Predicting Google Play Store app downloads

Members: Rauno Raa, Aleksander Kattel

GitHub link: https://github.com/raunoraa/IDS2022-2023-Project

Business understanding

Identifying your business goals

Our clients, android app developers, seek to make useful applications in different categories to improve Android phone users' lives. App developers want their app to be popular and profitable, however it is not obvious how to have a popular and profitable app. This is where machine learning can help - our goal is to help Android app developers decide what is important for their apps to achieve their desired profits and popularity in the app category.

Assessing your situation

For this project we are limited to two people with laptop and desktop computers, which have appropriate software for implementing machine learning algorithms. We also have access to data, which has information about 2.3 million different apps gathered in June 2021. We are required to provide the results by 12th of December 2022. The only reason why the project may be delayed is if one of us can't work in a certain period for some reason. The solution in that case would be to redivide the tasks between us so that in the period when the other can't contribute the other member will contribute to the project. The project will cost us our time and electricity but gaining knowledge about how to make popular and profitable apps will certainly outweigh the costs.

Terminology:

- Android app - A program which runs on the phones with Android operating system
- Machine learning algorithm - Algorithms to search for patterns in the data
- Hyperparameter tuning - Finding a set of optimal hyperparameter values for a learning algorithm (for example with decision tree algorithms it is possible to set the maximum depth, max splits etc.)
- Machine learning model - A file that has been trained to recognize certain types of patterns

- Model accuracy - How accurately is the model predicting the results (compared with the available real results)

Defining your data-mining goals and producing your project plan

Our data-mining goal is to implement machine learning algorithms on our data, gather the results, and make a presentation to show our results to the public. Our model accuracy should be at least 90%.

To complete our projects we have to follow multiple steps. First we need to gather the data, then we need to clean the data to be able to apply machine learning algorithms on it, then we need to think of the algorithms, which would be useful in our situation, then we need to do hyperparameter tuning, then apply these algorithms to our data, then we need to analyze the results and decide if they are accurate enough. If the results are not accurate enough, then it may be necessary to repeat the steps of finding suitable algorithms, hyperparameter tuning and applying the algorithms. If the results are accurate enough, then we can make a presentation based on the results and then present it to the public.

Data understanding

Gathering data

The requirements for our data are: the data should be gathered within the last 5 years, the data should include at least one million apps, the data should be diverse enough for us to train a machine learning model based on it and should be publicly available for us to use. We are using a public dataset from Kaggle that meets all of our requirements and is available. We do not plan to use all of the dataset, as some attributes are completely useless for our purposes. We have loaded the data into a Jupyter Notebook and verified that the data is usable.

Describing data

The source of the data is the Kaggle website. The data is in .csv form and consists of 24 columns and 2312944 rows. Each row of the dataset represents an application from the play store. The app is described using various attributes, that include both categorical and numeric data. There are quite many attributes, which we do not need, such as any info that is completely unique for every row or information related to the developer of the app for instance. The data that we plan to use is all present and usable.

Exploring data

For this part of the report, we are not including the attributes that we are planning to drop. As for the data we are using, for the most part, it consists of categorical attributes. Initially we planned on predicting the amount of installs as a numeric attribute. However, looking at the data showed us that the amount of installs is actually a categorical attribute. The amount of installs is defined by a range eg. 10-50, 500-1000, etc . This is a very useful discovery for our purposes, because it means we can use machine learning models for classification. We also discovered that some attributes that we first thought would be useslees, could instead turn out to be somewhat useful. This is the case with Privacy Policy. The attribute is a link to the privacy policy of the application. We figured we could set every value that has a link to 1 and non-existent ones to 0 and therefore have a categorical attribute which shows whether the app has a privacy policy. For some numeric attributes we might have to do some classification. For example, Rating, which is a numeric attribute from 0-5. It might be better to separate the values into bins (0-1, 1-2, etc) and use them as categories.

Verifying data quality

We have verified that the data exists and is usable, as we have loaded it into a notebook and observed the values. Upon first look, we did not spot any data quality issues. We observed the distributions and amount of null values of the features and did not spot any anomalies.

A detailed project plan

Tasks with contributors and time estimations:

- Creating a GitHub repository - Rauno: 0.25 hours
- Creating a Jupyter Notebook file - Aleksander: 0.25 hours
- Cleaning the data to be fit for applying machine learning algorithms on it - Rauno: 5 hours ; Aleksander: 5 hours
- Finding the suitable machine learning algorithms for our data - Rauno: 7.5 hours ; Aleksander: 7.5 hours
- Doing the hyperparameter tuning and applying the algorithms to our data - Rauno: 10 hours ; Aleksander: 10 hours
- Analyzing the results and making a presentation - Rauno: 7.5 hours ; Aleksander: 7.5 hours

Total hours = Rauno: 30.25 hours + Aleksander: 30.25 hours = 60.5 hours

Tools:

- For storing the code we are going to use GitHub
- We are going to use Jupyter Notebook for coding. It is convenient to run the code as small blocks (useful for developing and debugging as we don't have to rerun the whole code again each time we want to test something)
- For communicating, we can meet in person or use Facebook Messenger and Discord