POLITECNICO
MILANO 1863

School of Industrial and Information Engineering

Biomedical Engineering

E-HEALTH METHODS AND APPLICATIONS PROJECT

Academic Year 2018/19

Identification and characterization of health apps

Group 4:

Caterina Achil 883538

Fabio Cubeddu 896052

Giulia Ferrari 883329

Emanuele Stagno 900340

Francesco Raffaele Tranquillo 905980

Silvia Traldi 883291

Summary

1 Introduction

Given the huge number of health-related Apps available online and given their increasing relevance for healthcare purposes, there is the necessity to understand which and how many Apps are nowadays useful for the user, and to understand how they can improve the quality of life of the patients.

1.1 Aim of the Project

The aim of the project is to classify the health-related apps available in the U.S. Apple Appstore.

The project outline is divided into two parts: in the first one the goal is to identify all the Apps related to the “Medical” and “Health and Fitness” categories and to extract information from the Apps webpages in an automated way, collecting them into a database.

The second part aim is to characterize all the Apps retrieved, classifying them into the specific medical categories they belong to, and then, to analyze the Apps features by mean of the newest and most relevant methods available from the recent literature.

2 Part I: Identification

At the earliest stage of the project, the goal has been to collect all the Apps belonging to “Medical” and “Health and Fitness” categories. The task has been performed by building an automated R code able to browse the app store in both categories, for any given ‘initial’ of the Apps and for each page; for each App the Name, the URL and the ID have been extracted and saved into three different databases, collecting respectively Medical (41649) , H&F (80724), and the merge of the two (105958), after removing the Apps that belong to both categories.

After the gathering has been completed, the database has been populated with a set of 25 attributes for each App.

This task has been achieved by mean of a R code able to browse the HTML of each page, extract the required attributes and collect them in the available CSV file.

We decided to use the CSSselector for each attribute. The attributes using a specific add-on of the web browser Chrome, SelectorGadget which was also referred in the documentation of Rvest package [https://blog.rstudio.com/2014/11/24/rvest-easy-web-scraping-with-r/]. The add-on’s use is designed to be quite straight forward and it could retrieve the CSSselector of an html page with the use of only a few clicks. Clicking on an element of a web page, while SelectorGadget is active successfully retrieving the correct selector, which was ready to be copied and used as the argument of the function html_node.

In this phase of the project several issues have been found: the features of the R code made the time increase exponentially, the AppStore set a maximum number of queries per minute performable by a user and the number of Apps to be analyzed has been huge. In order to solve the timing and the efficiency issues, a limit of the number of queries per minute has been included in the code and the dataset has been divided among the Group components by mean of GitHub; these solutions allow the Group to perform the task in 7 hours, by working in parallel in 7 computers simultaneously, against the 100 hours of the prevision.

2.1 Preprocessing

Once the full database has been available, the task of Pre-processing has been performed: firstly, a language detection process has been necessary, in order to select only the English language Apps, secondly Non-ASCII characters has been removed from the database, because of their incompatibility with MetaMap tool and to avoid further problems with R.

For the language detection TextCat library [https://www.rdocumentation.org/packages/textcat/versions/1.0-6/topics/textcat] has been used: it is a specific R package which is able to automatically recognize the language of a text given as input.

To maintain ASCII characters only a specific regular expression ([^\x20-\x7E]) has been implemented, which is composed by a capture group that select all the characters that are not present in the range "20"-"7E", the hexadecimal representation of ASCII characters spacing from the character "SPACE" to "~".

3 Part 2: Characterization

The aim of the second part of the project has been, firstly, to classify the Apps retrieved among a list of medical branches, providing information about their distribution among them and other relevant statistics; by mean of the training set and the test set used, the algorithm performances have been calculated and evaluated.

The second task has been to extract all the Apps belonging to the "Dermatology" category and rank them by mean of an automated code based on "Albrecht et al.,2018 Criteria".

3.1 Classification

At the earliest stage of classification task, the Group has manually assigned the belonging categories to a set of 150 random-extracted Apps, in order to have well defined training and test sets, both used toimprove the performance of the classification model.

The classification process has been based on three steps:

1. Classification of Not-Medical Apps
2. Classification of Apps that belongs to Across specialities
3. Classification of the Medical specialties Categories

The first two have been approached by developing an automated way based on Machine Learning-related Method of Text Mining and Bayes Naïve methods.

Particularly the "*TM*" package has been used to clean all the database's descriptions via different transformations such as the removing of stopwords, numbers and document "stemming". Then, thanks to the calculated frequency of each term retrieved among all the medical and not medical Apps description, we obtained a "dictionary" to be used for the proper classification; the occurrence of the most frequent terms is shown in Figure 1, that highlight the distribution of each word among the two opposite categories.

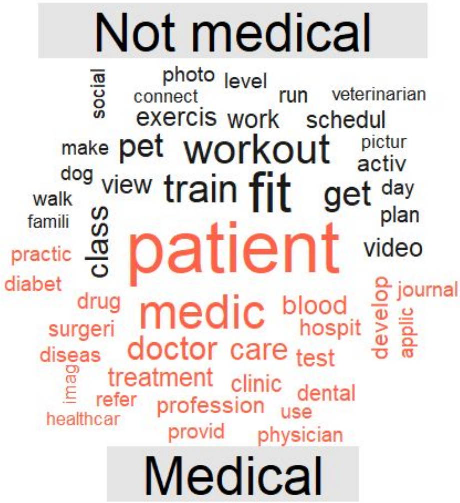

Figure 1: Occurrence of the most frequent terms

The output provided by *TM*, then, has been processed by the Bayesian Naive algorithm, that has computed the probability of a term to be associated to Not-Medical or Medical Categories, giving an highly accurate classification; to achieve this task, the "*Caret*" (Classification And REgression Training) package has been used.

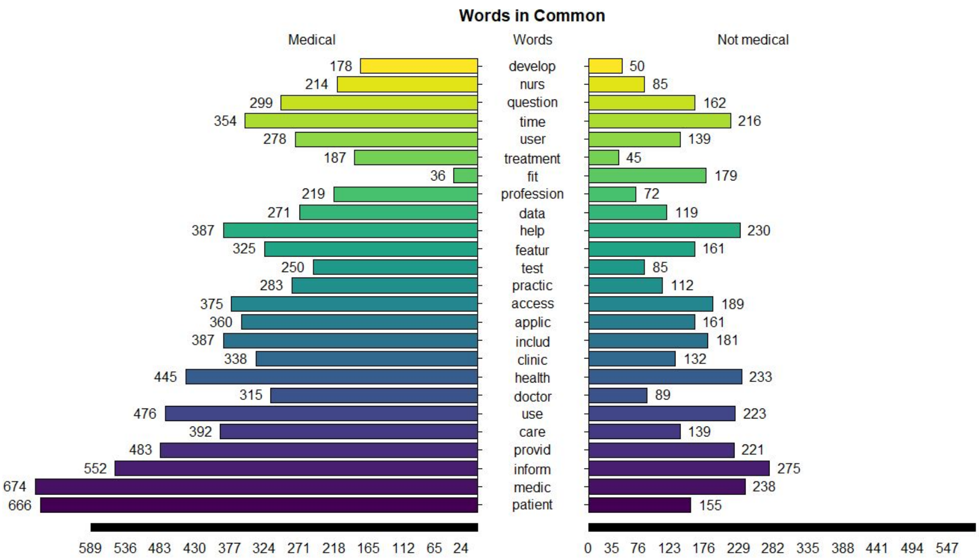

Figure 2: Words in common and their frequency in both category medical and not medical

How stated before, the training set has been a sample of 1300 manually classified Apps, while the testing has been performed in 75 Apps; the performances will be shown in chapter 3.2.

For what concern the Apps belonging to specific medical specialties, the task has been approached by mean of MetaMap and R in the last step of the process of classification. Specifically, through the use of Metamap the candidates preferred of each app have been retrieved and compared with the provided list of Mesh terms, than the most frequent categories have been saved. It was chosen to take in account only the top three categories, because during the manual classification it was noticed that usually the relevant one are no more than three.

During the optimization of the training set, it has been necessary to update the list of Mesh terms, by subtracting and adding relevant terms.

However, a relevant issue arised: the MetaMap Batch has not been able to perform the analysis of the descriptions submitted in an acceptable time, due to a long waiting list. To obviate the problem the dataset has been fragmented into 7 parts and the queries has been performed installing and using MetaMap on two computers.

The local version of MetaMap has been able to provide the output for 11000 Apps. Then, the list of Apps has been processed by R code.

The final output of the specific medical categories is shown in Figure 3 and Figure 4.

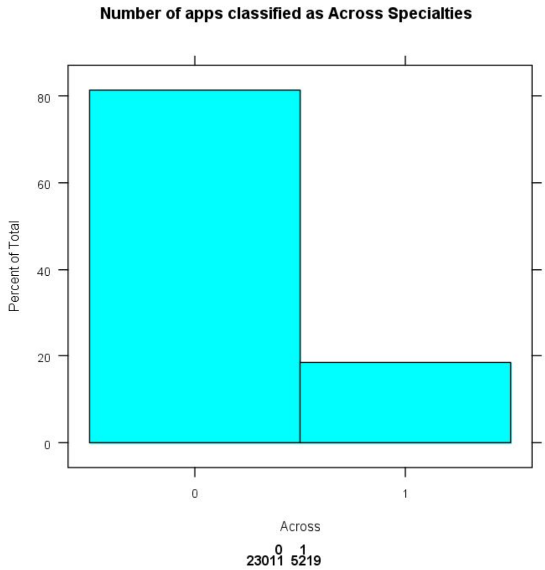

Figure 3: Number of apps classified as Across Specialties

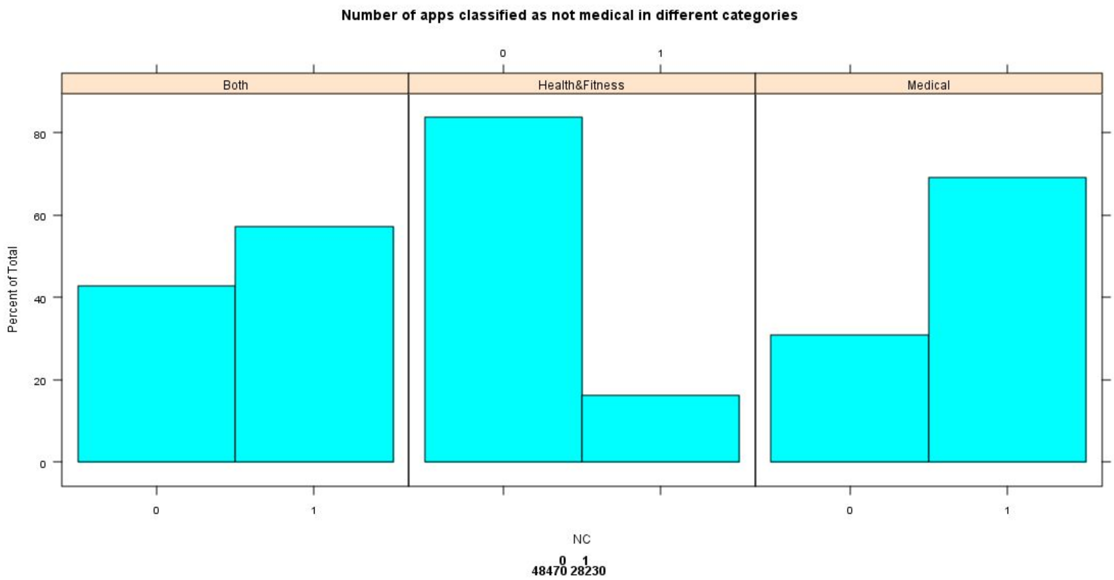

Figure 4: Number of apps classified as Across Specialties

Actually in the definition of the algorithm classification some other approaches have been evaluated. More in details, for the first two steps, it was considered to create two new mesh terms for NC and Across respectively, extracting the candidate preferred from the description of the Apps classified in the database of the entire class . However it was preferred to not proceed in the case of the Not Medical Apps, because Metamap is a tool made to analyse biomedical terms and at the same time the aforementioned text-mining based approach was getting good results. On the other hand, for Across Specialities, the interpretation of the categories was not consistent across the whole class' manual calssification, and the mesh obtained has not been as sufficiently confident and objective as to discriminate "Across Specialties" in a correct way. For these reasons it has been chosen to replicate the method used for the NC classification also for the Across Specialties.

3.2 Performance Measurement

This chapter aim is to show the performances obtained by the implemented algorithms on the test set.

It has been decide to show the confusion matrix for NC and Across, in order to represent the performance of the two Text Mining algorithm. Columns represent reference while rows represents predicted values

	0	1
0	28	3
1	9	35

Table 1: Confusion matrix of Across specialties

	0	1
0	48	16
1	2	9

Table 2: Confusion matrix of Not Medical

From the Confusion Matrix it was also calculated the Accuracy, Sensitivity and Specificity, using the formulae represented in the table. These values were also calculated for the different medical categories and in order to have an overall view on the performance on the categories it was also computed the average value.

	Accuracy	Sensitivity	Sensibility
Not Medical	0.84%	0.76%	0.92%
Across	0.76%	0.36%	0.96%
Category (Average on different specialties)	0.94%	0.51%	0.95%

Table 3: Performance on the test set

$$Accurancy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Sensitivity = \frac{TP}{TP+FN}$$

$$Specificity = \frac{TN}{TN+FP}$$

3.3 Apps feature analysis

The last task of the project has been to extract the *Dermatology* Apps retrieved, from the whole database, to analyze their features by mean of *Albrecht et al., 2018 criteria,* and to order them by several quality factors.

The classification criteria have been based on the attributes retrieved in the first part of the project.

A code has been implemented in order to evaluate the attributes, normalize them and obtain the final result weighting the criteria suggest by the reference. The apps have been ranked following the criteria scores.

The top ranked 60 Apps of the Dermatology category have been extracted and classified manually by the Group, according to function types and subject areas.

As optional part, the ranking of Not-Medical Apps has been performed. However, in this case it would have been opportune, in our opinion, to change the weighting factors of the different criteria. Indeed, for example, the criterion with the greatest weight was the Keyword-based evaluation that searched through the descriptions the regulatory status of the app (looking for the words "medical device", "FDA" etc..). For the NC category it would be appropriate in our opinion to significantly reduce the weight of this factor, given the fact that in NC many apps were not related to medical content at all.

The following graph shows the distribution of the functions retrieved.

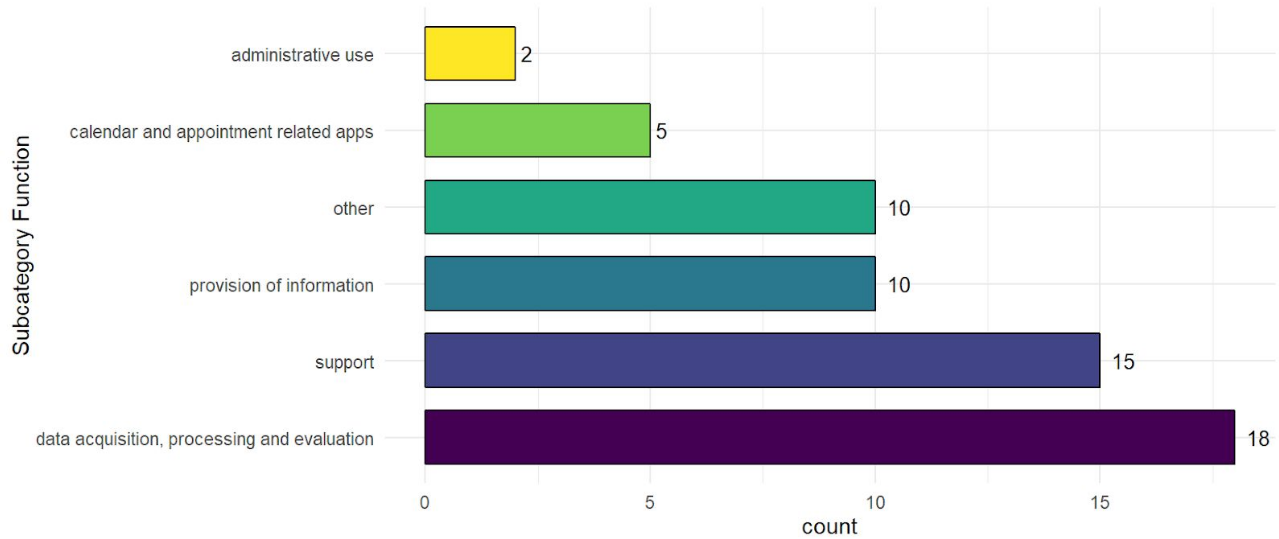

Figure 5: Distribuitio of the functions retrieved

Finally, it has been useful to compare the top 60 Dermatology Apps with the other Dermatology, in order to find and highlight the differences and the patterns; the table shows the most meaningful statistics found.

In general, Top ranked apps shows an average size and number of user rating (54.509 kB and 34) higher than the other Dermatology apps (35.262 kB and 0). This could be explanatory that higher ranks are charachterized by the presence of more content, such as videos or images and, in general, an higher number of user ratings could signify that users are more involved in the use of a higher rank app, and so more inclined to rate it. An interesting case is given by the variable Price. Given the high number of free apps, it's interesting to highlight that Top 60 apps are characterized by a lower average price (0.2825 $) than the rest of apps (2.197 $). This could be explained considering the fact that an higher rank medical app, to be useful for the society, should be available for free, providing health-related solutions to an higher spectrum of users.

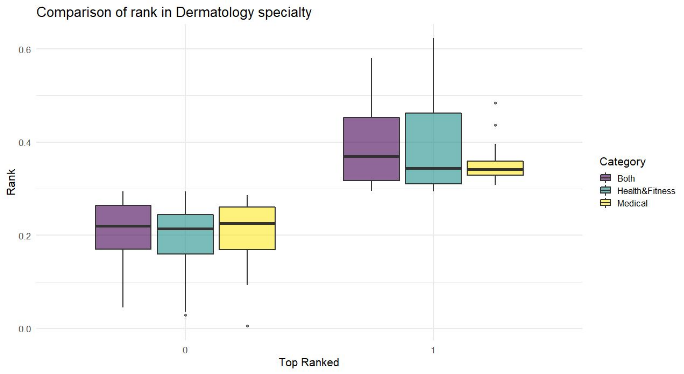

Figure 6: Comparison of rank in Dermatology specialty

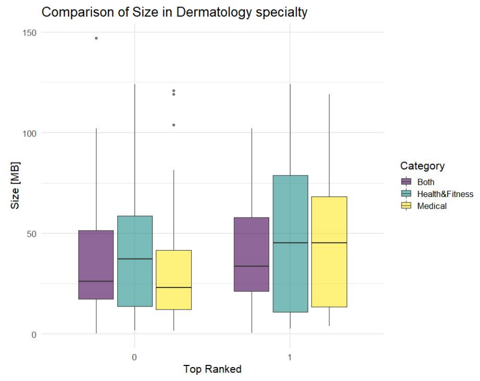

Figure 7: Comparion of size in Dermatology specialty

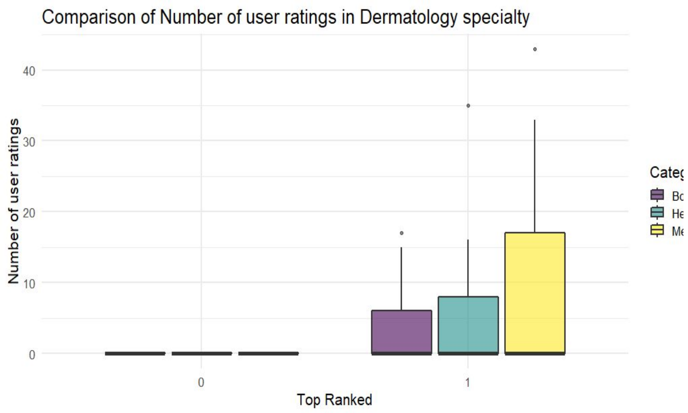

Figure 8: Comparison of number of user ratings in Dermatology specialty

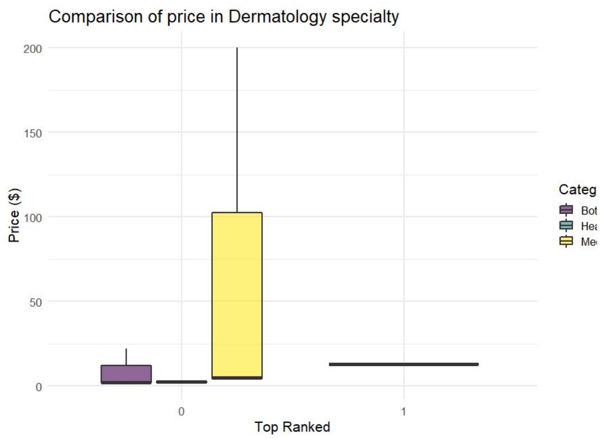

Figure 9: Comparison of price in Dermatology specialty

4 Conclusion and future developments

This chapter aim is to show the possible future development of the project, which are strictly related to the main problems found:

1. MetaMap Batch service did not give an answer for the whole database classification. However, the installation of a local version of MetaMap solved the problem. We think that further analysis should continue to be performed using MetaMap in local, since it also showed, during queries, to not be resource-expensive in terms of RAM used.
2. Apps description are often unclear and short, with several unuseful parts. A major complication is given by descriptions which contains any form of punctued lists. This text structure has been able to freeze the execution of MetaMap multiple times, slowing the classification process. Further works should implement an improved preprocessing phase, which for example should remove descriptions with lists.
3. Mesh terms lists was poor and need to be enriched with Text Mining algorithms, in order to find other Candidate Preferred to enlarge the list and improving the matching phase.

4. Given the good results for the classification of NC, we believe that future works should continue to use a text-mining based approach as a filter to reduce the number of apps to be studied. Other works could try to compare other classification methods in order to obtain even better performances.