T \cdot p_u \tag{4.10}$$

Porém, a estimativa de p_u e q_i não é simples, e deve levar em consideração a possibilidade de *overfitting*. Assim, utilizaremos um peso para calcular o erro quadrático de cada tentativa, diminuindo as chances de tal possibilidade ocorrer. Logo, tentaremos minimizar a seguinte equação:

$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{4.11}$$

Em tal equação, κ representa o conjunto de todos os pares (u, i) onde r_{ui} é conhecido, ou seja, onde o usuário u efetivamente avaliou o item i. Já λ irá representar o peso utilizado no erro quadrático, para evitar *overfitting*.

Dois algoritmos famosos para cálculo da Equação 4.11 são: Mínimos Quadrados Alternantes (do inglês *Alternating Least Squares* - ALS) e Gradiente Descendente Estocástico (do inglês *Stochastic Gradient Descent*). Como o primeiro é indicado em casos onde a matriz utilizada não é esparsa, foi utilizado o algoritmo Gradiente Descendente Estocástico. Este trabalha de forma a percorrer todas as avaliações, calculando r_{ui} e seu respectivo erro através da equação abaixo:

$$e_{ui} \stackrel{def}{=} r_{ui} - q_i^T p_u \tag{4.12}$$

Já no que consta ao método *Content-based*, outra parte da abordagem híbrida utilizada, a ideia é elencar características de cada item e, se o usuário avaliar positivamente um deles, recomendar ao usuário outros itens com aquelas mesmas características (Thorat et al., 2015). Para isso, serão seguidos os seguintes passos:

1. Inferir os atributos dos itens analisados.
2. Comparar os atributos de cada item com os atributos dos itens de preferência do usuário.
3. Recomendar ao usuário os itens identificados como parte de sua preferência.

A comparação entre itens com base em seus atributos é feita através do produto de vetores. Isso porque cada item pode ser representado por um vetor de pares (x, y), onde x representará um atributo e y se o item possui aquele atributo ou não. Dessa forma, quando um usuário se interessa em um item, ele estará manifestando interesse nos atributos de tal item. Assim, é possível criar também um vetor para o usuário, que representará porém, o nível de interesse do usuário em cada atributo x.

Solução Base

Como mencionado anteriormente, a fim de integrar as duas abordagens especificadas acima, foi utilizado o modelo *LightFM*, descrito por Kula (2015). Tal modelo utiliza a estratégia da matriz de fatoração, que utiliza fatores latentes para representar características de usuários e itens. Entretanto, é utilizada a estratégia *Content-based* ao representar cada fator latente como funções de definição de vetores. Desta forma, o modelo é capaz de se sair bem tanto em casos onde ainda não se possui informações de instalação suficientes acerca do usuário (*cold-start*), assim como em casos onde já se possui uma boa quantidade de aplicativos em sua sequência de instalação.

Sendo assim, cada fator é representado por valores escalares que indicam a tendência de tal. Logo, sendo $f \in \mathbb{R}^f$, representaremos a tendência de cada fator de usuários como b_f^U, e de cada fator de itens como b_f^I (b deriva da palavra tendência em inglês, *bias*). Assim, temos que a tendência de cada usuário ou item pode ser dada pela soma das tendências de seus fatores. Ou seja, se a tendência de um usuário (item) possuir um fator é dada por quanto desse fator tal usuário (item) possui, a tendência do usuário (item) será a soma de quanto de cada fator, o usuário (item) possui. Portanto, para cada usuário u, sua tendência é dada pela Equação 4.13, sendo que o mesmo vale para a obtenção da tendência de um item em específico, matematicamente representada pela Equação 4.14.

$$b_u = \sum_{j \in f_u} b_j^U \tag{4.13}$$

Sendo que f_u representa todos os fatores latentes do usuário u. Da mesma forma, f_i representará todos os fatores do item i.

$$b_i = \sum_{j \in f_i} b_j^I \tag{4.14}$$

Isto posto, a estimativa de avaliação também deve levar em consideração as tendências de cada fator latente e, portanto será dada pela Equação 4.15.

$$\hat{r}_{ui} = f(q_i^T \cdot p_u + b_u + b_i) \tag{4.15}$$

Assim sendo, inicialmente foi utilizada uma matriz de tamanho $N_{usuarios} \times M_{apps}$, de valores binários, indicando se um determinado usuário u possui ou não o aplicativo a. Porém, no que consta ao número de fatores latentes, como o *LightFM* também utiliza o gradiente de descendência estocástico, foram realizados testes e através de uma função de perda, determinou-se que seriam utilizados 50 fatores. Logo, foram geradas duas matrizes, sendo a primeira de $Usuarios \times Fatores\_Latentes$ ($N_{users} \times 50$) e a segunda de $Itens \times Fatores\_Latentes$ ($M_{apps} \times 50$).

Assim, uma vez obtidas as estimativas de avaliação, para cada usuário, foram recomendados os seus N *apps* com maior possível avaliação.

Solução Proposta

Para inserção dos dados demográficos e de dispositivos do usuário no modelo, também foi considerado que cada informação adicional seria tratada como um aplicativo. Entretanto, como mostram os resultados obtidos no modelo LDA, na Subseção 4.3.1, a incorporação dos dados enriquecidos não pode ser totalmente igual ao que é feito com os *apps*. Isso porque apesar de o fato de tais informações serem tratadas como aplica-

Tabela 4.3: Valores associados a cada dado demográfico.

Tipo de Dado Demográfico	Classe	Valor Atribuído
Dispositivo	**Entrada**	1
	Intermediário	2
	Mid-high	3
	High-end	4
	Premium	5
População	**Pequena**	1
	Média	2
	Grande	3
Renda	**Baixa**	1
	Intermediária Baixa	2
	Média	3
(Em salários mínimos)	**Intermediária Alta**	4
	Alta	5

tivos trazer benefícios quantitativos e facilidade de implementação, é importante que o modelo possa compreender que se tratam de dados diferentes.

Para este fim, cada tipo de informação (Tamanho da população, renda média e categoria de preço do dispositivo) foi rotulada como um aplicativo. Porém, como os dados referentes aos *apps* são binários, ou seja, o usuário possui ou não um aplicativo a, a identificação de cada dado demográfico foi representado por um valor escalar, de acordo com a Tabela 4.3.

Vale ressaltar que a escolha de se representar cada característica adicionada, por um valor escalar, foi dada pela necessidade de se diferenciar a forma como aplicativos e dados demográficos e de dispositivo eram passados ao modelo.

Assim, se um usuário u, for residente de uma cidade de população média, em uma região com renda média intermediária alta e possuir um dispositivo *Mid-high*, por exemplo, ele possuirá os valores 2, 4 e 3, nas dimensões de população, renda e dispositivo, respectivamente. Dessa forma, cada usuário possuirá apenas 3 dimensões a mais, no que consta à matriz de $Itens \times Fatores\_Latentes$, que agora será de $(Itens + Tipos\_Dados\_Demograficos) \times Fatores\_Latentes$.

No que consta à recomendação, uma vez calculadas as novas estimativas de avaliação, serão desconsideradas as dimensões referentes às informações demográficas e recomendados os N *apps* com maior estimativa.

Resultados

A Figura 4.10 mostra os resultados de precisão dos modelos enriquecidos em relação à solução *Base*, para a recomendação de aplicativos. Como pode-se observar, as soluções propostas obtiveram resultados iguais ou superiores à solução *Base* na maioria dos casos, se sobrepondo a esta em todos os cenários apresentados de $s = 3$ e

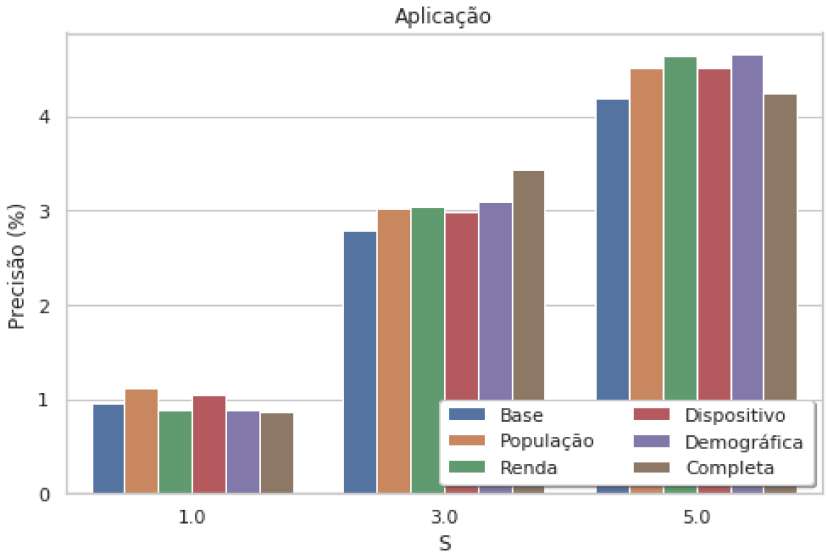
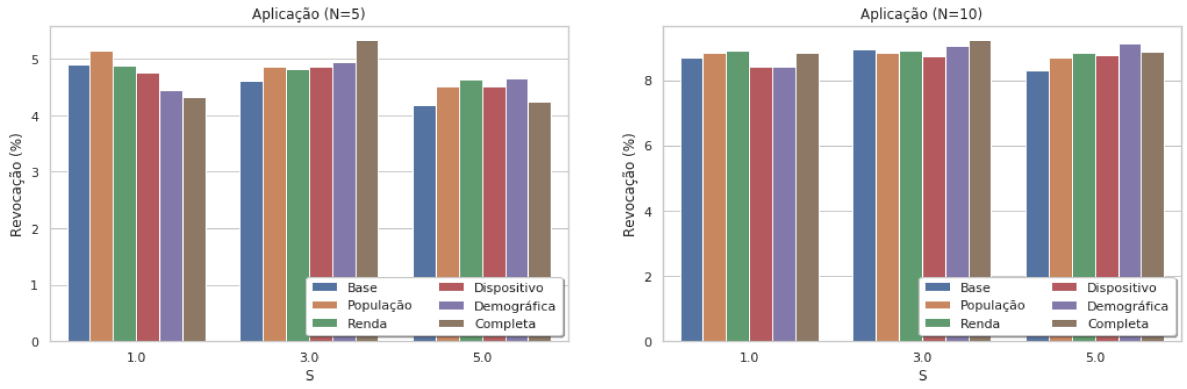


Figura 4.10: Resultados obtidos com os modelos do filtro colaborativo em termos de precisão, no contexto de aplicativos.

$s = 5$. Considerando os cenários apresentados, vale destacar o desempenho das soluções *População* e *Dispositivo*, que superaram a solução *Base* em todos os casos. Tal resultado indica que os pesos atribuídos aos seus respectivos dados podem ter melhor caracterizado os usuários, uma vez que o valor mais alto atribuído às categorias de cada solução fizeram referência também às categorias com um menor número de usuários. Tal situação pode ter contribuído para um melhor entendimento dos padrões de usuários de nicho. Além disso, vale destacar também o desempenho das soluções *Renda* e *Demográfica*, que se sobressaíram à solução *Base* quando $s = 3$ (9,23% e 11,52%, respectivamente) e $s = 5$ (aproximadamente 11% em ambos os casos).

Além disso, os resultados de revocação alcançados pelas soluções propostas mostraram que é interessante avaliar o cenário em que a recomendação será feita, visto que todas as soluções superaram a solução *Base* em algum cenário (Figura 4.11). Quanto à recomendação de 5 aplicativos, a Figura 4.11a mostra que o bom desempe-



(a) Recomendação de 5 aplicativos.

(b) Recomendação de 10 aplicativos.

Figura 4.11: Resultados de revocação obtidos no contexto de aplicativos, para os modelos de filtro colaborativo.

nho das soluções *População* e *Renda* também podem ser vistos quando se recomenda mais aplicativos, conseguindo resultados iguais (*Renda* em $s = 1$) ou superiores à solução *Base* em todos os cenários. Vale destacar também o desempenho da solução *Completa*, que conseguiu uma melhora de 15,2% em relação à *Base* quando analisado $s = 3$, mostrando que a agregação de informações de contexto do usuário podem ser benéficas. O comportamento não usual da solução *Completa*, de ter seu pico de desempenho quando $s = 3$, pode ser explicado pela dificuldade em se acertar muitos aplicativos para manter uma alta taxa de acerto, assim como esclarecido anteriormente na discussão dos resultados do modelo LDA.

Já quando recomendados 10 aplicativos, as soluções *População*, *Renda* e *Completa* mantiveram resultados iguais ou superiores à solução *Base* em todos os cenários, como pode ser visto na Figura 4.11b. Outro fato de destaque é que todas as soluções propostas se saíram melhor que a base quando $s = 5$, mostrando ser eficaz a utilização das informações de contexto analisadas nos cenários apresentados.

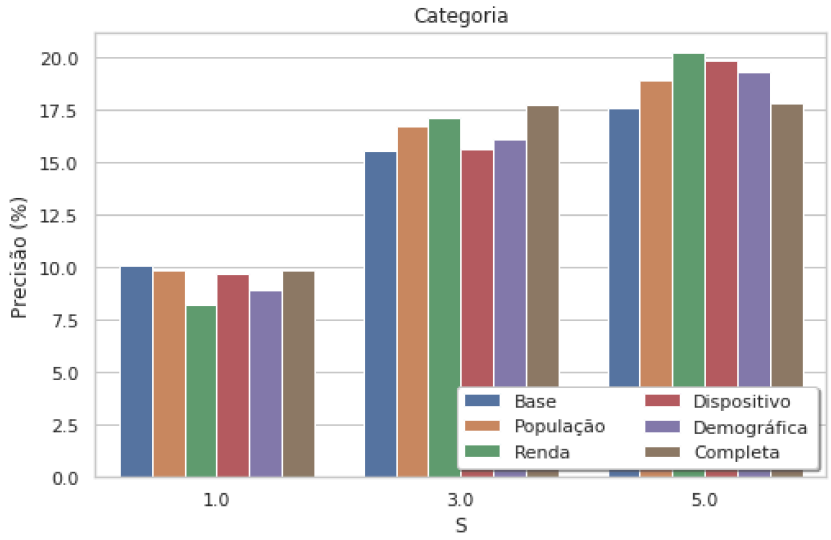
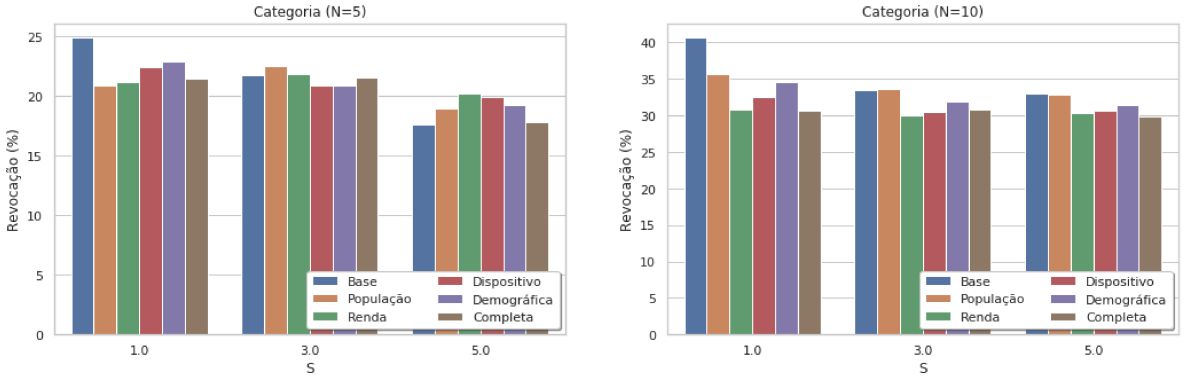


Figura 4.12: Resultados de filtro colaborativo, obtidos em termos de precisão, no contexto de categorias.

Já no que se refere às métricas de categorias, vemos na Figura 4.12 que todas as soluções propostas ultrapassaram a solução *Base* em algum cenário. Isso mostra que, para categorias, também é necessário identificar o momento adequado de se utilizar cada informação extra, já que algumas podem ser melhor empregadas quando analisados aplicativos instalados a longo prazo. Neste contexto, é interessante observar o aumento dos valores de precisão obtidos pelas soluções analisadas, tendo *Renda* alcançado aproximadamente 20% quando $s = 5$. Nesse mesmo cenário, tal solução aumentou em 15% a precisão da solução *Base*.

Da mesma forma, os resultados de revocação para categorias (Figura 4.13) também corroboram a hipótese de que é possível utilizar informações adicionadas, se



(a) Recomendação de 5 aplicativos.

(b) Recomendação de 10 aplicativos.

Figura 4.13: Resultados de revocação da abordagem de filtro colaborativo, obtidos no contexto de categorias.

avaliado o contexto. Isso porque, no que compete à análise dos últimos 5 *apps* instalados pelo usuário (Figura 4.13a) somente as soluções *População* e *Renda* obtiveram resultados iguais ou melhores que a solução *Base* quando $s = 1$ ou $s = 3$. Já quando recomendados 10 *apps* (Figura 4.13b), nenhuma das soluções propostas superaram a solução *Base*, apesar de aumentarem os resultados de revocação consideravelmente. Tal situação pode indicar uma priorização de aplicativos de determinadas categorias durante a recomendação. Apesar disso, vale a pena ressaltar que os modelos não foram treinados com base em categorias, sendo estas sido colhidas dos aplicativos já recomendados.

Conclusões

Com base nos resultados obtidos, foi possível verificar a efetividade do uso de informações demográficas, tendo as soluções *População* e *Renda* se saído bem na maioria dos casos. Grande parte desta efetividade, no entanto, pode ter influenciado a solução *Completa*, já que tal solução também apresentou resultados interessantes. Por outro lado, as informações referentes a soluções como *Dispositivo* e *Demográfica*, se mostraram adequadas a certos tipos de casos, o que mostra a necessidade de se aprofundar mais nas situações em que tais casos ocorrem.

Já no que compete à utilização do modelo, ainda há espaço para melhorias, sendo possível a análise de uma outra forma de incorporação dos dados enriquecidos ao perfil do usuário.

4.3.3 Matrizes de Transição de *Markov* (MTM)

A Matriz de Transição de Markov (MTM), ou matriz estocástica, é uma matriz construída para descrever a probabilidade de transições em uma Cadeia de Markov (Ross, 2014). Por sua vez, a Cadeia de Markov é um processo estocástico utilizado para observar eventos ao longo do tempo. Tal observação é dada de forma que cada evento em um processo representa um estado e, a partir de tal, outros eventos podem ocorrer com alguma probabilidade. Assim, temos que em uma Cadeia de Markov, cada estado tem uma probabilidade de ocorrer, dado o que se encontrava anteriormente. Da mesma forma, o estado anterior só foi possível porque outro foi adotado antes dele, e assim sucessivamente.

Logo, seja $\{X_n, n = 0, 1, 2, \dots\}$ um processo que assume um determinado valor i em algum momento n do tempo ($X_n = i$). Assim, a probabilidade de tal processo assumir o valor j, sendo que estava em i, é dado por P_{ij}, sendo que

$$P_{ij} = P\{X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_1 = i_1, X_0 = i_0\} \tag{4.16}$$

Além disso, considerando que sempre é feita uma transição de um estado para outro e que a probabilidade de tal transição é sempre não-negativa,

$$\sum_{j=0}^{\infty} P_{ij} = 1 \quad , i = 0, 1, \dots \tag{4.17}$$

Logo, seja P a matriz de transição de markov que pode ser dada por:

$$\mathbf{P} = \begin{Vmatrix} P_{00} & P_{01} & P_{02} & \dots \\ P_{10} & P_{11} & P_{12} & \dots \\ \vdots & \vdots & \vdots & \\ P_{i0} & P_{i1} & P_{i2} & \dots \\ \vdots & \vdots & \vdots & \end{Vmatrix}$$

Tal matriz P pode também ser chamada de matriz de transição de um passo, pois as probabilidades nela contidas representam a probabilidade de um estado ser atingido logo após outro. Entretanto, a matriz de transição de *Markov* pode também ser calculada com K passos, caso em que as probabilidades serão calculadas com base em um estado ser atingido após outro, sendo que outros $k - 1$ estados foram acessados entre eles. Isto é, em uma matriz de transição de dois passos ($K = 2$), será calculada a probabilidade de um estado j ser atingido após um estado i, dado que depois de i foi acessado um outro estado qualquer.

Solução Base

Com base nos procedimentos detalhados acima, a solução base foi construída de forma a obter padrões de instalação entre *apps* através da análise das probabilidades contidas na matriz de transição (Cheng et al., 2018). Assim sendo, seja $app_1, \ldots, app_{i-1}$ a sequência de aplicativos instalados pelo usuário. A probabilidade do usuário instalar app_i é dada por

$$p(app_i|app_{i-1}, \ldots, app_1) \tag{4.18}$$

Entretanto, vamos considerar que apenas os k últimos aplicativos irão influenciar nas próximas escolhas do usuário. Nesse caso, adaptando a Equação 4.18, temos que

$$p(app_i|app_{i-1}, app_{i-2}, \ldots, app_1) \cong p(app_i|app_{i-1}, \ldots, app_{i-k}) \tag{4.19}$$

Além disso, para analisarmos completamente a chance de um aplicativo ser instalado dados os últimos k *apps*, somente uma matriz de transição de k passos não é suficiente. Isso porque esta iria analisar a chance do *app* ser instalado, não importando os $k-1$ outros aplicativos instalados entre ele e o k-ésimo *app*. Assim sendo, vamos construir k matrizes de transição, $T_1, \ldots, T_k$, onde cada uma irá analisar a probabilidade de um *app* ser instalado em comparação com um dos k últimos aplicativos.

Sendo a matriz $\{T_j, j = 1, \ldots, k\}$ de tamanho $n_{apps} \times n_{apps}$, a posição (r, s) de tal matriz deve representar a probabilidade do app_s ter sido instalado após app_r, sendo que $j-1$ outros aplicativos foram instalados entre eles. Para calcular tal probabilidade, será feita a contagem de quantas vezes tal situação ocorreu, dividindo-a pelo número de vezes que app_r foi instalado. Assim,

$$p(app_i = s|app_{i-j} = r) = \frac{\sum_{app_{i-j}, app_i \in D} \mathbb{1}(app_{i-j} = r, app_i = s)}{\sum_{app_{i-j} \in D} \mathbb{1}(app_{i-j} = r)} \tag{4.20}$$

, onde D representa o conjunto de dados de treino e j representa o número de aplicativos instalados entre a sequência analisada na matriz T_j.

Finalmente, para que uma recomendação seja feita, serão utilizados os k últimos aplicativos instalados pelo usuário, consultando a probabilidade de todos os *apps* serem instalados nas tabelas correspondentes. Ou seja, será obtido o vetor de probabilidades referente à linha do j-ésimo último *app* instalado, na matriz T_j. Ao final, com a soma dos k vetores de probabilidade obtidos, tem-se as probabilidades gerais de cada aplicativo ser instalado. Logo, como serão recomendados N *apps*, foram selecionados os N aplicativos com maior probabilidade, excluídos aqueles já possuídos pelo usuário.

Vale ressaltar ainda que, neste trabalho foi considerado que apenas os dois últimos

aplicativos instalados pelo usuário iriam influenciar as próximas escolhas, assim como em Cheng et al. (2018). Assim, foram calculadas apenas as matrizes de transição T_1 e T_2, sendo que T_1 indica a probabilidade de um *app* ser instalado logo após o outro, e T_2 representa a probabilidade de um aplicativo ser instalado após o outro, sendo que outro *app* qualquer foi instalado entre eles.

Além disso, conforme mencionado na Seção 4.1, em algumas situações não é possível estabelecer a ordem correta em que os aplicativos foram instalados. Nesses casos, seguimos uma ordem aleatória, uma vez que forçar uma sequência válida, utilizando somente um aplicativo por dia de instalação, poderia desconsiderar algum *app* semanticamente importante.

Solução Proposta

Para incorporar as informações enriquecidas do usuário, além das matrizes de transição T_1 e T_2, foram calculadas matrizes de transição específicas para cada tipo de informação adicional. Cada matriz adicional representa a probabilidade de um aplicativo ser instalado em cada classe de informação. Isto é, foi criada uma matriz para representar o tamanho da população, outra para representar renda média, e uma terceira representando o preço do dispositivo. Na matriz População, por exemplo, cada linha representa uma classe desse tipo de informação (população pequena, média e grande). Assim, a linha de população média, por exemplo, possui a probabilidade de cada aplicativo ser instalado, dado que o usuário reside em uma cidade cujo tamanho da população é médio.

Logo, considerando c como uma classe de informação qualquer e s o aplicativo avaliado, a probabilidade contida em uma célula (c, s) é dada por

$$p(s|c) = \frac{\sum_{u \in U} \mathbb{1}(c \in u_{class}, s \in u_D)}{\sum_{app_i \in D} \mathbb{1}(app_i == s)} \tag{4.21}$$

, onde u_{class} é a classe de informação adicional que o usuário possui e u_D o conjunto de dados de treino do usuário.

Uma vez obtidas as matrizes de transição de cada tipo de informação, estas podem ser agregadas ao modelo convencional de recomendação. Para isto, basta obter também os vetores de probabilidade de acordo com a classe de informação adicional possuída pelo usuário. Por exemplo, se o usuário possui somente a classe de tamanho de população grande, somente o vetor correspondente a esta classe, na matriz de população, será somado às probabilidades das matrizes T_1 e T_2. Entretanto, para agregar relevância às informações adicionais, foram multiplicados pesos λ a cada vetor adicional. Assim, o vetor de probabilidades dos aplicativos são dados por

$$P_{apps} = \vec{t_1} + \vec{t_2} + \lambda_p \cdot \vec{t_p} + \lambda_r \cdot \vec{t_r} + \lambda_d \cdot \vec{t_d} \tag{4.22}$$

onde os vetores t são os vetores obtidos das matrizes de transição, sendo que t_p, t_r e t_d são derivados das matrizes de população, renda e dispositivo, respectivamente. Do mesmo modo, λ_p, λ_r e λ_d são os pesos aplicados a cada tipo de informação.

Tabela 4.4: Valores aplicados aos pesos para cada solução implementada.

Solução	Pesos
População	$\lambda_p = 10$
Renda	$\lambda_r = 4$
Dispositivo	$\lambda_d = 4$
Demográfica	$\lambda_p = 8$, $\lambda_r = 2$
Completa	$\lambda_p = 8$, $\lambda_r = 4$, $\lambda_d = 2$

Neste trabalho, para escolha dos valores aplicados a cada peso foi aplicada a técnica de *GridSearch*. Ao final, cada solução abordada possuiu um conjunto de pesos específicos, assim como mostrado na Tabela 4.4 acima.

Resultados

A Figura 4.14 mostra os resultados alcançados pelas soluções implementadas, utilizando o modelo MTM, em termos de precisão. Nela, podemos perceber que os resultados das soluções propostas mostram que todas se sobressaíram em relação à solução *Base*, obtendo resultados no mínimo similares. Tais resultados podem indicar a eficácia da aplicação de pesos às informações demográficas e de dispositivos adicionadas. De fato, apesar de o maior aumento representar cerca de 0,5 ponto percentual na solução *Completa*, quando analisados os últimos 5 aplicativos do usuário, por se tratar de um problema cujas taxas de precisão são naturalmente baixas, tal aumento pode ser significativo dependendo do contexto.

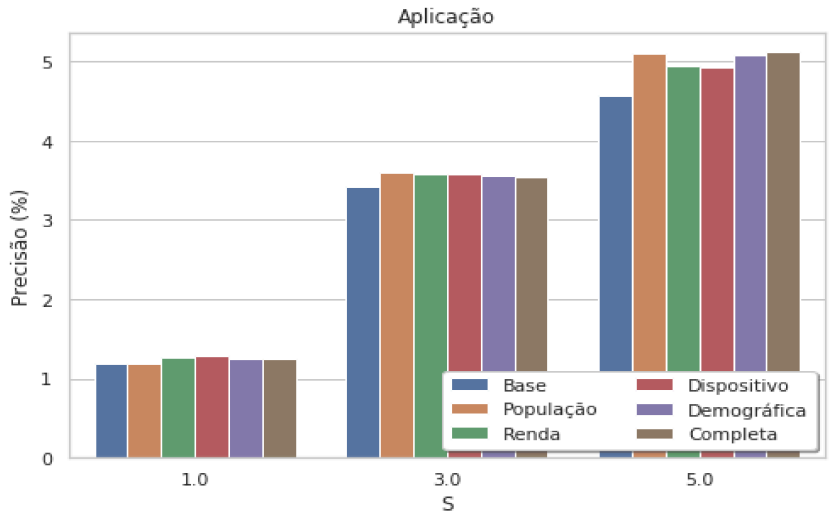
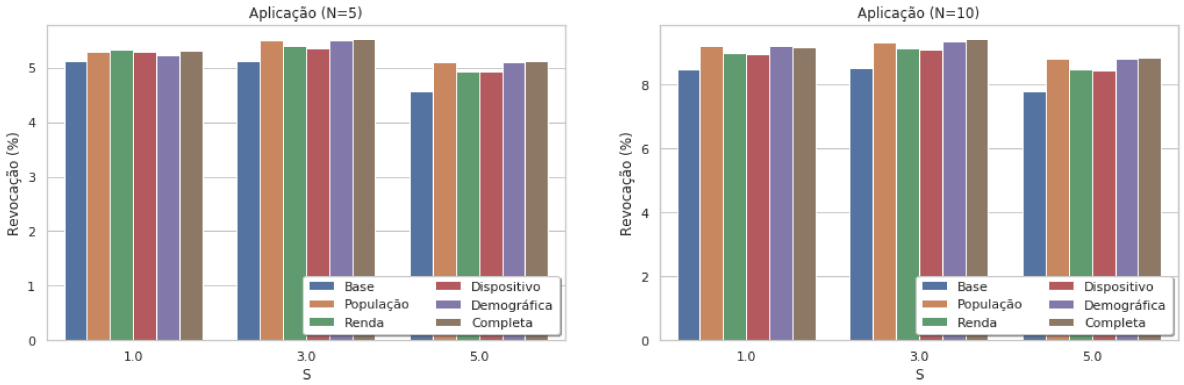


Figura 4.14: Resultados obtidos com os modelos do MTM em termos de precisão, no contexto de aplicativos.



(a) Recomendação de 5 aplicativos.

(b) Recomendação de 10 aplicativos.

Figura 4.15: Resultados de revocação dos modelos MTM, obtidos no contexto de aplicativos.

Da mesma forma, os resultados de revocação, explicitados na Figura 4.15, também mostram uma melhora das soluções propostas em relação à solução *Base*. Tais resultados só fortalecem a hipótese de que a adição de informações demográficas e de dispositivos, pode levar a recomendações mais assertivas de aplicativos, se devidamente aliada a uma forma de agregar maior importância a tais dados, matematicamente. Isto porque a taxa de aumento das soluções se manteve semelhante à alcançada na métrica de precisão, com mais soluções alcançando até 0,5 ponto percentual de melhora (soluções *População*, *Demográfica* e *Completa*), quando recomendados 5 aplicativos (Figura 4.15a).

Além disso, como mostrado na Figura 4.15b, ao se recomendar 10 aplicativos os resultados são ainda melhores, com aumentos de no mínimo 0,5 pontos percentuais (*Dispositivo* em $s = 1$). Ademais, a solução *Completa* chegou a alcançar um aumento de cerca de 13,7% em relação à solução *Base* quando levado em consideração os 5 últimos *apps* instalados. Além disso, a solução *População* também obteve um dos melhores resultados observados. Isso mostra que ao se analisar o peso da informação de tamanho populacional, é possível aumentar seu potencial, elevando também a taxa de acerto das soluções mais completas.

Já no que se refere à análise de categorias, na Figura 4.16 é possível visualizar os resultados obtidos. Nela é possível perceber que a maioria das soluções obteve resultados parecidos ao da solução *Base*, com exceção da *População*. Isso indica que tais soluções podem ser utilizadas mesmo levando em consideração somente as categorias, já que a diferença entre os resultados observados é muito pequena. Entretanto, vale ressaltar que as soluções propostas obtiveram resultados ligeiramente superiores à *Base* apenas quando $s = 1$.

Entretanto, quando analisados os resultados de revocação, apresentados na Figura 4.17, é possível perceber que as soluções propostas não alcançaram os mesmos resultados obtidos pela solução *Base*. Ainda assim, algumas soluções se aproximaram

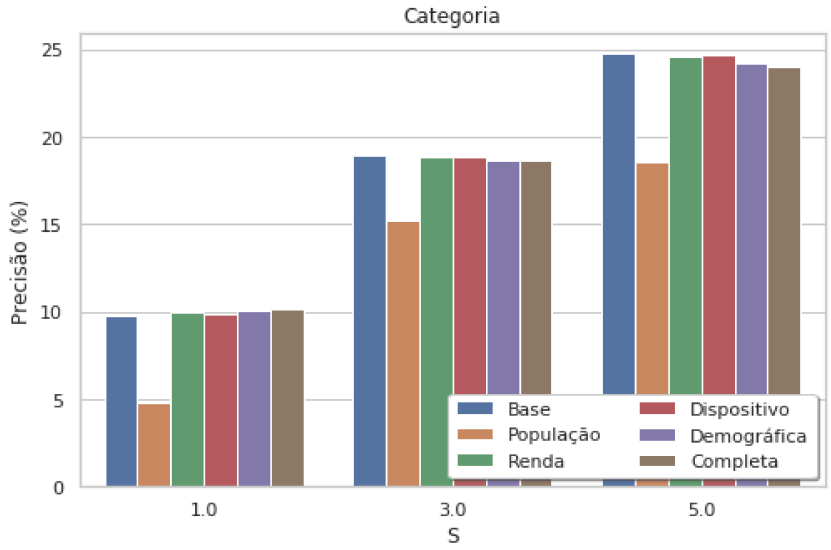
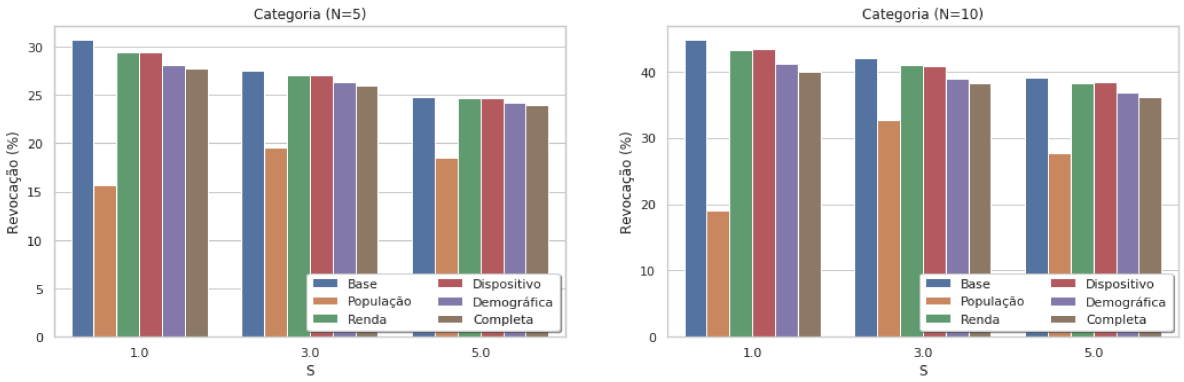


Figura 4.16: Resultados obtidos em termos de precisão, no contexto de categorias, dos modelos MTM.

bastante, como é o caso das soluções *Renda* e *Dispositivo*. Ao se recomendar 5 aplicativos (Figura 4.17a) e analisando $s = 5$, ambas as soluções *Renda* e *Dispositivo* obtiveram cerca de 24,6% de revocação, representando 99,6% da solução *Base*. Ao se recomendar 10 aplicativos (Figura 4.17b) com os mesmos últimos 5 *apps* instalados pelo usuário, a solução *Dispositivo* foi a melhor, atingindo 98% da solução *Base*.

Os resultados apresentados acima podem indicar que as recomendações errôneas cometidas pelas soluções propostas, não acertaram também na categoria do *app* que seria escolhido. Esta situação faria com que seus resultados em termos de aplicativos fossem bons, porém, ao se analisar as categorias, houvesse uma diminuição de revocação por parte das soluções propostas. Outro fator que pode ter impactado negativamente tais soluções, foi o fato de que os pesos utilizados para cada solução podem



(a) Recomendação de 5 aplicativos.

(b) Recomendação de 10 aplicativos.

Figura 4.17: Resultados de revocação obtidos no contexto de categorias, pelos modelos MTM.

fazer com que categorias de aplicativos mais relacionadas às informações demográficas e de dispositivos abordadas, fossem privilegiadas durante a recomendação. Tal privilégio poderia fazer com que mais *apps* de tais categorias fossem recomendados, diminuindo a chance de aplicativos de categorias mais relacionadas ao usuário fossem escolhidos.

Conclusões

Com base nos resultados obtidos, é possível perceber que a utilização das informações demográficas obteve resultados positivos, em relação a aplicativos. Tal desempenho pode ser derivado do uso de pesos de forma mais explícita. Além disso, a solução *Renda*, que se destacou nos modelos anteriores, obteve resultados equiparáveis ou superiores à *Base* na maioria dos cenários, o que pode indicar sua eficácia na recomendação de aplicativos. Entretanto, os resultados em relação às categorias não atingiram o esperado, indicando que talvez a generalização dos pesos possa impactar negativamente os resultados.

Assim, podem ser realizadas melhorias na forma como tais pesos são escolhidos, buscando formas de adaptá-los também ao contexto das categorias. Do mesmo modo, podem ser estudadas outras técnicas de inserção dos pesos no modelo, de forma a ampliar ainda mais o impacto das informações adicionadas. Outro fator importante que pode ter interferido nos resultados obtidos, é que modelos que se baseiam em transições, como o MTM, utilizam sequências de acontecimentos bem definidas. Entretanto, a sequência de aplicativos utilizada não é verificadamente correta, visto que se mais de um *app* é instalado no mesmo dia, não há como definir qual foi adotado primeiro.

4.4 Discussões Gerais

Com base nos resultados apresentados nas seções anteriores, foi feita uma comparação entre as abordagens utilizadas, avaliando seu uso para cada solução proposta. Além disso, também foi analisado o desempenho de cada solução em termos dos resultados obtidos pela solução base. Assim, considerando que se deseja recomendar aplicativos e verificar o desempenho geral alcançado, foi realizada uma análise mais profunda dos resultados considerando apenas os últimos cinco aplicativos instalados pelo usuário ($s = 5$). Além disso, para analisar a revocação dos modelos, estabelecemos uma recomendação de 10 aplicativos ($n = 10$).

Logo, analisando o desempenho geral das soluções propostas em termos de precisão, é possível perceber que houve melhora dos resultados na maioria dos casos (Figura 4.18). Tal análise é possível pelo fato de que o resultado das soluções base, representadas por barras translúcidas, esteve no máximo igual aos resultados das soluções propostas dos modelos LDA, Filtro Colaborativo (FC) e MTM.

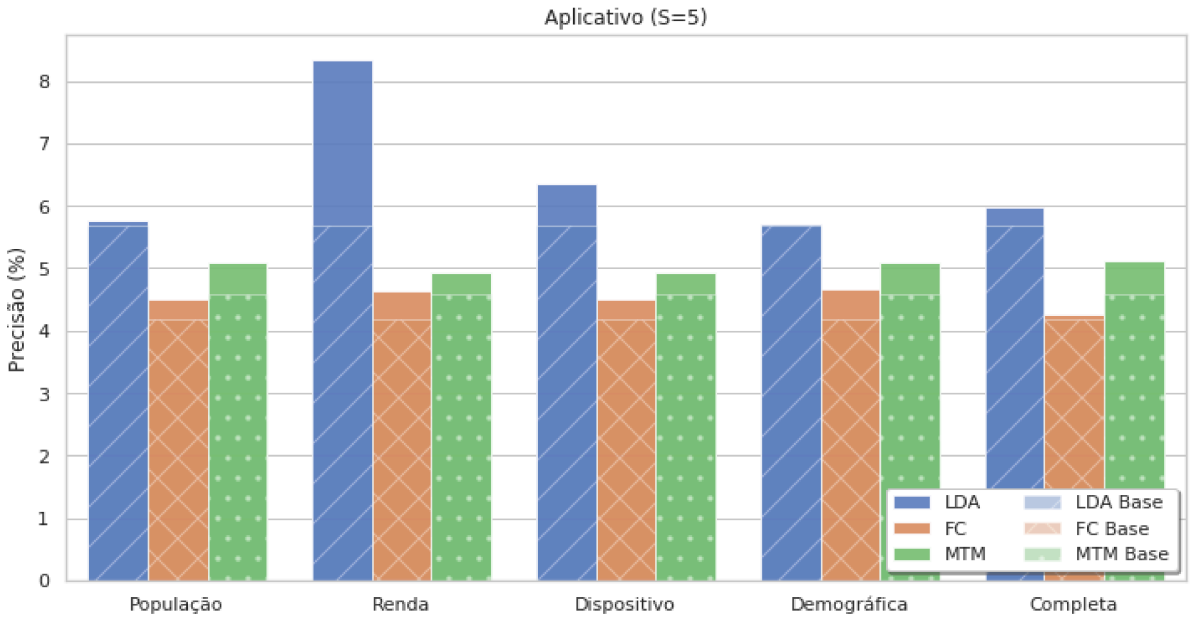


Figura 4.18: Resultados gerais de precisão obtidos com as soluções propostas, com base na recomendação de aplicativos.

Os modelos MTM e FC apresentaram as melhoras mais estáveis entre si, com a maioria das soluções propostas ultrapassando a solução *Base* em aproximadamente 11,5%. Os valores alcançados pelos modelos podem ser resultantes dos pesos utilizados para identificação das informações adicionadas (FC) e agregação das matrizes de transição (MTM), já que estes podem ter suavizado o impacto das características adicionadas. Outro fator que pode ter interferido nos resultados do MTM, é a quantidade de aplicativos que acredita-se influenciar na próxima instalação (Valor de K nos

modelos). Neste trabalho foi utilizado $K = 2$ para o modelo; entretanto uma avaliação mais profunda quanto à utilização de outros valores, como feito para o modelo LDA, pode apresentar resultados positivos.

O modelo LDA atingiu os melhores resultados, tanto em termos da solução *Base*, como das soluções propostas. No entanto, o desempenho de tais soluções não foi tão estável quanto dos modelos MTM, com as soluções *Demográfica* e *População* obtendo resultados muito similares à *Base*. Isso pode ter relação com o fato de o tamanho da população não separar bem os usuários em termos de seus aplicativos nesta situação. Isso porque, com a maioria dos usuários residindo em cidades de população média (Figura 4.2), os aplicativos instalados por tais usuários podem abranger um grande número de aplicativos semanticamente distintos. Além disso, como *apps* e características demográficas e de dispositivo foram inseridas da mesma forma no modelo LDA, características mais gerais (e.g., tamanho da população) podem ser consideradas de pouca relevância. No entanto, conhecer a renda média da região do indivíduo parece prover informações mais valiosas acerca do tipo de aplicativo instalado. A melhora com essa informação chegou a quase 3,5 pontos percentuais, o que representa aproximadamente 64% a mais que solução *Base*.

Já o gráfico da Figura 4.19 mostra os valores médios de revocação obtidos em recomendações de dez aplicativos ($n = 10$). Nele, é possível perceber que mais uma vez, os resultados do modelo LDA atingiram os melhores resultados, e com destaque da solução *Renda*. O desempenho desta superou os resultados da *Base* em aproximadamente 27,91% para o modelo citado, além de também se sair melhor que a base considerando os outros modelos analisados. Da mesma forma, a solução *Dispositivo* também obteve destaque, ainda que apresentando crescimento mais moderado (1,64%). Tais destaques podem indicar que, em modelos LDA, é preferível a adição

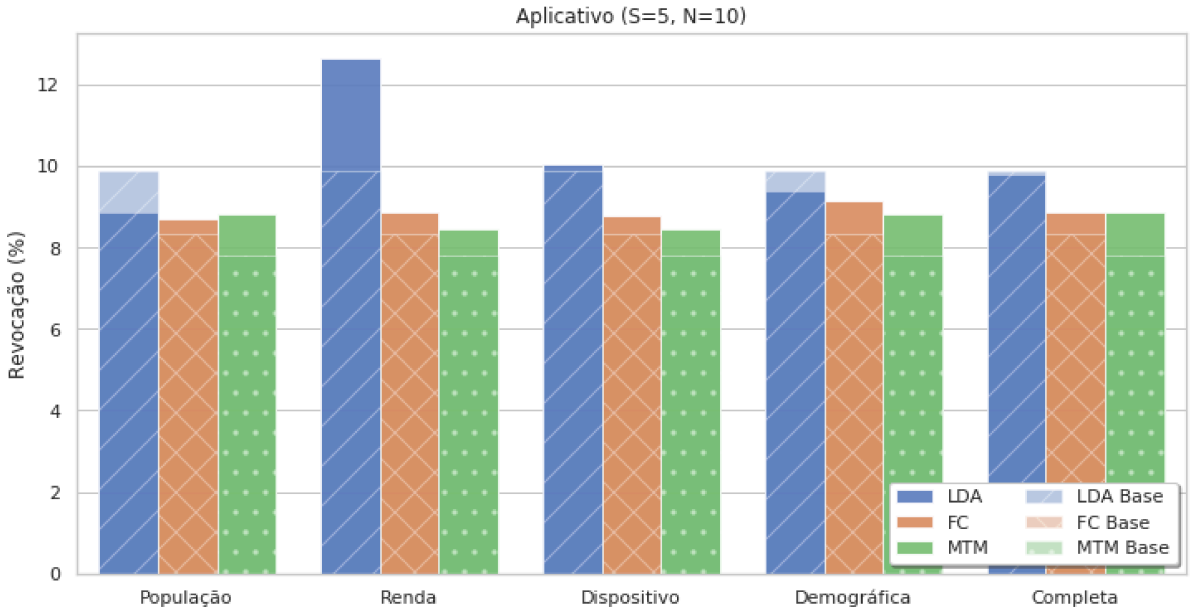


Figura 4.19: Resultados gerais de revocação obtidos ao recomendar aplicativos.

de informações que dividam usuários em grupos menores e mais relacionados a uma característica demográfica mais específica.

Além disso, os resultados obtidos pelos modelos MTM e FC foram mais uma vez similares. Todas as soluções dos dois modelos obtiveram resultados parecidos, se saindo melhor quando utilizadas as informações adicionais. Os destaques são as soluções *População*, *Demográfica* e *Completa*, que obtiveram maior vantagem (aproximadamente 13,5%) em relação à *Base* para o modelo MTM; e a solução *Demográfica* que obteve vantagem de aproximadamente 10% em relação à base do modelo de filtro colaborativo. Tais resultados reforçam a hipótese de que é necessário que o modelo possa dar mais importância às informações adicionadas, para que estas sejam devidamente aproveitadas.

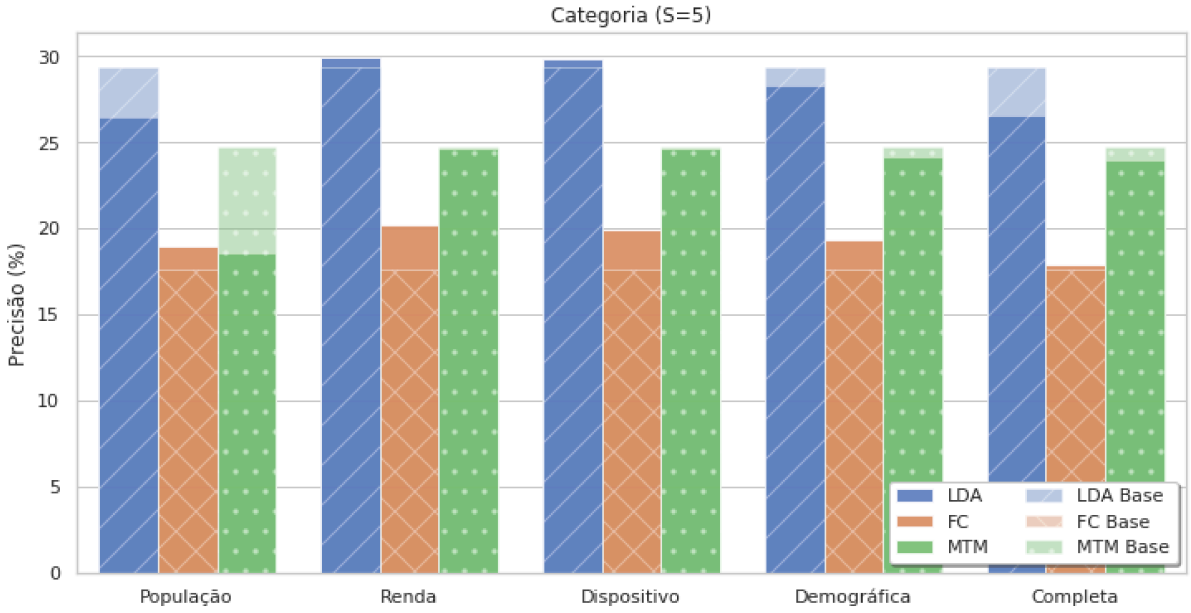


Figura 4.20: Resultados gerais de precisão ao se analisar categorias de aplicativos.

Já no que compete aos resultados de categoria, o gráfico da Figura 4.20 mostra que as soluções propostas alcançaram resultados relativamente satisfatórios. Isso porque somente as soluções de filtro colaborativo obtiveram todos os resultados iguais ou superiores à respectiva solução base. Quanto aos outros modelos, somente as respectivas soluções *Renda* e *Dispositivo* alcançaram resultados ligeiramente superiores que as soluções base.

Tais resultados mostram que a adição das informações demográficas e de dispositivo pode priorizar a recomendação de *apps* de certas categorias, fazendo com que se diminua a diversidade na recomendação. Vale destacar que as melhores soluções se mostraram ser as de *Renda* e *Dispositivo*, mais uma vez, já que foram as únicas a manter resultados no mínimo iguais às soluções base, em todos os modelos avaliados. Tais valores podem indicar que a adição de características que sejam mais específicas

ao usuário, podem fazer com que sejam identificados mais padrões de instalação do mesmo.

Por fim, quanto aos resultados de revocação para categorias, vemos na Figura 4.21 que somente a solução *População* obteve resultados iguais aos da solução *Base*. Quanto aos demais resultados, as soluções propostas não conseguiram superar as respectivas bases. Tais resultados corroboram com a hipótese de que a adição de informações de contexto pode fazer com que se priorize certas categorias, diminuindo sua diversidade na recomendação.

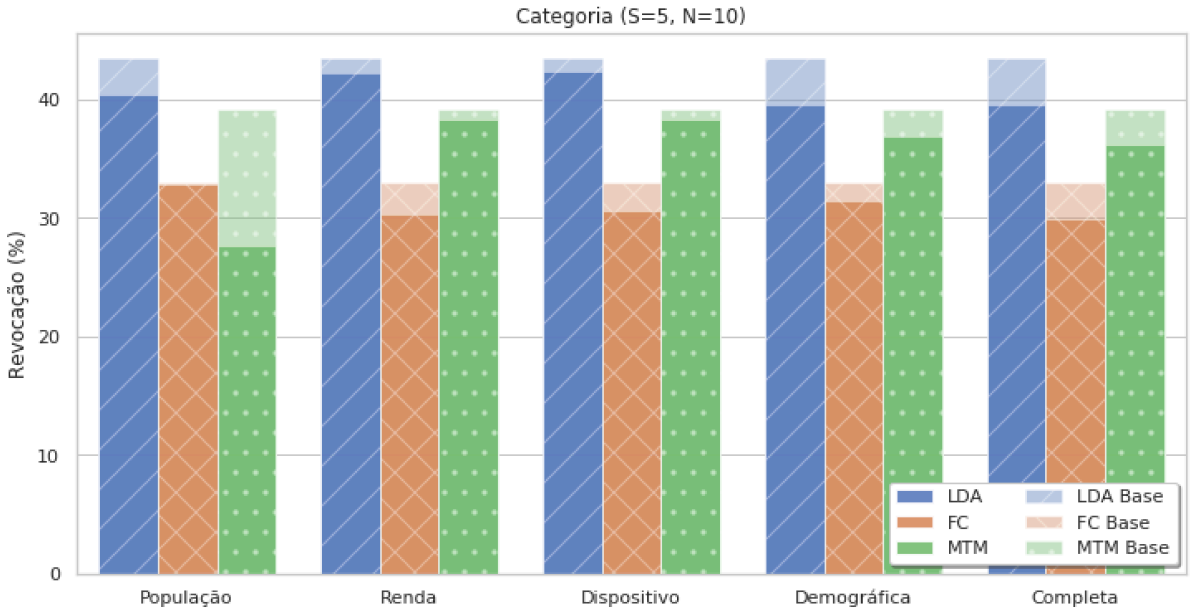


Figura 4.21: Resultados de revocação obtidos pelas soluções, levando em consideração as categorias dos aplicativos recomendados.

Com base nos resultados discutidos acima, percebeu-se um maior desempenho geral quando utilizada a abordagem LDA. Tais resultados podem ter relação com o fato de modelos LDA serem capazes de avaliar relações entre vários termos (aqui representados por aplicativos) e em diferentes contextos. Já os modelos Filtro Colaborativo e Matrizes de Transição de *Markov* obtiveram resultados menos eficazes, provavelmente pelo fato de analisarem apenas relações par-a-par entre os aplicativos. Além disso, modelos MTM baseiam-se na relação entre aplicativos instalados de forma sequencial, o que não pôde ser sempre confirmado com os dados utilizados.

Quanto à adição de dados demográficos e de dispositivos para recomendação de aplicativos, a informação de renda se provou a mais eficaz, atingindo melhores resultados quando aplicada a modelos LDA. A renda média do setor censitário em que o usuário reside é um fator relevante para o contexto socioeconômico em que o indivíduo está inserido, que por sua vez é determinante para os interesses dos usuários. A adição da categoria de dispositivo também obteve bons resultados, porém não equiparáveis aos de renda, o que pode ter sido causado pelo fato de que *apps* instalados

em dispositivos com poucos recursos nem sempre espelham os reais interesses do usuário. Já o tamanho da população não agregou conhecimento aos modelos, o que pode ser explicado pelo fato de ser uma informação generalista, que agrupa muitos indivíduos com interesses e características socioeconômicas distintas.

Já no que diz respeito às categorias, não foram encontrados benefícios com a incorporação de dados demográficos. Uma possível explicação seria o fato de os modelos não terem sido treinados com base em categorias, e sim em aplicativos. Assim, é importante que futuramente sejam criados modelos baseados na instalação de categorias de aplicativos, para que possa se aprofundar nos motivações dos resultados obtidos.

Capítulo 5

Conclusão

A pesquisa apresentada nesta dissertação investigou o impacto do uso de informações demográficas e de dispositivos no contexto de usuários, para a recomendação de aplicativos móveis. Para isso, tais informações foram agregadas a um perfil de usuário que continha somente registros dos aplicativos já instalados, uma localização aproximada e o nome do dispositivo móvel utilizado. A partir de tais dados, foi possível avaliar os resultados obtidos com a recomendação de aplicativos por três estratégias diferentes: *Latent Dirichlet Allocation* (LDA), Cadeias de Markov (MTM), e Filtro Colaborativo.

Além disso, foi elaborado um questionário que teve como propósito a investigação dos fatores que levam um indivíduo a instalar algum aplicativo. Dentre as respostas observadas, nota-se a relevância das recomendações feitas por pessoas próximas ao usuário, já que este meio foi o que mais obteve quantidade de notas máximas no quesito satisfação. Além disso, percebeu-se que somente 17 das 270 pessoas que participaram declararam não instalar aplicativos recomendados por pessoas próximas. Ademais, os resultados apontaram que poucos usuários consideram popularidade e recomendação de mídias como uma boa motivação de instalação. Por fim, segundo os participantes, a finalidade, segurança e os benefícios são os principais fatores por trás de uma instalação. Esses resultados mostraram que existe um espaço para melhoria nos sistemas de recomendação para atender melhor aos interesses dos usuários.

Já quanto aos resultados obtidos com a aplicação dos modelos, foi possível verificar que as soluções *Renda*, *População* e *Dispositivo* obtiveram resultados gerais mais benéficos em comparação com as outras. Tais soluções são indicadas quando recomendados aplicativos, já que os resultados gerais para categorias não superaram as soluções base. Entretanto, vale ressaltar que os modelos foram treinados com aplicativos, e não com categorias.

Além disso, percebeu-se uma superioridade do modelo LDA em comparação com os demais, principalmente com a utilização da renda do usuários. Isso pode ter relação com o fato de que modelos LDA em geral trabalham com análises de correlação entre termos, o que pode ter auxiliado na descoberta de aplicativos instalados em conjunto, mesmo a longo prazo. Entretanto, esta análise é inversa às suposições de modelos MTM, por exemplo, que trabalham através da ideia de que somente aplica-

tivos instalados a curto prazo irão interferir nos próximos a serem adotados. Quanto ao filtro colaborativo, percebeu-se uma melhora geral quando adicionadas informações demográficas e de dispositivo do usuário, porém o desempenho do modelo não atingiu grandes marcas.

5.1 Trabalhos Futuros

Com base nas discussões levantadas anteriormente, um próximo passo para esta pesquisa seria avaliar a inserção de pesos de forma explícita em modelos LDA, no que compete à adição de informações extras. Isso porque, apesar de ter alcançado bons resultados, percebeu-se que o modelo não foi capaz de extrair os potenciais padrões das características adicionadas. Além disso, como os modelos foram treinados somente com aplicativos, um próximo passo seria o treinamento também com categorias, avaliando os prós e contras de cada abordagem. Também, seria possível estudar os casos específicos em que cada abordagem se saiu melhor, levantando os prós e contras de cada uma. Com isso, seria possível construir um modelo híbrido que pudesse escolher o melhor modelo de acordo com casos específicos de cada usuário. Por fim, avaliar o impacto da incorporação de laços sociais nos modelos, uma vez que este se mostrou um tópico relevante durante a análise do questionário.

5.2 Publicações

Através da pesquisa desenvolvida nesta dissertação, foi possível publicar um artigo no Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP 2020) (Souza et al., 2020) que abordou a análise preliminar aqui apresentada. Vale ressaltar que tal artigo recebeu Menção Honrosa no evento no qual foi publicado.

Além disso, outros trabalhos foram desenvolvidos durante o período de mestrado. Um deles diz respeito a uma pesquisa realizada a fim de solucionar o problema *Indoor-Outdoor Detection*, publicado na revista *Big Data Research* (Souza et al., 2021). Também foi realizada uma participação na escrita de um capítulo de livro acerca de conceitos e técnicas para manipulação de dados geoespaciais (Domingues et al., 2020), publicado nas Jornadas de Atualização em Informática (JAI 2020).

Referências Bibliográficas

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Cheng, V. C., Chen, L., Cheung, W. K., and Fok, C.-k. (2018). A heterogeneous hidden markov model for mobile app recommendation. *Knowledge and Information Systems*, 57(1):207–228.

Domingues, A., Silva, F., Santos, L., Souza, R., Coimbra, G., and Loureiro, A. A. F. (2020). Dados geoespaciais: Conceitos e técnicas para coleta, armazenamento, tratamento e visualização. *Sociedade Brasileira de Computação*.

Frey, R. M., Xu, R., Ammendola, C., Moling, O., Giglio, G., and Ilic, A. (2017). Mobile recommendations based on interest prediction from consumer's installed apps–insights from a large-scale field study. *Information Systems*, 71:152 – 163.

Goel, S. and Kumar, R. (2018). Folksonomy-based user profile enrichment using clustering and community recommended tags in multiple levels. *Neurocomputing*, 315:425–438.

GSMA (2020). The Mobile Economy - The Mobile Economy.

IBGE (2020). [Online; accessed 5. Nov. 2020].

IBGE (2021). Censo Demográfico | IBGE. [Online; accessed 7. Jan. 2021].

Ipea (2020). Ipea. [Online; accessed 25. Fev. 2021].

Jacobi, J. A., Benson, E. A., and Linden, G. D. (2006). Recommendation system. (US7908183B2).

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Kula, M. (2015). Metadata embeddings for user and item cold-start recommendations. *arXiv preprint arXiv:1507.08439*.

Liu, B., Kong, D., Cen, L., Gong, N. Z., Jin, H., and Xiong, H. (2015). Personalized mobile app recommendation: Reconciling app functionality and user privacy preference. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 315–324, New York, NY, USA. ACM.

Liu, B., Wu, Y., Gong, N. Z., Wu, J., Xiong, H., and Ester, M. (2016). Structural analysis of user choices for mobile app recommendation. *ACM Trans. Knowl. Discov. Data*, 11(2):17:1–17:23.

Ma, Q., Muthukrishnan, S., and Simpson, W. (2016). App2vec: Vector modeling of mobile apps and applications. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 599–606.

Maia, W., Silva, F., and Silva, T. (2020). Um estudo sobre a relação entre smartphones e dados demográficos. In *Anais do IV Workshop de Computação Urbana*, pages 302–315, Porto Alegre, RS, Brasil. SBC.

Matters, . (2021). Google Play Statistics and Trends 2021| 42matters. *42matters AG*. [Online; accessed 7. Jan. 2021].

Medeiros, H. (2019). Faturamento com smartphones cresce 6% no Brasil e alcança R$ 58 bilhões em 2018 - Mobile Time. [Online; accessed 25. Fev. 2021].

Pan, W., Aharony, N., and Pentland, A. S. (2011). Composite social network for predicting mobile apps installation. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pages 821–827. AAAI Press.

Peng, M., Zeng, G., Sun, Z., Huang, J., Wang, H., and Tian, G. (2018). Personalized app recommendation based on app permissions. *World Wide Web*, 21(1):89–104.

Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, page 399–408, New York, NY, USA. Association for Computing Machinery.

Ross, S. M. (2014). *Introduction to probability models*. Academic press.

Sabanoglu, T. (2021). Global retail e-commerce market size 2014-2023 | Statista.

Sarwar, B. M., Karypis, G., Konstan, J. A., Riedl, J., et al. (2001). Item-based collaborative filtering recommendation algorithms. *Www*, 1:285–295.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.

Souza, R., Santos, L., Silva, M., Silva, F., and Silva, T. (2020). Impacto do uso de informações demográficas para a recomendação de aplicativos móveis. In *Anais do XII Simpósio Brasileiro de Computação Ubíqua e Pervasiva*, pages 111–120, Porto Alegre, RS, Brasil. SBC.

Souza, R. P., dos Santos, L. J., Coimbra, G. T., Silva, F. A., and Silva, T. R. (2021). A big data-driven hybrid solution to the indoor-outdoor detection problem. *Big Data Research*, 24:100194.

Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009.

Tan, P., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*. Always learning. Pearson Addison Wesley.

Thorat, P. B., Goudar, R., and Barve, S. (2015). Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, 110(4):31–36.

Xu, X., Dutta, K., Datta, A., and Ge, C. (2018). Identifying functional aspects from user reviews for functionality-based mobile app recommendation. *Journal of the Association for Information Science and Technology*, 69(2):242–255.

Yin, H., Chen, L., Wang, W., Du, X., Nguyen, Q. V. H., and Zhou, X. (2017). Mobi-sage: A sparse additive generative model for mobile app recommendation. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 75–78.

Apêndice A

Questionário - Motivação de Instalação de Aplicativos Móveis

Esta pesquisa visa entender os motivos que levam as pessoas a instalarem aplicativos de dispositivos móveis. As respostas são anônimas e serão utilizadas apenas para fins acadêmicos.

1. Qual a sua idade?
2. Com qual gênero você se identifica?

   ☐ Feminino ☐ Masculino ☐ Prefiro não dizer ☐ Outro...

3. Qual sistema operacional você usa?

   ☐ Android ☐ iOS

4. Você instala aplicativos recomendados por redes sociais?

   ☐ Sim ☐ Não

   (a) Se sim, qual o seu nível de satisfação com a recomendação por redes sociais?

	1	2	3	4	5	
Pouco satisfeito	☐	☐	☐	☐	☐	Muito satisfeito

   (a) Quantos aplicativos, em média, você instalou a partir de redes sociais nos últimos 6 meses?

   ☐ Nenhum ☐ 1-5 ☐ 6-10 ☐ 11-15 ☐ Mais de 15

5. Você instala aplicativos recomendados por lojas de aplicativos?

   ☐ Sim ☐ Não

	1	2	3	4	5	
Pouco satisfeito	☐	☐	☐	☐	☐	Muito satisfeito

(a) Se sim, qual o seu nível de satisfação com a recomendação por lojas de aplicativos?

(a) Quantos aplicativos, em média, você instalou a partir de lojas de aplicativos nos últimos 6 meses?

☐ Nenhum ☐ 1-5 ☐ 6-10 ☐ 11-15 ☐ Mais de 15

6. Você instala aplicativos recomendados por pessoas próximas?

☐ Sim ☐ Não

(a) Se sim, qual o seu nível de satisfação com a recomendação por pessoas próximas?

	1	2	3	4	5	
Pouco satisfeito	☐	☐	☐	☐	☐	Muito satisfeito

(a) Quantos aplicativos, em média, você instalou a partir de pessoas próximas nos últimos 6 meses?

☐ Nenhum ☐ 1-5 ☐ 6-10 ☐ 11-15 ☐ Mais de 15

7. Quais outros motivos você normalmente utiliza para instalação de aplicativos?
   ☐ Recomendação de usuários (pessoas não próximas)
   ☐ Comentários de usuários do aplicativo em questão
   ☐ Notas do aplicativo
   ☐ Anúncios em outras mídias
   ☐ Descrição do conteúdo em outras mídias
   ☐ Outros...

8. Quais fatores você considera fundamental para a instalação de aplicativos móveis?

   ☐ Segurança
   ☐ Reputação
   ☐ Finalidade
   ☐ Familiaridade
   ☐ Confiança na loja ou no desenvolvedor
   ☐ Benefícios percebidos

☐ Privacidade
☐ Outros...

9. Você estaria disposto a instalar um aplicativo mesmo que este representasse algum risco segundo seus critérios?

☐ Sim ☐ Não