# 1. About the project

The main idea behind this project was to explore the recommendation algorithms in the area of mobile applications, using collaborative filtering to discover the similarity between users and recommend according to the found information.  
  
The dataset needed for this project was extracted from this location: http://baltrunas.info/research-menu/frappe. The frappe dataset contains 96202 records by 957 users for 4082 apps. The huge amount of apps that can be found on Google Play has made it very difficult for users to discover great and relevant apps, and the solution to this problem has been offered through this project.

The recommendation system has been set to use the Pearson correlation coefficient to calculate similarity between users based on their download history of apps. The algorithm recommends k apps with the highest value for the correlation coefficient, where k is set to be 5 in the given solution, but can be adjusted for different implementations.

# 2. Dataset

The dataset has been split into two files that contain coma separated values (csv). For the implementation of recommendation algorithms, the three most important parts are item, user and rating, which in this case, is represented by the history of users download of certain app.

The first file: ''frappe.csv'' contains 11 columns and 96202 rows which represent the user's history with mobile apps.

The columns it contains are:

* **user** column contains user id's, and is therefore represented as integer.
* **item** is a column that contains item id's which can be traced to a different file which contains more detailed information about each of the apps.
* **cnt** is a number of times application was used (in a specific context).
* **daytime** hasone of 4 possible values: Morning (6am to 12am), Afternoon (12am to 6pm) Evening (6pm to 12pm), Night (12pm to 6am).
* **weekday** has one of 7 possible values: Mon, Tues, Weds, Thurs, Fri, Sat or Sun.
* **isweekend** has one of 2 possible values: Weekend or Working day.
* **homework** gives the information about the place where the app has been used.
* **cost** is determined by the price that has or has not been paid for the app.
* **weather** has one of 9 possible values: Sunny, Cloudy, Foggy, Windy, Drizzle, Rainy, Stormy, Sleet, Snowy
* **country** in which the app has been downloaded
* **city** -coded city names

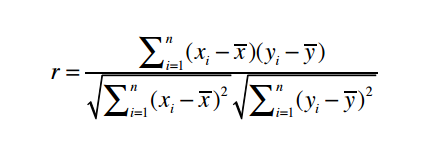
The second file used is ''meta.csv'' and it contains more detailed information about the applications. The columns it contains give more precise view of the apps, as it has details about the name of the app, description, its rating on app store and so on. This file has been used to recommend the mobile application by its name based on its id that is given in the first file in item column.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

# 3. The solution

The main idea behind this recommender system is to explore the dataset to discover the similar users to a user being suggested an application, so it can recommend the items those users marked as useful according to their usage history. Applications which would then be suggested should not already be used by user for which the recommendation is being built. Instead, the first k applications sorted by the biggest predicted score would be recommended. The metric that is used for determining the similarity between users is Pearson correlation coefficient.

The Pearson Correlation Coefficient is a measure of correlation between two variables (in this specific case the correlation between two users). It ranges between -1 and 1 inclusive. 1 indicates perfect agreement. -1 indicates perfect disagreement. This way we can use it to find the individual who is most similar to the person we are interested in. The formula for the Pearson Correlation Coefficient, if we have one dataset {*x*1,...,*xn*} containing *n* values and another dataset {*y*1,...,*yn*} containing *n* values then that formula for *r* is:



with the sample mean being: {\displaystyle {\bar {y}}}



and analogously for  .

# 4. Technical realization

This application is written in Python.

The pandas package has been imported for reading and manipulating the csv file with the loadDataSet function inside the recommender class.

The overall usage history of application is recorded inside the dictionary which, because of its hashable nature, allowed to join together users and all of their used apps in a most convenient form for recommendation.

The recommender class contains the function which computes the nearest neighbors, in this case, the most similar users. This function is being used in recommend function which requests user id as an input parameter and returns k application which are best suited for the given user.

# 5. Acknowledgements

This application has been developed as a part of the project assignment for the subject [Intelligent Systems](http://is.fon.rs/) at the Faculty of Organization Sciences, University of Belgrade, Serbia.

The literature that has been used in realization of this project can be found here:

1. *A programmer's guide to data mining* http://guidetodatamining.com/
2. *Programming collective intelligence* by [Toby Segaran](http://shop.oreilly.com/product/9780596529321.do#tab_04_2)

# 6. Licence

This software is licensed under the [MIT License](http://opensource.org/licenses/MIT).

The MIT License (MIT)

Copyright (c) 2017 Bojana Ranković - [bojana.nu@gmail.com](mailto:atasic90@gmail.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.