**GROUP PROJECT**

**DATASET : *Google Play Store App***

*Group Members:*

1. Abidemi Lawal
2. Michael Ayako

Data source link:

<https://www.kaggle.com/lava18/google-play-store-apps#googleplaystore_user_reviews.csv>

The dataset comprises of 2 tables.

1. The first table contains 100 reviews of each applications on google play store. (googleplaystore\_user\_reviews.csv)
2. The second table contains google app data(googleplaystore.csv)

**Step 2 (due 11/3/2018): Computations (at least 500 words)**

Choose at least 4 computations to implement over your data, highlight them, and walk through them in depth.

* For each computation, include high-level diagram and/or explanation of the steps you are going to follow to complete the computation.
* At least one computation must be implemented in map-reduce with two Map reduce jobs where the output of the first job is used as input to the second job.
* At least one computation must be implemented as a Spark application.
* The other 2 computations can be implemented either in map-reduce or spark.

The four computations include:

**Computation 1**

**Does the type of App affect the Sentiment of the users? Are the customers using free apps more satisfied than the customers using paid apps?**

Using the Sentiment column in the first Table and the Type column in the second table. There is going to be two jobs, because of the JOIN. For Job 1, the mapper input will be **(Linenum(AppName,Sentiment))** from Table 1 and the Key for the Table 2 will be **(Linenum(AppName,Type))**

**Mapper 1**

**Input 1:**

**(Linenum(AppName,Sentiment))**

**Input 2:**

**(Linenum(AppName,Type))**

**Output:**

**(AppName(Type,Sentiment))**

The mapper output will be the **(AppName,( Sentiment, Type)).** This will also be the reducer input. The reducer output will be **(Type,Sentiment)**

**Reducer 1**

**Input:**

**(AppName(Type,Sentiment))**

**Output:**

**(Type,(Sentiment))**

For Job 2, the mapper input will be the output of the reducer in Job 1. “**(Type, Sentiment)”.**

The aim of the Job 2 is to aggregate positive sentiments and negative sentiments to a type of payment i.e. Paid or Free in order to proceed with the analysis. Mapper input “**(linenum(Type,Sentiment))”.** Mapper output is “**(Type(Sentiment))”.** Go through Shuffle and Sort, then filter using the Type. The Type with the highest number of positive sentiments.

**Mapper 2:**

**Input:**

**(linenum(Type,Sentiment))**

**Output:**

**(Type,(Sentiment))**

Reducer input will be a filter of the Paid Apps and Free Apps. If the Free App have more positive sentiments than the paid, this shows that customers using Free Apps are most satisfied than the customers that pay for the Apps the same applies for the vice versa. For each Type, a positive sentiment and a negative sentiment count will be processed, and the result will be used to provide the final analysis for the question asked.

**Reducer 2:**

**Input:**

**(Type,(Sentiment))**

**Output:**

**(Type,(Sentiment count))**

**Computation 2**

**Does the genre of the App affect the sentiments of the user?**

The aim of this computation is to provide analysis on what genres have positive sentiments and negative sentiments. We want to line up Genre and sentiments for this analysis to work. There are two jobs because of the JOIN. First job will perform a join. The second job will perform an aggregation of positive to negative sentiment of each Genre.

For Job 1, the mapper input:

**Mapper 1:**

**Input 1:**

**(linenum,(AppName,Genre))**

**Input 2:**

**(linenum,(AppName,Sentiment))**

**Output:**

**(AppName,(Genre,Sentiment))**

The mapper output will be the **(AppName, (Sentiment, Genre)).** This will also be the reducer input. The reducer output will be **(Genre,(Sentiment))**

**Reducer 1:**

**Input:**

**(AppName,(Genre,Sentiment))**

**Output:**

**(Genre,(Sentiment))**

For Job 2, the mapper input will be the output of the reducer in Job 1. “**AppName(Sentiment,Genre)”.** The aim of the Job 2 is output the aggregate and provide the ratio of positive to negative reviews of each genre, this will provide us with the analysis we were looking for. Mapper input “**(linenum,(Genre,Sentiment))”.** Mapper output is “**AppName(Sentiment,Type)”.** The Average of the two Types.

**Mapper 2:**

**Input:**

**(linenum,(Genre,Sentiment))**

**Output:**

**(Genre,(Sentiment))**

Reducer input will be an output of the Genre and the number of sentiments. The genres will be highlighted in positive to negative sentiment ratios

**Reducer 2:**

**Input**

**(Genre,(Sentiment))**

**Output:**

**(Genre,(Sentiment ratio))**

**Computation 3**

Find the top 5 most downloaded apps. Using Table 2, the Key is **AppName** and Value is **Installs.** Using Java MapReduce, this projects how many people download the apps using the **Installs** column.

**Mapper:**

**Input:**

**(lineNum,(AppName,Installs))**

**Output:**

**(AppName,(Installs))**

**Reducer:**

**Input:**

**(AppName,(Installs))**

Here we will sort the Input and output the top 5 downloads. This will output the top 5 most installed apps in the database

**Output:**

**(AppName,(Top 5 Installs))**

**Computation 4**

Find the category rating. Using Table 2, Key is **Category** and Value is the **Rating**. Using Java, we will sort through all app categories and store those values onto mapper output,

**Mapper:**

**Input:**

**(linenum,(Category,Rating))**

**Output:**

**(Category,(Rating))**

The reducer input will be as stated above. The output will be an average rating of all apps in that category.

**Mapper:**

**Input:**

**(Category,(Rating))**

**Output:**

**(Category,(Average Rating))**