Project Report

October 11, 2019

1 DATA1902 Project Stage 1

490398331, 490407356, 480048691
Lab: Friday 01pm Link Building 122
University of Sydney | DATA1902 | October 2019

The report considers Apple App Store and Google Play Store datasets. The aim of this report is to describe the two datasets and provide simple statistics about mobile apps. Firstly, it highlights the source of the data, its contents and format. The second section explains the data cleaning process in Python. Finally, the third section provides simple analysis about the data.

1.1 I. Data Source

The Apple App Store and Google Play Store datasets were both sourced from Kaggle. Kaggle is a website that provides a public dataset platform. Here, each dataset provides its licensing such that it can be used appropriately.

Both datasets are comma-separated values (CSV) files. Therefore, the datasets separates its fields with commas.

The Apple App Store data contains 16 columns and 7197 rows. That is, it provides information about 16 variables about 7195 mobile applications on iOS. Similarly, the Google Play Store data has 14 variables to describe around 10 840 Android apps (of which 9660 are unique values).

User ratings, app categories and reviews are independent variables common to both datasets. Therefore, the contents of our datasets are relevant to the user experience in mobile apps. Moreover, the datasets contain information about the number of downloads. Thus, we can investigate the popularity of particular apps and app categories.

1.2 II. Initial transformation

Firsly, given our goal is to have a brief yet cohesive understanding of the two datasets, we have decided to narrow down the number of variables to 8 main common variables that we consider to be important for simple analysis and remove irrelevant data. The variables are of the followings:

1. App: the name of the apps
2. Ratings: user ratings out of 5

3. **Number of ratings:** total number of user ratings
4. **Genres:** categories such as entertainment, health, etc.
5. **Price:** straightforward app pricing, not including in-app purchases
6. **Size:** file size of the mobile app (in Bytes)
7. **Content rating**
8. **Current version:** the most recent version of the app at the time being scraped

Furthermore, since the datasets were scraped multiple times at different time points, we experience duplicates which will be removed. In order to reformat efficiently, we employ the use of the following packages `panda`, `io`, `csv`, and `backsports`.

[1]:
```
import pandas as pd

#Reformatting Apple Store dataset

df = pd.read_csv("AppleStore.csv")
df["Store"] = "App Store"
df.rename(columns={'track_name':'App', 'user_rating':
 ↪'Rating','rating_count_tot':'Reviews', 'prime_genre':'Genres', 'size_bytes':
 ↪"Size", 'price':'Price', 'cont_rating':'Content Rating', 'ver':'Current␣
 ↪Version'}, inplace = True)
del df['id']
del df['currency']
del df['rating_count_ver']
del df['user_rating_ver']
del df['sup_devices.num']
del df['ipadSc_urls.num']
del df['lang.num']
del df['vpp_lic']
df = df[["Store", "App", "Rating", "Reviews", "Price", "Size", "Content␣
 ↪Rating", "Genres", "Current Version"]]

df.drop_duplicates()

df.to_csv("Apple_Store.csv", encoding='utf-8')
```

[2]:
```
#Reformatting Google Store dataset

df = pd.read_csv("googleplaystore.csv")
df["Store"] = "Google Play Store"

df.rename(columns={'Current Ver':'Current Version'}, inplace = True)
del df['Category']
del df['Installs']
del df['Type']
del df['Last Updated']
del df['Android Ver']
```

```
df = df[["Store", "App", "Rating", "Reviews", "Price", "Size", "Content␣
 ↪Rating", "Genres", "Current Version"]]

df.dropna(axis='rows')

df = df.drop_duplicates(subset='App', keep='first')

df.to_csv("Play_Store.csv", encoding='utf-8')
```

1.2.1 Highlights of transformation

1. Among a total of 17953 mobile applications coming from both Google Play Store and Apple Store, 1474 apps do not have a rating. We set the default rating for these apps as 0.
2. We synchronised the format of app size and content ratings for the entries in both datasets for convenient comparison.
3. We removed an entry in Google Play Store with missing data in the "Category" variable which has caused a column shift.

1.2.2 Merge Datasets

[3]:
```
import backports
import io
import csv

## Need to create blank csv file beforehand
output_file = io.open("cleaned.csv", 'w', newline='', encoding='utf-8')
csv_writer = csv.writer(output_file)

with io.open("Apple_Store.csv", newline='', encoding='utf-8') as f:
    for row in csv.reader(f):
        csv_writer.writerow(row)

with io.open("Play_Store.csv", newline='', encoding='utf-8') as f:
    rowcount = 0
    for row in csv.reader(f):
        rowcount += 1
        if row[2] == "Life Made WI-Fi Touchscreen Photo Frame":
            continue
        elif rowcount >= 2:
            csv_writer.writerow(row)
```

1.2.3 Further reformatting (for Content Rating and App Size)

[4]:
```
cleaned_file = io.open('cleaned.csv', newline='', encoding='utf-8')
csv_reader = csv.reader(cleaned_file)

output_file = io.open("cleaned2.csv", 'w', newline='', encoding='utf-8')
csv_writer = csv.writer(output_file)
for row in csv_reader:
    if row[7] == "4+":
        row[7] = "Everyone"
    if row[7] == "12+":
        row[7] = "Teen"
    if row[7] == "17+":
        row[7] = "Mature"
    if row[7] == "Mature 17+":
        row[7] = "Mature"
    if row[3] == "":
        row[3] = 0
    csv_writer.writerow(row)
```

[5]:
```
cleaned_file2 = io.open('cleaned2.csv', newline='', encoding='utf-8')
csv_reader2 = csv.reader(cleaned_file2)

output_file = io.open("cleaned3.csv", 'w', newline='', encoding='utf-8')
csv_writer = csv.writer(output_file)

for row in csv_reader2:
    if row[1] == 'Google Play Store':
        size = row[6]
        split_entry = list(size)
        size = size[:-1]
        if split_entry[-1] == "M":
            size = float(size) * 1000000
        elif split_entry[-1] == "k":
            size = float(size) * 1000
        elif split_entry[-1] == "G":
            size = float(size) * 1000000000
        row[6] = size
        csv_writer.writerow(row)
    else:
        csv_writer.writerow(row)
```

1.2.4 Final look of the datasets after cleaning

[6]:
```
df = pd.read_csv('cleaned3.csv')
df.head(3)
```

[6]:
```
   Unnamed: 0      Store                                                App  \
0           0  App Store                                     PAC-MAN Premium
1           1  App Store                           Evernote - stay organized
2           2  App Store  WeatherBug - Local Weather, Radar, Maps, Alerts

   Rating  Reviews Price       Size Content Rating        Genres  \
0     4.0    21292  3.99  100788224       Everyone         Games
1     4.0   161065   0.0  158578688       Everyone  Productivity
2     3.5   188583   0.0  100524032       Everyone       Weather

  Current Version
0           6.3.5
1           8.2.2
2           5.0.0
```

[7]:
```
df.tail(3)
```

[7]:
```
       Unnamed: 0              Store                       App  Rating  Reviews  \
16805       10790  Google Play Store  HipChat - beta version     4.1     1035
16806       10791  Google Play Store       Winter Wonderland     4.0     1287
16807       10792  Google Play Store  Soccer Clubs Logo Quiz     4.2    21661

      Price        Size Content Rating         Genres Current Version
16805     0  20000000.0       Everyone  Communication        3.20.001
16806     0  38000000.0       Everyone           Word             1.0
16807     0  16000000.0       Everyone         Trivia          1.3.81
```

1.3 III. Simple Analysis

Our investigation leads us to compare the user ratings in the Apple App Store to the Google Play Store. While App Store users most valued Productivity apps (as seen by its average rating), Google Play Store users rated Art and Design apps highest. Moreover, Facebook's mobile app had the most ratings across both mobile application stores.

The report also considers the file sizes of apps on both platforms. In the App Store, the largest app is "ROME: Total War" and the smallest app is "GraphModeling". For the Google Playstore, the largest app is "Stickman Legends: Shadow Wars" and the smallest is "Essential Resources". Notably, the App store had a much larger average app size with 199.13 MB compared to the Google Playstore's 17.82 MB.

Finally, the datasets revealed free apps were most common in the App Store, which concludes the simple analysis.

1.3.1 Largest and Smallest app in App Store dataset, average app size

[8]:
```
## App Store Sizes

cleaned = io.open('cleaned3.csv', newline='', encoding='utf-8')
csv_reader = csv.reader(cleaned)

largest_app = -1
smallest_app = 9**99
total_app_size = 0
total_apple_apps = 0

for row in csv_reader:
    if row[1] == "App Store":
        total_apple_apps += 1
        app_size = row[6]
        app_name = row[2]
        if float(app_size) >= float(largest_app):
            largest_app = app_size
            largest_app_name = app_name
        if float(app_size) <= float(smallest_app):
            smallest_app = app_size
            smallest_app_name = app_name
        total_app_size += float(app_size)

avg_app_size = total_app_size / (total_apple_apps) / 1000000

print('Apple App Store App Sizes:\n')
print('The largest app is "{}", with a size of {:.2f} MB.'.
 ↪format(largest_app_name, float(largest_app) / 1000000))
print('The smallest app is "{}", with a size of {:.2f} MB.'.
 ↪format(smallest_app_name, float(smallest_app) / 1000000))
print("The average app size is {} MB.".format(round(avg_app_size, 2)))
```

```
Apple App Store App Sizes:

The largest app is "ROME: Total War", with a size of 4025.97 MB.
The smallest app is "GraphModeling", with a size of 0.59 MB.
The average app size is 199.13 MB.
```

[9]:
```
## Google Play Store Sizes

cleaned2 = io.open('cleaned3.csv', newline='', encoding='utf-8')
csv_reader2 = csv.reader(cleaned2)

largest_app = -1
smallest_app = 9**99
```

```
total_app_size = 0
total_apple_apps = 0

for row in csv_reader2:
    if row[1] == "Google Play Store":
        total_apple_apps += 1
        app_size = row[6]
        app_name = row[2]
        if app_size != 'Varies with device' and app_size != 'Varies with devic':
            if float(app_size) >= float(largest_app):
                largest_app = app_size
                largest_app_name = app_name
            if float(app_size) <= float(smallest_app):
                smallest_app = app_size
                smallest_app_name = app_name
            total_app_size += float(app_size)

avg_app_size = total_app_size / (total_apple_apps) / 1000000

print('Google Play Store App Sizes:\n')
print('The largest app is "{}", with a size of {} MB.'.format(largest_app_name,␣
 ↪float(largest_app) / 1000000))
print('The smallest app is "{}", with a size of {} MB.'.
 ↪format(smallest_app_name, float(smallest_app) / 1000000))
print("The average app size is {} MB.".format(round(avg_app_size, 2)))
```

```
Google Play Store App Sizes:

The largest app is "Stickman Legends: Shadow Wars", with a size of 100.0 MB.
The smallest app is "Essential Resources", with a size of 0.0085 MB.
The average app size is 17.82 MB.
```

1.3.2 Most expensive app in App Store dataset, average price, most common app price

[10]:
```
cleaned = io.open('cleaned3.csv', newline='', encoding='utf-8')
csv_reader = csv.reader(cleaned)

most_expensive_app = -1
total_app_price = 0

app_prices = {}

for row in csv_reader:
    if row[1] == "App Store":
        app_price = row[5]
        app_name = row[2]
        if float(app_price) >= float(most_expensive_app):
```

```
            most_expensive_app = app_price
            most_expensive_app_name = app_name
        total_app_price += float(app_price)
        if app_price not in app_prices:
            app_prices[app_price] = 1
        elif app_price in app_prices:
            app_prices[app_price] += 1
avg_app_price = total_app_price / total_apple_apps

print('The most expensive app is "{}", with a price of ${} USD.'.
 ↪format(most_expensive_app_name, most_expensive_app))
print('The average appp price is ${} USD.\n'.format(round(avg_app_price, 2)))

print("The most 5 common app prices, and number of apps of that price:")
x = 0
for item in sorted(app_prices):
    x += 1
    if x <= 5:
        print("${}: {}".format(item, app_prices[item]))
```

```
The most expensive app is "LAMP Words For Life", with a price of $299.99 USD.
The average appp price is $1.29 USD.

The most 5 common app prices, and number of apps of that price:
$0.0: 4056
$0.99: 728
$1.99: 621
$11.99: 6
$12.99: 5
```

1.3.3 Most popular genres in the Apple App Store

[11]:
```
cleaned = io.open('cleaned3.csv', newline='', encoding='utf-8')
csv_reader = csv.reader(cleaned)

count_genre = {}
total_apple_apps = 0
total = 0

for row in csv_reader:
    if row[1] == "App Store":
        total_apple_apps += 1
        genre = row[8]
        #print(genre)
        if genre in count_genre:
            count_genre[genre] += 1
        elif genre not in count_genre:
```

```
            count_genre[genre] = 1
        total += 1

freq_genre = {}

for key in count_genre:
    freq_genre[key] = (count_genre[key] / total) * 100

table_display = []
for key in freq_genre:
    table_display.append((freq_genre[key], key))

table_sorted = sorted(table_display, reverse = True)

print("The 5 most popular categories in Apple App Store and their percentages␣
 ↪are:\n")
i = 0
while i < 5:
    print("{}: {:.2f}%".format(table_sorted[i][1], table_sorted[i][0]))
    i += 1
```

```
The 5 most popular categories in Apple App Store and their percentages are:

Games: 53.66%
Entertainment: 7.43%
Education: 6.29%
Photo & Video: 4.85%
Utilities: 3.45%
```

1.3.4 Top 5 highest rated app categories in both Apple App Store and Google Play Store

[12]:
```
df = pd.read_csv('cleaned3.csv')
store = df['Store']
apple = (store == "App Store")
apple_data = df[apple]

apple_genre = apple_data.groupby('Genres')
genre_user = apple_genre['Rating'].mean().to_dict()

table_genre_rating = []
for key in genre_user:
    table_genre_rating.append((genre_user[key], key))

table_sorted = sorted(table_genre_rating, reverse = True)

print("The 5 highest rated app categories in Apple App Store are:\n")
i = 0
```

```
while i < 5:
    print("{}: {:.2f}".format(table_sorted[i][1], table_sorted[i][0]))
    i += 1
```

```
The 5 highest rated app categories in Apple App Store are:

Productivity: 4.01
Music: 3.98
Photo & Video: 3.80
Business: 3.75
Health & Fitness: 3.70
```

[13]:
```
df = pd.read_csv('cleaned3.csv')
google = (store == "Google Play Store")
google_data = df[google]


google_genre = google_data.groupby('Genres')

google_genre_user = google_genre['Rating'].mean(numeric_only = True).to_dict()

google_genre_rating = []
for key in google_genre_user:
    google_genre_rating.append((google_genre_user[key], key))

table_sorted = sorted(google_genre_rating, reverse = True)

print("The 5 highest rated app categories in Google Play Store are:\n")
i = 0
for item in table_sorted:
    if i <= 5:
        if ";" not in item[1]:
            print("{}: {:.2f}".format(item[1], item[0]))
            i += 1
```

```
The 5 highest rated app categories in Google Play Store are:

Art & Design: 4.21
Word: 4.15
Role Playing: 4.15
Action: 4.14
Casino: 4.07
Adventure: 4.06
```

1.3.5 Top 5 apps with the most rating counts in both Apple App Store and Google Play Store

[14]:
```
cleaned = io.open('cleaned3.csv', newline='', encoding='utf-8')
csv_reader = csv.reader(cleaned)

most_rated_app = -1
total_app_ratings = 0
total_apple_apps = 0
total_google_apps = len(google_data)

rate = []
for row in csv_reader:
    if row[1] == "App Store":
        total_google_apps += 1
        app_rate = row[4]
        app_name = row[2]

        rate.append((float(app_rate), app_name))

        if float(app_rate) >= float(most_rated_app):
            most_rated_app = app_rate
            most_rated_app_name = app_name
        total_app_ratings += float(app_rate)

avg_app_rate = total_app_ratings / total_google_apps

print("Top 5 apps with the most ratings on the App Store:\n")
i = 0
for x in sorted(rate, reverse = True):
    if i < 5:
        print("{}: {} ratings".format(x[1], int(x[0])))
        i += 1

print()
print("Average number of ratings for each app in Apple App Store: {:.2f} ".
 ↪format(avg_app_rate))
```

```
Top 5 apps with the most ratings on the App Store:

Facebook: 2974676 ratings
Instagram: 2161558 ratings
Clash of Clans: 2130805 ratings
Temple Run: 1724546 ratings
Pandora - Music & Radio: 1126879 ratings

Average number of ratings for each app in Apple App Store: 5520.60
```

[15]:
```
cleaned = io.open('cleaned3.csv', newline='', encoding='utf-8')
csv_reader = csv.reader(cleaned)

most_rated_app = -1
total_app_ratings = 0
total_google_apps = 0

rate = []
for row in csv_reader:
    if row[1] == "Google Play Store":
        total_google_apps += 1
        app_rate = row[4]
        app_name = row[2]

        rate.append((float(app_rate), app_name))

        if float(app_rate) >= float(most_rated_app):
            most_rated_app = app_rate
            most_rated_app_name = app_name
        total_app_ratings += float(app_rate)

avg_app_rate = total_app_ratings / total_google_apps

print("Top 5 apps with the most ratings on the Google Play Store:\n")
i = 0
for x in sorted(rate, reverse = True):
    if i < 5:
        print("{}: {} ratings".format(x[1], int(x[0])))
        i += 1

print()
print("Average number of ratings for each app in Google Play Store: {:.2f} ".
 ↪format(avg_app_rate))
```

```
Top 5 apps with the most ratings on the Google Play Store:

Facebook: 78158306 ratings
WhatsApp Messenger: 69119316 ratings
Instagram: 66577313 ratings
Messenger  Text and Video Chat for Free: 56642847 ratings
Clash of Clans: 44891723 ratings

Average number of ratings for each app in Google Play Store: 217563.12
```