Journal of Internet Services and Applications, 2022, 12:1 doi: 10.5753/jisa.2022.2379

Investigating the impact of demographic and device information in the recommendation of mobile applications

Raissa P. P. M. Souza [**Federal University of Minas Gerais** | ***raissa.papini@dcc.ufmg.br***]
Leonardo J. A. S. Figueiredo [**Federal University of Minas Gerais** | ***leonardo-figueiredo@ufmg.br***]
Mateus P. Silva [**Federal University of Viçosa** | ***mateus.p.silva@ufv.br***]
Fabrício A. Silva [**Federal University of Viçosa** | ***fabricio.asilva@ufv.br***]
Thais R. M. B. Silva [**Federal University of Viçosa** | ***thais.braga@ufv.br***]
Antonio A. F. Loureiro [**Federal University of Minas Gerais** | ***loureiro@dcc.ufmg.br***]

Received 30 November 2021 • **Accepted** 28 June 2022 • **Published** 29 september 2022

Abstract
The number of people with access to mobile devices, as well as applications to these devices (i.e., apps), has been increasing significantly. Thus, users have to choose among a large number of apps proposing to do the same functions, those that better serve them. A possible solution to this problem is the adoption of recommendation systems. Meanwhile, usually these systems consider only users' preferences to create a profile or request sensitive data (e.g., call and message logs). This work investigates the impact of using demographic and device information on app recommendation by using only easy-to-obtain data to enrich a user profile. We evaluate two approaches: a similarity-based Collaborative Filtering with a limited number of apps and a topic-based approach (i.e., LDA) with wider large-scale data. We also inspected the results under both apps and categories context. The general results reveal that the enriched data provides a better app recommendation with the addition of information about the user's region mean wage achieving up to 210% (or 12 percentage points) of improvement in terms of recall.

Keywords: Mobile Applications, Recommendation System, Demographic Information

1 Introduction

Currently, there has been a significant increase in the number of people with access to mobile devices (GSMA, 2020). Their use is no longer limited to basic features such as sending and receiving calls and SMS messages. With the advent of ubiquitous and pervasive computing, mobile devices have been helping people from complex tasks like banking services to routine activities such as setting alarms. As a result, the number of mobile applications (apps) has grown in such a way that users must choose among those that best serve them. However, this choice is not simple given the large number of apps, with the Google Play Store reaching over 3.4 million apps available (Matters, 2021).

To mitigate this issue, a mobile app recommendation system can make it easier for users to find the desired app. This way, given that companies are developing more and more apps with the same purpose, the user will no longer have to install several of them until finding the one that best suits him. Likewise, app development companies can save resources by targeting marketing campaigns to users who have more chances to install their apps.

There are several ways to recommend apps to users. A simple recommendation could use only data related to the applications themselves to recommend to the user those that have a higher acceptance rate in terms of popularity. On the other hand, other recommendation systems take into account data referring to each user, aggregating both easy-to-obtain (e.g., approximate location of the user) and more sensitive and difficult do obtain information (e.g., call logs). This information can be obtained through permissions granted by each user and can include several sensors built into the devices, such as GPS, accelerometer, WiFi, among others. It is important to note that users may not allow this data collection due to privacy concerns.

However, a common problem with these approaches is that the added data can contain information such as: privacy preferences, usage history, call logs, and others that are not easy to obtain. Therefore, such approaches can lead to a lack of knowledge about users, making it difficult to be adopted in a large-scale scenario. In addition, we argue that choosing an app may also be related to demographic and device factors, as they may be related to the users' interest and not all apps will be available for any region or device. Despite this, we have not found solutions that use demographic information about users or their devices.

In this context, the hypothesis of this work is that the use of demographic and device information favors the recommendation of mobile applications. Based on this, the goal of this research is to validate this hypothesis. To do this, we create a user profile that is composed of easy-to-obtain data, which includes demographic information of the user's region of residence and mobile device. First, we investigate how this profile would benefit the app recommendation considering the Collaborative Filtering strategy. The results were promising, and then we used LDA (Latent Dirichlet Allocation) to investigate further their benefits. Overall, the results reveal that demographics and device information are important sources of information to improve the recommendation of apps.

The main contributions of this work are:

- We create a user profile based on demographics and devices, using only easy-to-obtain data.
- We investigate how this profile leverage the recommen-

Table 1. Related Work

Work	Collaborative Filtering	Other Techniques	Details
Frey *et al*. (2017)		X	LDA and Random Forest
Cheng *et al*. (2018)		X	LDA, MTM, and Multinomial distribution
Liang *et al*. (2020)		X	Factorization Machine
Pan *et al*. (2011)		X	Graphs
Xu *et al*. (2018)		X	Graphs and *PageRank*
Ma *et al*. (2016)		X	*Word2Vec* reformulation
Yin *et al*. (2017)	X	X	*Bag of Words* and Latent Topics
Zhu *et al*. (2021)	X	X	Collaborative Filtering and Voronoi Diagram
Peng *et al*. (2018)	X		Factorization Matrix
Liu *et al*. (2015)	X		Latent Factors
Liu *et al*. (2016)	X		Latent Factors

dation of mobile apps considering the Collaborative Filtering strategy.

- We investigate how this profile leverage the recommendation of mobile apps considering the LDA (Latent Dirichlet Allocation) strategy.
- We demonstrate that demographic information, specially wage, as well as the user's device price, are good sources of information to improve the recommendation of apps.

This work is organized as follows. In Section 2, we present the problem statement and the main studies in the literature that deal with mobile app recommendation. Next, in Section 3 we describe the preliminary results using the collaborative filtering strategy. A more complete study is presented in Section 4, where the LDA strategy is used. We discuss the limitations of this work in Section 5. Finally, we conclude our work and present future directions in Section 6.

2 Fundamentals

2.1 Problem Statement

Identifying the best apps to recommend to a user requires analyzing information about the apps and the users themselves. Although traditional recommendation systems usually obtain good results, recommending mobile applications still shows some difficulties. The first one is the vast quantity of items that can be presented to a user since there are many apps built for the same purpose, and this number is increasing. Besides, unlike recommending movies, which is interesting, for example, recommending items similar to others the users have watched, it is unlikely that a user will install an app with the same purpose as another one already installed.

This peculiarity leads good mobile app recommenders to look deeper at the user's interests, searching for what makes a user install an app. Thus, recommending mobile applications may not rely only on the history of the apps installed. The recommendation systems on *Google Play Store* and *Apple App Store*, for example, also suggest popular apps to their users in addition to the ones related to the user's profile.

So, we can initially describe a user $u \in U$ as a sequence of their installed apps $u = \langle App_1, \ldots, App_N \rangle$ and use this sequence to learn patterns and recommend similar applications. But, as we previously discussed, it is meaningful to consider the user's interests and enrich the information tuple with its attributes. Thus, we can represent a user with a tuple in the form $u = \langle App_1, \ldots, App_N, Attr_1, ..., Attr_K \rangle$, where $Attr_K$ represents the K'th user attribute.

It is also important to acknowledge the kind of attribute used in terms of their privacy and sensitive data. Although some recommenders use information about the items' usage and personal data like call logs, accurate locations, and wages to help improve their results, the user profile built had to be dealt with caution. The use of sensitive data can lead to unwanted bias, difficulties collecting data, not to mention an ethical discussion about invading users' privacy. In this context, we can use approximate data, such as locations and demographic information, since it is not accurate enough to identify a user and preserve sensitive details.

Nevertheless, all data-driven system is bound to local privacy laws, such as Brazilian General Data Protection Law (LGPD). Usually meant to save users' privacy, some laws require the user to approve the collection of their personal information as well as manage it (e.g., remove collected data). Besides, it is necessary to inform the user about why their data is collected and for what purpose, so they can decide to share information or not.

2.2 Related Work

In this section, we present the main studies found in the literature that involve the recommendation of mobile applications.

Many works were developed with the goal of recommending mobile apps, giving rise to different approaches and strategies. To better recommend applications to users, the works of authors Frey *et al*. (2017) and Cheng *et al*. (2018) use LDA (Latent Dirichlet Allocation). The first work uses LDA to select the main topics among the application descriptions, using the probability that a user likes each topic as input for a model based on the Random Forest algorithm. The authors of the second work, on the other hand, use the order of installation of applications to observe three aspects, namely: short-term contexts, where the probability of a user installing an application is estimated, given other applications they have; co-installation patterns, where it was analyzed which apps users normally install together, applying

LDA; and random installs, where the recommender indicates popular apps with a high chance of being accepted.

Two other works use graphs to represent associations between users. In the work of Pan *et al.* (2011), information from smartphone sensors is used to build a graph representing connections between users, enabling the calculation of the potential for installing an application based on the neighbors of a given user. In this case, it is necessary to gain access to data often blocked by general users, making it difficult to be adopted on a large scale. The work prepared by Xu *et al.* (2018) considers the functionality of each application, making it possible to predict the user's next needs through a graph of co-occurrence.

The work developed by Ma *et al.* (2016) proposes a change to the Word2Vec (Mikolov *et al.*, 2013) algorithm, aiming to predict the installation of applications based on the recent use of others. The work by Yin *et al.* (2017) uses users' preferences to recommend applications through the description and permissions of each application. For this, the authors used Latent Topics to characterize a user's interests and relate them to permission preferences for each application category. The work of Zhu *et al.* (2021) also uses app usage information as well as contextual user mobility data. The authors associate users across dynamic geographic areas, so applications are recommended based on neighboring users in the same region.

Other works take users' privacy preferences into account. The work of Peng *et al.* (2018) identifies applications that require a lot of permissions and are ranked lower, giving them low priority. The authors matched both the apps which pass this filter and the user interests through a factoring matrix. Liang *et al.* (2020)'s work also uses privacy information as well as apps' description and popularity to predict the rating a user would give a determined app. For this, the authors used a factorization machine with additional views to deal with each type of feature. Likewise, the work of Liu *et al.* (2015) also lists the user's privacy and behavior preferences. However, these factors are related to a latent profile. Later, the authors modified this work by adopting a new strategy, where the category and functionality of each application are analyzed (Liu *et al.*, 2016). Again, they used latent vectors, this time relating the functionality of each application to the user's interests through its installation list.

Table 1 presents a comparison between the described studies. To improve app recommendations, other authors may use real app installation sequences (Cheng *et al.*, 2018; Pan *et al.*, 2011) or apps detailed usage (Ma *et al.*, 2016; Zhu *et al.*, 2021), which need multiple collections over time; and call logs and Bluetooth proximity logs (Pan *et al.*, 2011), which users may not feel comfortable granting access to due to privacy concerns. The usage of this kind of data usually leads to fewer users or difficulty applying the approach on a large-scale scenario. The availability of data may also be impacted since data from self-built apps and questionnaires (Frey *et al.*, 2017; Pan *et al.*, 2011), or crawlers of reviews, downloads, and permissions (Xu *et al.*, 2018; Yin *et al.*, 2017; Liang *et al.*, 2020; Liu *et al.*, 2015, 2016) is hard to collect large-scale data.

One of the significant differences of our work in comparison to the existing ones lies in using easier-to-obtain and less invasive data concerning user privacy. As previously discussed, the existing works use call logs and detailed app usage records, among other privacy-invasive information. Differently, our proposal requires only the applications installed by the user, one approximate location, and information on the users' mobile devices. This way, our approach does not require information about when, how and for how long the user used the apps. Besides, we need only an approximate location, which already leads us to the user's city and the general region characteristics, and not actual locations of the user's visited places.

Such data, easier to be obtained, can provide a series of other information through data enrichment. We did such enrichment through the use of location to incorporate contextual, demographic information about the user. Furthermore, the device used can lead us to know its price (Maia *et al.*, 2020), and therefore which category of such device (e.g., entry, premium, among others). We chose Collaborative Filter as it is a widely used approach when it comes to recommendation systems, and we used it as a baseline during the preliminary study. We selected the LDA approach because it has good results in app recommendation, as shown by Cheng *et al.* (2018).

3 Collaborative Filtering Analysis

In this section, we describe the preliminary analysis performed in order to measure the demographic data usage effectiveness in the recommendation of apps.

3.1 The Data

In this investigation, we used a real-world dataset collected from June 4 to August 8, 2019, that contains Android users' information provided by a private company under a non-disclosure agreement. The data collection was made every day at midnight, gathering all the apps installed at the moment, the user's device, the date, and an approximate location. To guarantee the coherence of the results, we removed data from users who were not present during the entire period in the collection or joined after its start, ending up with 7,406 users.

The dataset has one record per user per day, containing the user's current installed mobile applications. So, we could represent the Problem Statement's user tuple as $u = \langle date, [installed\_apps] \rangle$. However, as we aimed to perform only experimental research at this section, we selected only 1% of the most installed apps by all users based on exploratory analysis. This led to 249 most popular apps in the dataset and the tuple $u = \langle App_1, App_2, ..., App_{249} \rangle$. We chose these apps to facilitate a primer investigation of the impact of demographic information, since the dataset contained many apps with very few installations.

We also individually analyzed and classified these apps into categories according to their functions and descriptions. After analyzing each app, we ended up with 28 categories that separate all of them according to their purposes. That is because the categories provided by the Play Store depend directly on the person who published the app, which shows inconsistencies. For example, in the list, there are basketball games that are classified as sports and not as games.

To create the models, we used only the first day (June 4) for training, assigning the other days' data for testing. We did that based on a stochastic assumption in which the probability of a future event is based only on a previous one. So, we assumed that the users decide to install an app based on the current apps installed. Note that once we used only one day for training, the user tuple does not need to include the date.

This approach is a standard one in Collaborative Filtering that usually uses only a snap of explicit or implicit user feedback, such as ratings, clicks, and purchases (Sarwar *et al.*, 2001). Also, this kind of feedback usually is a one-time rating or one-time purchase that we would provide to the recommender as training while other items' future feedback remain for testing. In our scenario, we use the installation record as implicit feedback that the user liked an app.

Figure 1 shows the number of users that have at least one app in each category, both in test and training sets. It is possible to notice that the number of users increased in all categories during the observed period, showing the installation of these by other users.

The dataset also contains each user's home approximate location. This information was extracted from the location history of the user, which we did not have access to because of privacy concerns. We enriched the users' home location with the census tract information provided by Brazilian Institute of Geography and Statistics (IBGE) (IBGE, 2021). As some data in the census are related to specific region traits (e.g., is the region part of Legal Amazonia?) and are not related to app installation decisions, we chose only 24 of them related to demographics, and therefore associated with social life, income, age, education, gender, and race, as shown in Table 2.

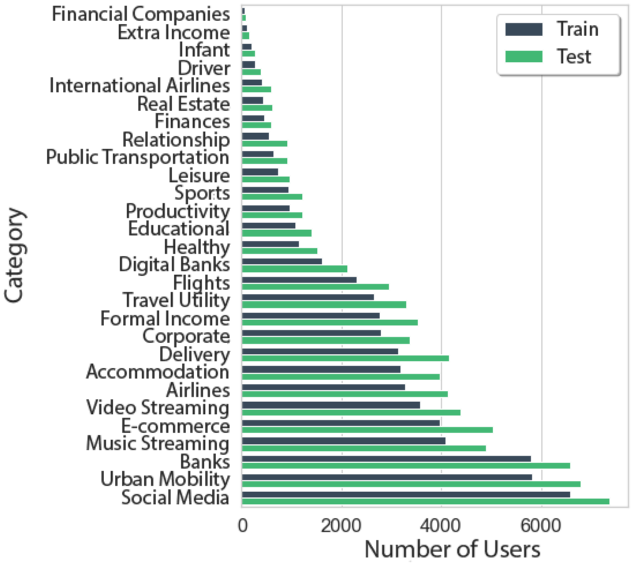


Figure 1. Number of users per category in training and test datasets.

We made this decision based on the Brazilian socioeconomic situation, where the number of inhabitants in a region and its wealth leads to the presence or absence of apps in categories such as delivery, public transportation, and airline companies. For example, a region with a low number of inhabitants usually does not have access to food delivery since the companies do not benefit enough from this public. Also, users from wealthier regions will have the opportunity to fly and so use this kind of app. Instead, users from deprived regions will need apps for extra income or only use basic apps.

Table 2. Demographic aspects used on model construction.

Demography	Category
Number of residences	Social
Number of residents	Social
Average number of residents per residence	Social
Number of women	Gender
Number of literate women	Education
Number of men	Gender
Number of literate men	Education
Number of white people	Ethnic Group
Number of black people	Ethnic Group
Number of brown people	Ethnic Group
Number of yellow people	Ethnic Group
Number of indigenous people	Ethnic Group
Number of people with Age <= 10	Age
Number of people with 11 <= Age <=20	Age
Number of people with 21 <= Age <= 30	Age
Number of people with 31 <= Age <= 40	Age
Number of people with 41 <= Age <= 50	Age
Number of people with 51 <= Age <= 60	Age
Number of people with 61 <= Age <= 70	Age
Number of people with 71 <= Age <= 100	Age
Mean of average wage per resident with or without income	Wage
Mean of average wage per resident with income	Wage
Mean of average wage per responsible resident with income	Wage
Mean of average wage per responsible resident with or without income	Wage

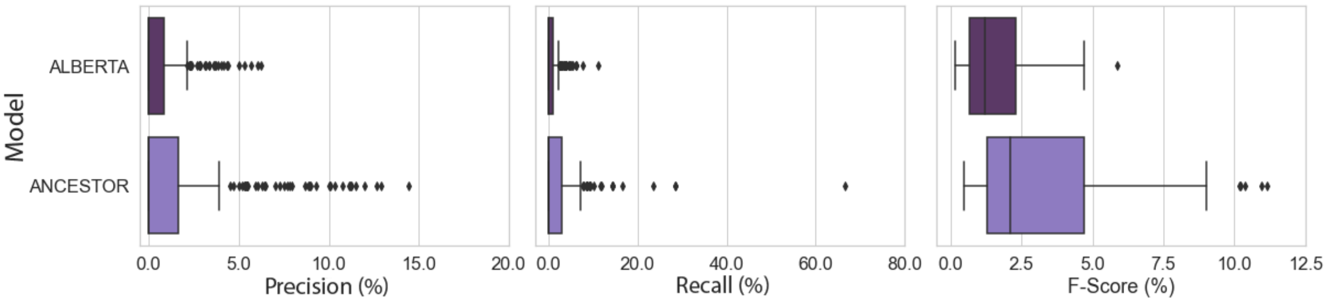


Figure 2. Precision, recall and f-score results.

So, to select the 24 traits, we semantically analyzed each one and looked for their socioeconomic impacts.

So, our final dataset contained information about users $u_{enriched} \in U_{enriched}$ in the form of the tuple $u_{enriched} = \langle App_1, App_2, \ldots, App_{249}, CT_1, CT_2, \ldots, CT_{24} \rangle$, where CT_i is the i'th census tract for that user.

Please note that our motivation for using this data is to preserve user privacy. We only need the user's list of installed apps and a single approximate location of their city to generate recommendations. Other data could be used, such as the applications in use, but would require several collections during the day, invading the user's privacy. In addition, as mentioned before, any data-oriented solution is subject to current legislation, and the user must be aware and authorized.

3.2 Models

In this section, we discuss the details of the proposed solution called ANCESTOR (Application aNd CEnsus baSed recommendaTion algORithm) and the baseline, which we called ALBERTA (AppLication BasEd RecommendaTion Algorithm). ALBERTA is a traditional collaborative filtering approach for the recommendation problem, which considers only the user's installed apps. On the other hand, the ANCESTOR uses the 24 census tracts' info as additional columns on the dataset we collected besides the app's installations. In this way, the hypothesis that demographic data leads to a better app recommendation could be validated or refuted.

For both ANCESTOR and ALBERTA, we use the Memory-based Item-Item approach. We could have also used a User-Item algorithm, but as this method does not present good scalability as the number of users grows (Sarwar *et al.*, 2001), we decided not to use it. On the Item-Item approach, we analyze the similarity mainly between the items themselves, creating a *Items*×*Items* symmetric matrix. As the number of items usually does not grow so fast as the users, Item-Item collaborative filtering can achieve the same quality with lower computational cost and better scalability than the User-Item approach.

In the baseline solution (ALBERTA), each item represents an app. A value in the *User*×*App* matrix is 0 if that user does not have the respective app installed and, 1 otherwise. The collaborative filtering approach is then applied with the objective of finding the most appropriate apps to recommend to each user.

The proposed model, ANCESTOR, uses the apps installed and the 24 census tracts' info of the user (Table 2). We also normalized the census data in the range between 0 and 1 to maintain the installation database's coherence. We have created on the original database a column for each census info. This strategy can lead to the identification of apps related to some demographic contexts and the analyzes of similarities between an app and a demographic region. The collaborative filtering is then applied, now considering not only the apps as items, but the demographic tracts as well.

3.3 Results

We computed the traditional metrics: precision, recall, and f-score. The values for each metric are calculated for each application individually. Figure 2 shows the distribution of precision, recall, and f-score considering all applications. We can see that ANCESTOR, in general, stands out from the ALBERTA model. These results show that the addition of demographic information contributed to increase the accuracy of the prediction of installations of most apps, both from the point of view of the correctness of the recommended applications and from the point of view of the applications that the user actually installed. We observed that the maximum precision achieved by the ANCESTOR model is almost twice the ALBERTA's one.

Although precision is usually low for recommendation models in general, we can draw some conclusions. First, by adding user demographics, the precision has increased for most apps. This gain shows that the similarity of users in terms of demographic characteristics is a relevant factor.

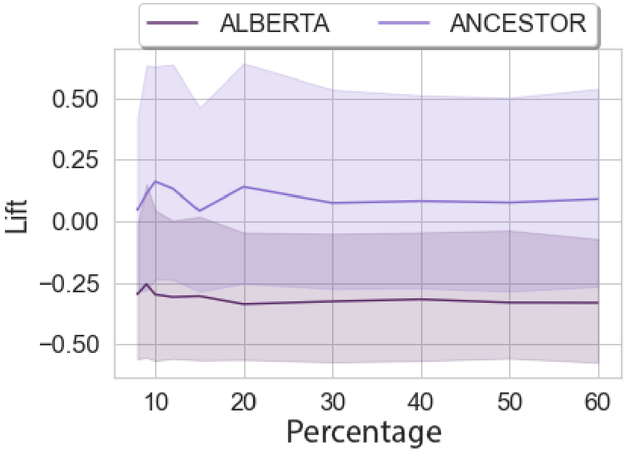


Figure 3. Average lift for all sample sizes, with confidence interval of 95%.

Another observation is the conversion rate or returns when a marketing campaign is designed to attract new users to a particular app. It may be the case that, even with low precision, the rate of increase (i.e., Lift) when targeting users recommended by the model is more numerous than when using random samples of users. In other words, the return on investment per user achieved will be greater. Figure 3 presents the results of this analysis. It is possible to see that the Lift does not have great variation for changes in sample size (x-axis), in addition to the fact that ANCESTOR has, in general, the best results. Furthermore, we see that even with the confidence interval, ANCESTOR always has a higher draw rate than ALBERTA.

In general, these results show that when using the ANCESTOR recommendation model, the rate of return tends to be more effective than when using random samples of users as targets. In other words, the chance that users recommended by the ANCESTOR model will install an application after a targeted marketing campaign is higher than when random users are selected.

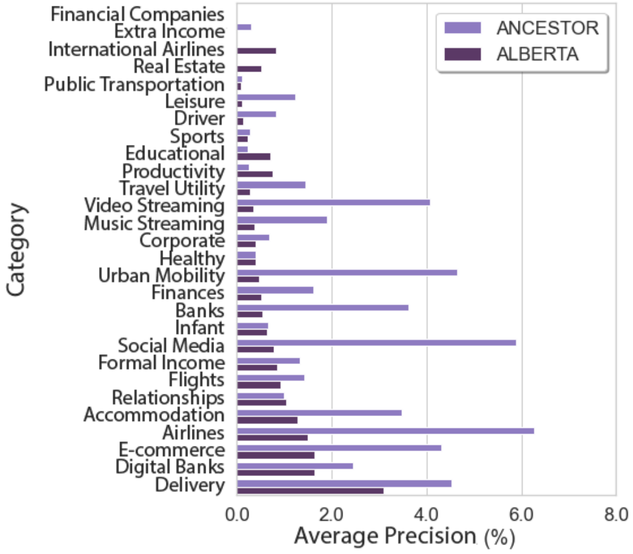


Figure 4. Average precision per category.

To better visualize the behavior of models, we calculated the average precision for each app category. With that, we hope to be successful in identifying why the accuracy of some apps has increased relatively with the addition of demographic data. In Figure 4, we see that most of the categories with higher average precision may be associated with demographic factors, such as urban mobility and delivery applications, which are more present in cities or places more populated. Apps related to airlines, streaming of video or music, hosting, and e-commerce may depend on other factors, such as wages, since spending money on travel, streaming, and e-commerce require the person to have sufficient income to do that. There are also categories of apps (e.g., infants and relationships) that are not as dependent on demographic data of the location and, therefore, ANCESTOR does not show a significant improvement over ALBERTA.

Regarding recall, we see that ANCESTOR also stands out when compared to ALBERTA, as we see in Figure 5. Here, we also observe some categories such as urban mobility, physical banks, and social networks, which also presented good accuracy results. On the other hand, other apps categories stood out only in the recall, such as supplementary income and driver apps, both of which can be related to factors such as wage. The fact that ANCESTOR achieved better results in recall when compared to precision can perhaps be explained by the number of users who installed apps from these categories. If few users installed an app, it is more likely that them are included in the recommendation set.

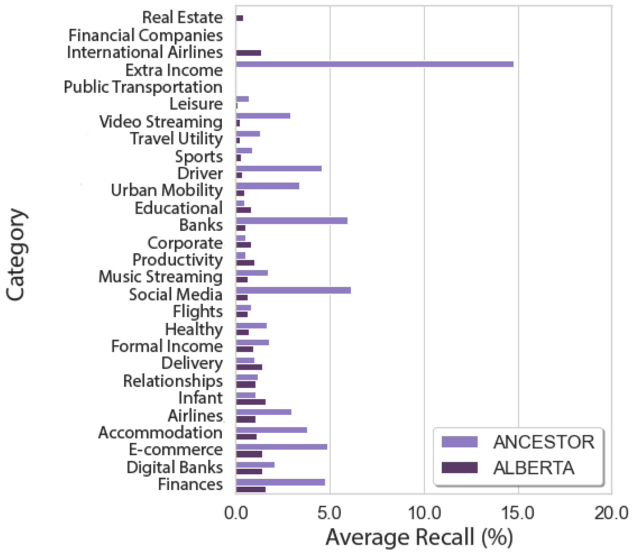


Figure 5. Average recall per category.

3.4 Conclusions

This section introduced a solution for recommending apps to users based on their installed apps and demographic information. The results show that the proposed solution obtained good results in terms of precision, recall, and lift metrics. Thus, it reinforces the assumption that demographics are relevant in helping to recommend apps installation. In addition, the fact that the solution does not require historical data (i.e., applications and users) makes the construction of the model simpler.

In the next section, we make a more advanced analysis, with a larger number of users and the adoption of the LDA technique, aiming to achieve more concrete results.

4 LDA Analysis

Given the promising results in Section 3, we performed further analyses about the demographic data usage. Besides, we measured the impact of the user's device price.

4.1 The Data

In this analysis, we used a similar yet larger dataset than the one described in Section 3, also collected by the same private company. We also used the information of the user's device collected by the partner company. So, our dataset was composed of users $u \in U$ in the form of a tuple $u = \langle date, location, device, [installed\_apps] \rangle$. The data initially had 18,560 distinct apps from 87,903 different users' records collected from November 1, 2019, to January 31, 2020. We maintained only users who had installed at least seven apps over a period greater than three days. We also removed apps that had less than four installations over the observed period. We chose these values based on the first quartile threshold.

As we also evaluated the model performance based on apps categories, we used the ones provided by the Google Play Store. We chose to do that since it was impracticable classifying all apps manually. We also used the Z-score approach to identify potential collection errors. Thus, the final database had information about 14,660 users and 13,329 apps grouped into 48 categories.

Besides that, as mentioned before, the collected data has only one user's record per day, so we could not know the exact order of installation if more than one app was installed on the same day. So, we built a potential installation sequence and the final user record can be expressed as $u = \langle device, location, [apps\_sequence], [apps\_categories] \rangle$, where *device* is the last device used on installations, *location* represents the approximate location of user's home, and *apps_categories* lists the categories of all apps on *[apps_sequence]*, following the same order and with repetitions.

4.1.1 Device and Demographic Data

We have also semantically analyzed each demographic information used in the previous study (see Section 3) to bring up only the ones that seem to have stronger associations with mobile recommendation. So, we assumed only data relative to the wage and populational size of a city; in addition, we also consider in this study the user's device price.

- **Wage:** We use the IBGE census tracts' mean wage per person to categorize the user's wage according to its home region. As the last census tract took place in 2010, we took the Brazilian minimum wage (In Brazilian Reais) at that time to create the categories below:
  - **Lower:** Less than half minimum wage (Wage < R$255);
  - **Intermediate Lower:** From half to one minimum wage (R$255 ≤ Wage < R$510);
  - **Intermediate:** From one to two minimum wages (R$510 ≤ Wage < R$1.020);
  - **Intermediate High:** From two to four minimum wages (R$1.020 ≤ Wage < R$2.040);
  - **High:** More than four minimum wages (Wage ≥ R$2.040).
- **Populational Size:** Similarly to the wage data, we have identified the user's city and used its population size to classify it into Small, Intermediate, and Big cities using Ipea (2008). To do that, we adopt the following thresholds:
  - **Small Town:** Cities in which Population Size < 100.000;
  - **Intermediate City:** Cities with $100.000 \leq$ Population < 500.000;
  - **Big City:** Population Size ≥ 500.000.
- **Device Price:** Unlike the demographic data above, we got the device's price using a web crawler developed by Maia *et al.* (2020). With this information, we expect to be able to identify patterns related to the user's consumption and indirectly wage, as well as restrictions caused by the device configuration. We have also classified the price based on Medeiros (2019), as shown below:
  - **Basic:** Devices with Price $< R\$700$;
  - **Intermediate:** Devices on the range $R\$700 \leq$ Price $< R\$1.000$;
  - ***Mid-high***: Devices with $R\$1.000 \leq$ Price $< R\$2.000$;
  - ***High-end***: $R\$2.000 \leq$ Price $< R\$3.000$;
  - ***Premium***: Devices with Price $\geq R\$3.000$.

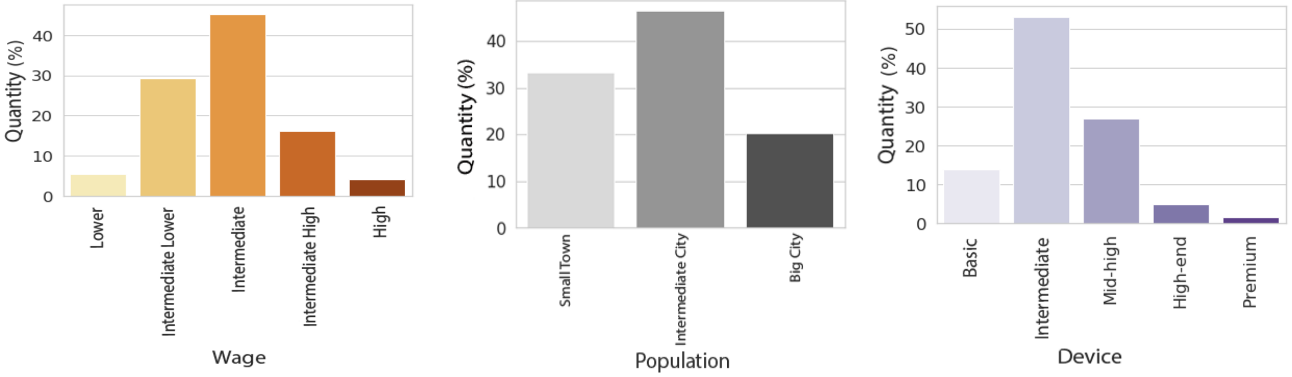
So, after the enrichment, the final users dataset $u_{enriched} \in U_{enriched}$ was represented by

$$u_{enriched} = \langle wage\_range, pop\_range, device\_price\_range, [apps\_sequence], [apps\_categories] \rangle$$

Figures 6a, 6b and 6c show the wage, population and device distributions, respectively.

4.2 Metrics

In this study, we took different methods from the previous one described in Section 3. First, we established that the last



(a) Wage **(b)** Population **(c)** Device price

Figure 6. Device and demographic data distribution.

s applications installed would be used as a test set, with $s \in S = \{1, 3, 5\}$. In this way, we have more data to train the model since we used all other apps not included in s in the training dataset. In addition, we consider $n \in N = \{5, 10\}$ as the size of the recommendation set. In other words, the model will recommend n applications for the user, and we will compare this set of recommendations to the ground-truth (i.e., test dataset).

We have also adapted the metrics specifications accordingly. Thus, with U being the set of all users, we assume A_u^* as the set of all applications installed by the user u, with A_u^s being the set of all s last installed applications by that user, and R_u^n as the set of n apps recommended for the same u user as the model output. Taking this into account, we will obtain precision and recall values as a function of the variables s and n. Therefore, we have:

- **Precision:** to achieve 100% of precision when all n recommended apps are installed (i.e., are in the s last installed apps), precision should only be calculated for values of $s = n$. It is also worth mentioning that only for this metric, we used $N = \{1, 3, 5\}$ so that it would be coherent with the set S. Therefore, the precision will define the percentage of the total correct recommendations, analyzing recommendations of size $n = s$ against the total installed applications.

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

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

- **Recall:** represents the percentage of the s apps installed by the user that was in the recommended set.

$$Recall_{s,n} = \frac{1}{|U|} \sum_{u \in U} \frac{|A_u^s \cap R_u^n|}{s} \tag{2}$$

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

As the precision and recall metrics needed to be calculated for several different recommendations, we could not calculate the F-Score metric.

As mentioned before, to calculate the precision, it is necessary to have $n = s$. We can assume that it predicts the following s user installations. On the other hand, as it is possible to recommend more applications than those present in the test set and still obtain 100% recall, we can infer that this metric represents better the performance of the solutions compared to what is expected in a real-life recommendation system.

In addition to the app-related metrics raised above, we also evaluate the models considering the categories of the recommended applications. Therefore, we calculated the Precision and Recall metrics for recommendations from the apps and their apps' categories. It is noteworthy that we use the applications recommended for each user to calculate the category-related metrics.

To obtain such categories, we replace each app recommended for each user with its respective category. This way, we can recommend an app category more than once to the same user. Since the same category could be recommended more than once, it was necessary to adapt the definition of the previous metrics. Therefore, let $R_{u,cat}^n$ be the multiset of the n categories recommended to the user u, and $A_{u,cat}^s$ the multiset with all categories of the last s applications installed by that user u. So, we define:

$$R_{u,cat}^n = \left\{ cat_1^{m_r(cat_1)}, \ldots, cat_n^{m_r(cat_n)} \right\} \tag{3}$$

$$A_{u,cat}^s = \left\{ cat_1^{m_a(cat_1)}, \ldots, cat_s^{m_a(cat_s)} \right\} \tag{4}$$

Let $m_r(x)$ and $m_a(x)$ represent the multiplicity functions of the multisets $R_{u,cat}^n$ and $A_{u,cat}^s$, respectively, defined as:

$$m_r(x) = \sum_{c \in R_{u,cat}^n} \nVdash (c = x) \tag{5}$$

$$m_a(x) = \sum_{c \in A_{u,cat}^s} \nVdash (c = x) \tag{6}$$

In other words, the respective multiplicity function indicates the number of times a category appears in the analyzed multiset.

Furthermore, the cardinality of the intersection of the two multisets $A_{u,cat}^s$ and $R_{u,cat}^n$ (also called minimum common divisor) is given by:

$$T_u = \sum_{c \in A_{u,cat}^s} \min(m_a(c), m_r(c)) \tag{7}$$

Based on the equations above, we define the precision and recall of categories as:

- **Category Precision's:** Defined as the percentage of total recommended app categories that have been installed by the user. It is noteworthy that, as well as the Precision metric for applications, we use $N = \{1, 3, 5\}$ to calculate this metric.

$$Precision_{cat\, s,n} = \frac{1}{|U|} \sum_{u \in U} \frac{T_u}{n} \tag{8}$$

where $s \in S$ and $n = s$.

- **Category Recall's:** defined as the percentage of categories that the user u installed and that at least one of its apps was recommended for that user.

$$Recall_{cat\, s,n} = \frac{1}{|U|} \sum_{u \in U} \frac{T_u}{s} \tag{9}$$

where $s \in S$ and $n \in N$.

4.3 *Latent Dirichlet Allocation* (LDA)

Latent Dirichlet Allocation (LDA) is a probabilistic model initially developed for text processing and widely used in the Information Retrieval area. The model assumes that each document is composed of a text that can be described as a set of topics. In turn, each topic has terms that are most frequently associated with it (Blei *et al.*, 2003). For example, in a topic related to arts, it would be very likely to find terms such painting, drawing, pencil, light, and shadow.

A significant model's point is the scope of themes contained in each document, defined by the hyperparameter α.

If the *corpus* of the documents has few themes, the value of α will be smaller. The same idea is applied to β hyperparameter, that provides information about the distribution of words per topic.

Once created, a model should be evaluated on its adequacy to what it was proposed. One way to carry out such an assessment is through the analysis of topic coherence (Röder *et al.*, 2015), which seeks to analyze the terms with the greatest relevance (distribution) in each topic, and verify whether their semantics can be related.

4.3.1 Baseline

To create the baseline LDA model, we used the proposal of Cheng *et al.* (2018), which considers the users as documents and the name of the installed apps as terms. The steps to create a model are also provided by Cheng *et al.* (2018).

We choose the number of topics and the α hyperparameter based on empirical tests through the following steps:

1. Let A be the set of all possible good choices for α.
2. Let Ψ be the set of all possible good choices for the numbers of topics.
3. For each combination of $a \in A$ and $\psi \in \Psi$, repeat η times:
   (a) Create a LDA model in which $\alpha = a$ and $K = \psi$.
   (b) Evaluate the model's topics coherence.
4. Choose the best model.

It is worth mentioning that we created η models with the same hyperparameters because a LDA model is probabilistic, and so will lead to different results. We select the best model in terms of topics coherence, and so, we used the model with seven topics and $\alpha = 31$ generated on the first iteration. The other hyperparameters were set statically, being $\eta = 5$ and $\beta = \frac{1}{K}$. To recommend apps applying this model, we followed the steps described by Cheng *et al.* (2018).

4.3.2 Proposed Solution

In addition to the apps' names, the proposed solution assumes as terms the users' demographics and device price, according to the categories defined in Section 4.1. To analyze the impact of each type of demographic information, we evaluate five solutions, as discussed below. Also, to use the best model possible for each solution, we generate new models for each one following the same steps described previously. The solutions are:

- ***Population***: We add only data about the population size of the user's city.
- ***Wage***: We add only data relative to the average wage of the census tract of the user's home location.
- ***Device***: We add only data about the price range of the user's device.
- ***Demographic***: We add data about the average wage and the population size.
- ***Complete***: We add all external data of the user (i.e., city's population size, average wage, and device's price range).

So, for each solution, we will add the respective data to the user's installation records. In this way, the terms in a document will refer to both apps and demographic/device information. Besides that, with the new models, we also had to fit again the hyperparameters. To do that, we followed the same steps described in Section 4.3.1 and Figure 3 shows the chosen hyperparameters.

Table 3. Hyperparameters of the best models obtained for each solution.

Solution	Hyperparameters	
	Number of Topics	Alpha
Population	8	28
Wage	8	40
Device	8	30
Demographic	7	26
Complete	7	25

Finally, we only had to certify that the models were not recommending any additional information to the users. Thus, we followed the steps below:

1. For each drawn topic:
   (a) Remove the items that represent the added demographic/device information.
   (b) Sort the apps in a descending manner, according to the topics distribution.
2. For each user u, draw its distribution θ_u from the trained model.
3. Repeat the following steps n times to recommend n apps:
   (a) Draw a x topic from the user topics distribution.
   (b) Recommend the best non-recommended topic app that was not installed by the user.

4.4 Results

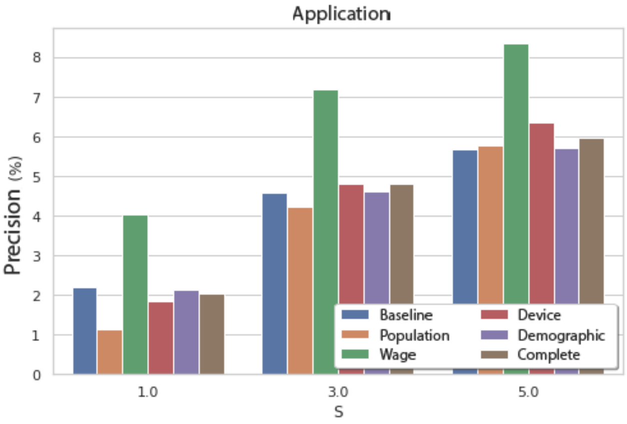
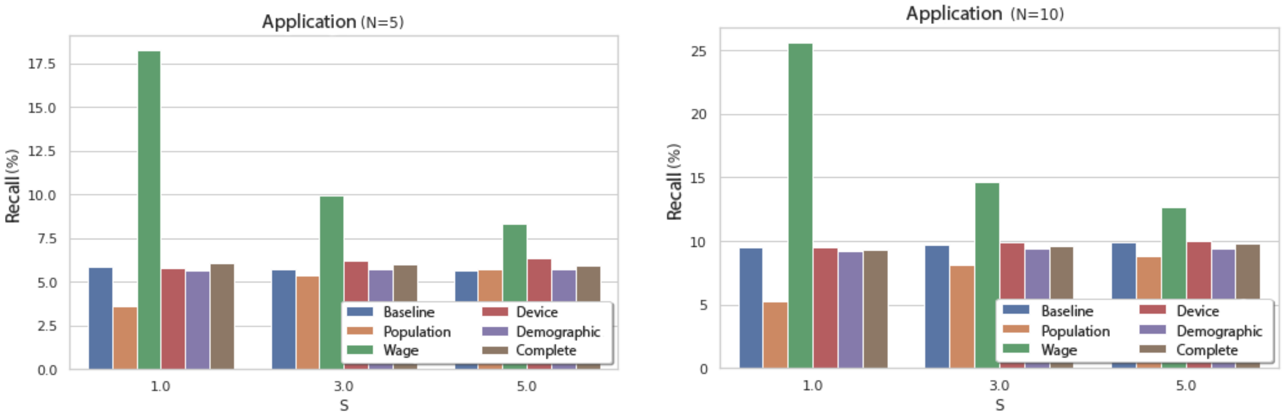


Figure 7. Precision of the LDA models.

As we can see in Figure 7, in general, the results of the proposed solutions perform better regarding precision than the baseline one when analyzing the last 3 and 5 installed applications ($s = 3$ and $s = 5$). In fact, we notice an improvement in results when we increase the value of s. This result



(a) Recall for n=5.

(b) Recall for n=10.

Figure 8. Recall results based on apps context.

is in line with expectations, as increasing the number of recommended applications increases the probability of getting more correct.

The inclusion of the population size of the user's city of residence did not lead to good results, surpassing the baseline only when 5 applications are recommended. The disparity in the percentage of users residing in each type of city (i.e., small, medium, and large cities) may have caused this situation. Thus, with a large number of users in the same category, it is possible to have a great diversity of interests, making it more difficult to find patterns. Furthermore, the information on the size of the city is generalist itself, as it naturally groups a large number of users with different interests and lifestyles.

On the other hand, information such as the type of device and the average wage of the user's census tracts say more about the socio-economic context in which a person is inserted, thus being more specific. As for the device category information, we noticed a slight improvement in recommendations of 3 and 5 apps (4.51% and 11.71%, respectively). This may indicate that the smartphone a user possess is an important aspect of his/her interests. We also observe that adding only wage information has a positive impact (over 84% when $s = 1$), which may indicate a better correlation of the type of application consumed by people from different social classes.

The recall was also affected positively for almost all solutions, as shown in Figures 8a and 8b. However, once again the addition of population size information achieved a slight improvement when $s = 5$ and $n = 5$ (1.44%, seen in Figure 8a). In addition, the incorporation of information referring to wage showed relevant improvements. Thus, knowing the average income of the user's region leads to an improvement of up to approximately 12 percentage points (or 210.22%), if 5 applications are recommended (Figure 8a), and by 16 percentage points (or 167.45%), with the recommendation of 10 apps (Figure 8b).

Regarding categories, most of the proposed solutions were not effective, despite achieving higher levels of precision than the same metric for applications (Figure 9). The only exceptions are the use of income classes, with a slight improvement of 1.92% when $s = 5$, in addition to the use of device embedding, which did better for $s = 5$ (1.77%). We can explain this situation by the fact that the analysis of categories is done based on the conversion of the recommended applications to their respective categories (Section 4.2). Thus, it's possible to recommend more than one application of the same category. Therefore, the recommendation of more than one app from wrong categories would have a larger negative impact than the base solution.

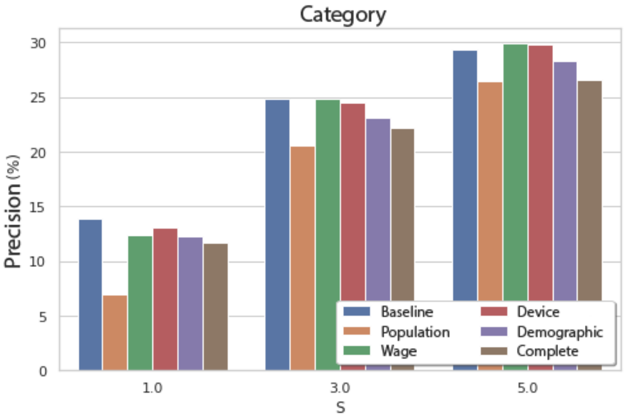
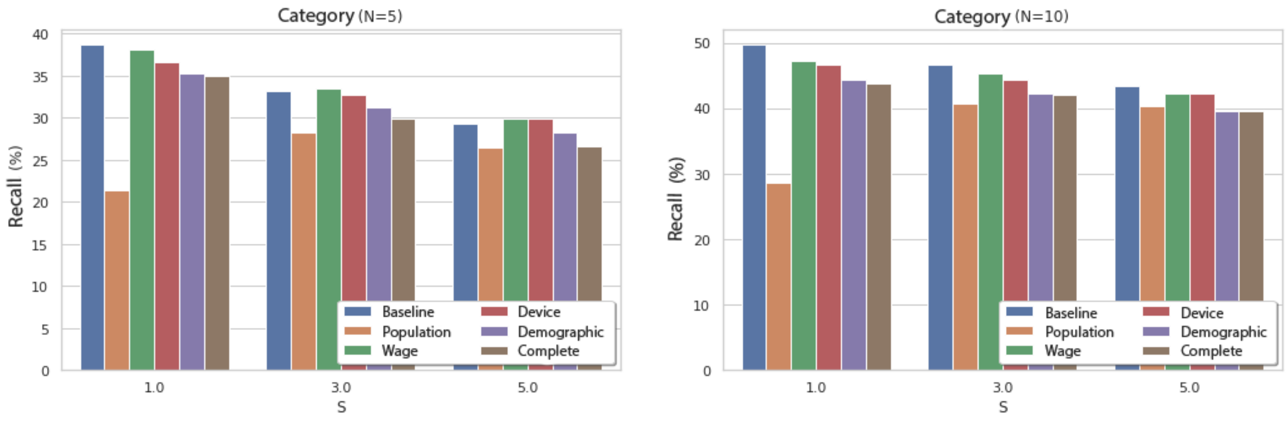


Figure 9. Precision results when the categories are considered.

Finally, the analysis of categories in terms of recall confirmed the worsening trend about the Population, Demographic and Complete solutions, as seen in Figures 10a and 10b. However, it is possible to notice that the Wage and Device solutions proved effective when recommending 5 apps and comparing them with the last 3 or 5 apps of the user. Such a situation might indicate that adding more specific context information can help in recommending applications.

5 Research Limitations

We faced some limitations while conducting this research. The first one is the location accuracy of the collected data. While the approximate locations guarantees the user's privacy, it also may lead to misidentifying the user's census tracts and enriching its data with wrong information. We acknowledge this limitation, but collecting a precise location is



(a) Recall for n=5.

(b) Recall for n=10.

Figure 10. Recall results based on categories context.

hard to obtain due to privacy concerns and so could lead to a study with potentially less users.

Another limitation regards the periods of the data used. While we used a dataset from 2019/2020, the census tracts' data were from the last Brazilian census in 2010. Besides that, we used census data only to identify the classes of mean wages in that region and not the absolute values. Even so, besides the income has increased in the last years, the country's geopolitics did not change significantly. The Human Development Index (HDI) went from 0.727 in 2012 to 0.765 in 2019, while the GINI rose from 0.5304 to 0.543 in the same period. Although showing that inequality increased, the change was not high enough to assume that regions before richer turned to poor, or vice-versa.

Finally, it is worth remembering that we did not perform A/B tests and neither evaluated the direct impact of our recommendations on users in this study. So, the main contribution is the evaluation of the user likelihood to install an app through the use of demographic and device information. Thus, we did not conduct any experiments to recommend apps to users in their day-by-day activities.

6 Conclusions and Future Work

This article investigated the impact of using demographic and device information in the context of users for mobile app recommendations. For this, we added such information to a user profile that contained only records of applications already installed, an approximate location, and the name of the mobile device used. From such data, we could evaluate the results obtained with application recommendations by two different strategies: Collaborative Filtering and Latent Dirichlet Allocation (LDA).

As for the results obtained with the application of the models, we can verify that the Wage and Device solutions concerned better overall results compared to the others. Such solutions are indicated when recommending applications, as the overall results for categories did not outperform the base solutions. However, it is noteworthy that the models were trained with apps, not categories.

In addition, we observe a superiority of the LDA model compared to the others, mainly with the use of user wage. These results may be related to the fact that LDA models work with association analysis between terms, which may have helped in the discovery of applications installed together. As for the collaborative filter, we noticed an overall improvement when adding user demographic information, but the model's performance did not hit big marks.

Based on the discussions raised above, one possible next step is to evaluate the explicit insertion of weights in LDA models. This is because, despite having achieved good results, we realized that the model was not able to extract potential patterns from the added features. Furthermore, as the models were only trained with applications, it would be interesting to train it with the categories themselves, performing a comparison between the approaches.

References

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022. DOI: 10.5555/944919.944937.

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. DOI: 10.1007/s10115-017-1124-3.

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. DOI: 10.1016/j.is.2017.08.006.

GSMA (2020). The Mobile Economy - The Mobile Economy. https://www.gsma.com/mobileeconomy.

IBGE (2021). Censo Demográfico | IBGE. https://www.ibge.gov.br. [Online; accessed 7. Jan. 2021].

Ipea (2008). Ipea. http://www.ipea.gov.br. [Online; accessed 25. Fev. 2021].

Liang, T., Zheng, L., Chen, L., Wan, Y., Yu, P. S., and Wu, J. (2020). Multi-view factorization machines for mobile app recommendation based on hierarchical attention. *Knowledge-Based Systems*, 187:104821. DOI: 10.1016/j.knosys.2019.06.029.

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. DOI: 10.1145/2684822.2685322.

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. DOI: 10.1145/2983533.

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. DOI: 10.1109/ASONAM.2016.7752297.

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. DOI: 10.5753/courb.2020.12371.

Matters, . (2021). Google Play Statistics and Trends 2021| 42matters. https://42matters.com. [Online; accessed 7. Jan. 2021].

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

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. DOI: 10.48550/arXiv.1301.3781.

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. DOI: 10.5555/2900423.2900554.

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. DOI: 10.1007/s11280-017-0456-y.

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. DOI: 10.1145/2684822.2685324.

Sarwar, B. M., Karypis, G., Konstan, J. A., Riedl, J., *et al.* (2001). Item-based collaborative filtering recommendation algorithms. *Www*, 1:285–295. DOI: 10.1145/371920.372071.

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. DOI: 10.1002/asi.23932.

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. DOI: 10.1109/ICDE.2017.43.

Zhu, K., Xiao, Y., Zheng, W., Jiao, X., and Hsu, C.-H. (2021). A novel context-aware mobile application recommendation approach based on users behavior trajectories. *IEEE Access*, 9:1362–1375. DOI: 10.1109/ACCESS.2020.3046654.