# PROJECT PROGRESS REPORT

## CSE-4990/6990 - Big Data and Data Science

### Predicting Google App Rating

Team 3: 5DBminds

Date: October 13, 2015

Ayush Raj Aryal
Daniel John Sween
Lucas Andrade Ribeiro
Naila Bushra
Naresh Adhikari

## Project overview

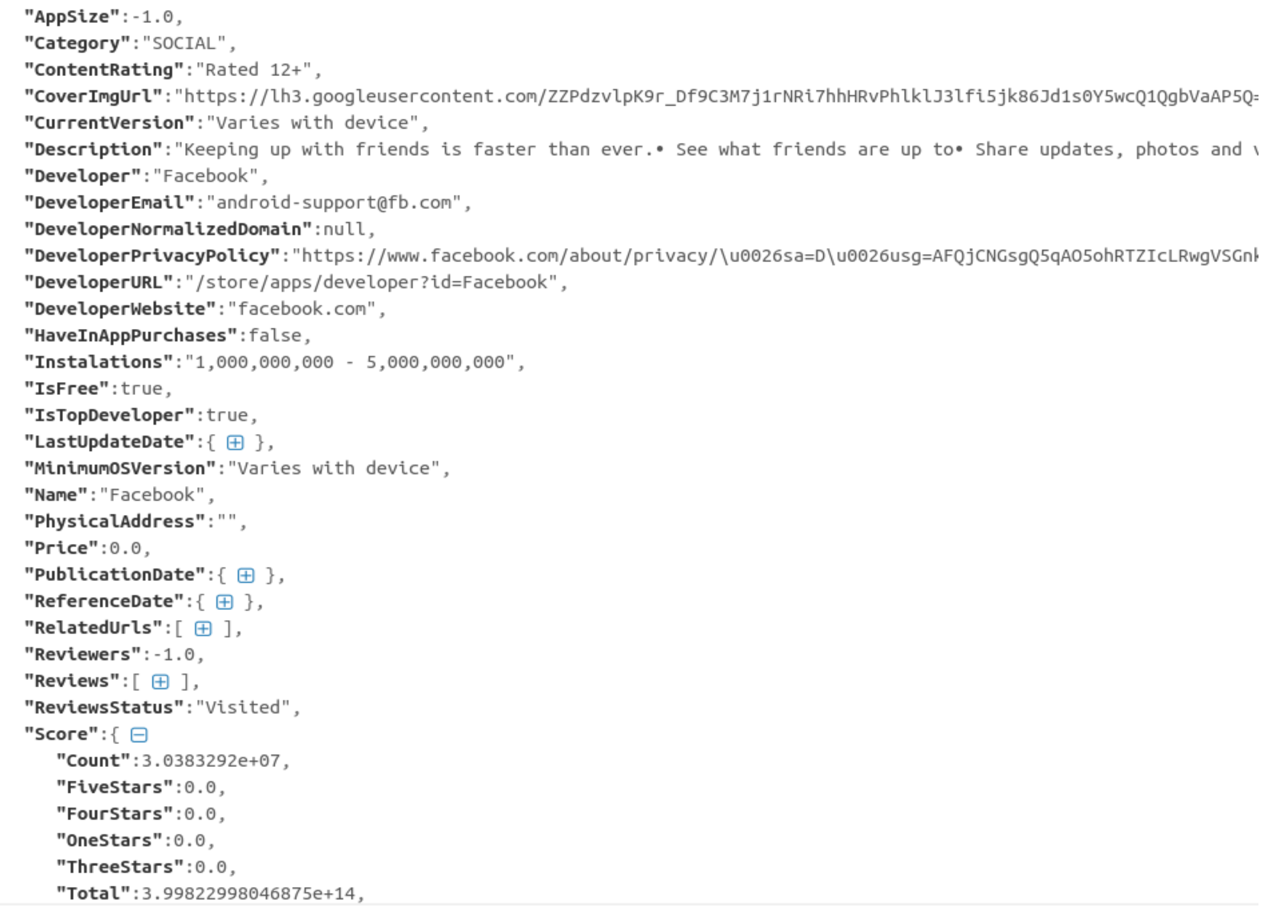
As a part of our course work we are doing a data-analytics project entitled “Predicting PlayStore Rating for Apps” which uses publicly available data. The goal of the project is to predict PlayStore rating for the newly uploaded application. To achieve that we will proceed by analyzing the existing PlayStores data of a large number of applications and then build a model which will be able to predict ratings for future apps based on the attributes of that application.

## Data Collection

The data of this project has been collected from the author of the github repository GooglePlayAppsCrawler

GitHub Link: https://github.com/MarcelloLins/GooglePlayAppsCrawler.git)

The project is open for contribution and the crawled data for Google Playstore Application is available upon request to the author. Here is simple view of the attributes of the data



```
"AppSize":-1.0,
"Category":"SOCIAL",
"ContentRating":"Rated 12+",
"CoverImgUrl":"https://lh3.googleusercontent.com/ZZPdzvlpK9r_Df9C3M7j1rNRi7hhHRvPhlklJ3lfi5jk86Jd1s0Y5wcQ1QgbVaAP5Q=
"CurrentVersion":"Varies with device",
"Description":"Keeping up with friends is faster than ever.• See what friends are up to• Share updates, photos and
"Developer":"Facebook",
"DeveloperEmail":"android-support@fb.com",
"DeveloperNormalizedDomain":null,
"DeveloperPrivacyPolicy":"https://www.facebook.com/about/privacy/\u0026sa=D\u0026usg=AFQjCNGsgQ5qAO5ohRTZIcLRwgVSGn
"DeveloperURL":"/store/apps/developer?id=Facebook",
"DeveloperWebsite":"facebook.com",
"HaveInAppPurchases":false,
"Instalations":"1,000,000,000 - 5,000,000,000",
"IsFree":true,
"IsTopDeveloper":true,
"LastUpdateDate":{ ⊞ },
"MinimumOSVersion":"Varies with device",
"Name":"Facebook",
"PhysicalAddress":"",
"Price":0.0,
"PublicationDate":{ ⊞ },
"ReferenceDate":{ ⊞ },
"RelatedUrls":[ ⊞ ],
"Reviewers":-1.0,
"Reviews":[ ⊞ ],
"ReviewsStatus":"Visited",
"Score":{ ⊟
   "Count":3.0383292e+07,
   "FiveStars":0.0,
   "FourStars":0.0,
   "OneStars":0.0,
   "ThreeStars":0.0,
   "Total":3.99822998046875e+14,
```

**Fig 1: Screenshot of the attributes**

## Data Handling

The data itself is in a json format in one large file named PlayStore_2015_07.json. In order to handle this data we have split it into around 62 files each file having 1000 lines of json data.

```
split -l 1000 PlayStore_2015_07.json <split_directory>
```

## Data Preprocessing

To process the data into the our program we have used pandas DataFrame structure which reads all the split json files into Dataframes and converts them into csv files which are convenient to use when dealing with range of features of the playstore application.

From the attributes of each application in the json data we have identified a subset of the attributes which will be useful for building a prediction model. The list is given below:

- AppSize
- Category
- ContentRating
- Developer
- Description
- HaveInAppPurchases
- Instalations
- IsFree
- IsTopDeveloper
- LastUpdateDate
- Name
- Price
- PublicationDate
- Reviewers
- Reviews
- ReviewsStatus
- Score:{"Count","FiveStars","FourStars","OneStars","ThreeStars","Total","TwoStars"}

Among the above mentioned attributes the score is our label for the data which is calculated by first calculating how many stars the app has been given in total by users then dividing it by the count of the users. Following is the ER diagram for our application database.

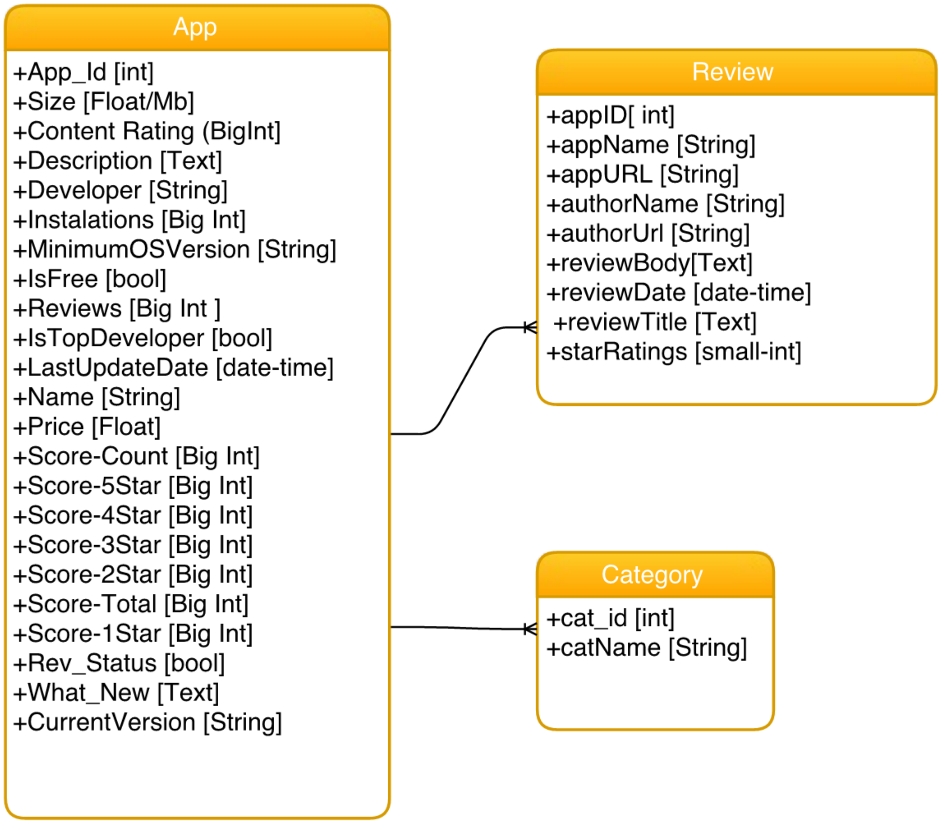


**Fig 2: ER Diagram of the Google App Rating Data**

## Issues

- Some of the attributes values are not numerical, so we are to assign numerical values to the feature values in order to extract and form the feature vectors. We will get the final training set after all the pre-processing.

## Risk

- We are currently dealing with the data of nearly 11,000 apps so our biggest concern is the computational time it will take to build a classification model using all the app data at a time.

## Next Phase

- Assign numerical values to the Application features
- Standardize the data using standard score
- Construct feature vectors
- Apply supervised learning algorithm to build a prediction model using the training data
- Split the data into 70/30 ratio for train and test data.
- To avoid overfitting we will randomize the train and test data using approaches like k fold cross validation/ random sub-sampling validation etc.