IST652 Final Project Report

Group Members: Yuang Huang | Lvwei Chen | Huiyu Li

Background

As well-known, mobile applications have played a significant role in making life easier, influencing people's life and work behaviors and entertaining. With fast development, there are thousands of applications on different platforms. However, defining what types of applications are popular from the user side can help businesses and developers capture the
market. Meanwhile, the Google play store apps data has enormous potential to drive apps making businesses to success.

Dataset

The dataset for the final project is about Google Play store apps (https://www.kaggle.com/datasets/lava18/google-play-store-apps) from the Kaggle website. There are two datasets related to the project. One of the datasets records details of applications such as category, size, rating, and installations with 10841 rows and 13 columns. Another dataset contains App reviews from users with 64295 rows and 5 columns. There will be enough data for the final project of supporting analysis of the datasets to find out interesting stories and insights.

Library Sources

For this project, we will mainly deal with two kinds of data for analyzing Google Play Store App, which are structured data and semi-structured data. For processing the structure data, the libraries used to this project are Pandas, NumPy, Seaborn, Matplotlib, Datetime. In contrast, for processing the semi-structure data, the libraries used to this project are NLTK, NLTK.tokenize, Word cloud.

Data Preprocessing

Data Cleaning

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up

This figure shows the original shape of the application dataset. Before cleaning the dataset, we need to go through possible problems may exist in the dataset. Therefore, in data cleaning, we will focus on delete the duplicated rows and will remove the NA entries. Besides, we will rename column names and convert the possible columns to numeric entries and datetime type, helps us easier to analyze the data and extract interesting insights.

After renamed columns contained blank spaces and added two new columns which are last updated year and month “Up_year” and “Up_month”, we checked the data types of each column in the datasets to figure out which columns need to do data transformation to make data useful and meaningful.

```
print (app_data.dtypes)
```

```
App                object
Category           object
Rating            float64
Reviews            object
Size               object
Installs           object
Type               object
Price              object
Content_rating     object
Genres             object
Last_updated       object
Current_ver        object
Android_ver        object
dtype: object
```

Figure1

```
print (app_data.dtypes)
```

```
App                        object
Category                   object
Rating                    float64
Reviews                   float64
Size                       object
Installs                    int64
Type                       object
Price                     float64
Content_rating             object
Genres                     object
Last_updated               object
Current_ver                object
Android_ver                object
LastUpdates        datetime64[ns]
Up_year                     int64
Up_month                    int64
dtype: object
```

Figure 2

Figure 1 is the data types in the dataset before data cleaning, and figure 2 is the data types that we transformed. It’s clearly shown that the data become to clearer than before data transformation. Especially we transformed ‘Last_updated’ from object to datetime to track when the application lastly updated and categories of application compositions on the Google App Play platform may change different by each year.

Moreover, we transformed ‘Reviews’, ‘Installs’, and ‘Price’ from string to float to calculate the number of reviews, installations, and price by several dimensions to track the variety and differences.

Last but not least, It’s always important in data processing to check NAs in the dataset. Figure 3 shows NAs in each column before data cleaning, and figure 4 shows a completely cleaned dataset. Based on figure 3, we can clearly see from the data that the rating contains a large number of missing data, and direct deletion will lead to insufficient data, so we calculate the Rating based on mean value and then we drop the rest of missing values.

```
# missing data
app_data.isna().sum()
```

```
App                0
Category           0
Rating          1474
Reviews            0
Size               0
Installs           0
Type               1
Price              0
Content_rating     0
Genres             0
Last_updated       0
Current_ver        8
Android_ver        2
LastUpdates        0
Up_year            0
Up_month           0
dtype: int64
```

Figure 3

```
#checking NAs
app_data.isna().sum()
```

```
App                0
Category           0
Rating             0
Reviews            0
Size               0
Installs           0
Type               0
Price              0
Content_rating     0
Genres             0
Last_updated       0
Current_ver        0
Android_ver        0
LastUpdates        0
Up_year            0
Up_month           0
dtype: int64
```

Figure 4

```
In [12]: #calculating the RATING based on mean value
         app_data['Rating'].fillna((app_data['Rating'].mean()), inplace=True)
```

In addition, we notice that the values of several data columns "set", "Price" and "Size" contain special characters. To enable them to be numericized, we do the following:

1. Remove '+' from 'Number of Installs' to make it numeric:

```
# Remove '+' from 'Number of Installs' to make it numeric
app_data['Installs'] = app_data['Installs'].apply(lambda x: x.replace('+', '') if '+' in str(x) else x)
# While converting the column into float,we have to remove ',' too
app_data['Installs'] = app_data['Installs'].apply(lambda x: x.replace(',', '') if ','in str(x) else x)
app_data["Installs"] = pd.to_numeric(app_data["Installs"])
```

2. Remove "$" from the data entries in the column "Price" so that it can be converted to float:

```
app_data['Price'] = app_data['Price'].apply(lambda x: str(x).replace('$','') if '$' in str(x) else str(x))
# Convert the data in "Price" to float
app_data["Price"]= app_data.Price.astype(float)
```

3. Remove the "M" which is the Mb for the size, and replace k and change the unit to Mb (1024k=1Mb):

```
# removing the "m" which is the mb for the size
app_data.Size= app_data["Size"].str.replace("M", "")
app_data.Size= app_data['Size'].str.replace("Varies with device","-1")
# Here we replace k and change the unit to Mb
app_data['Size']=app_data['Size'].apply(lambda x: str(round((float(x.rstrip('k'))/1024),1) )if x[-1]=='k' else x)
```

The cleaned data and the Correlation Heatmap of the data:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19	10000	Free	0.0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14	500000	Free	0.0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5000000	Free	0.0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25	50000000	Free	0.0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8	100000	Free	0.0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up

Structured Data Analysis

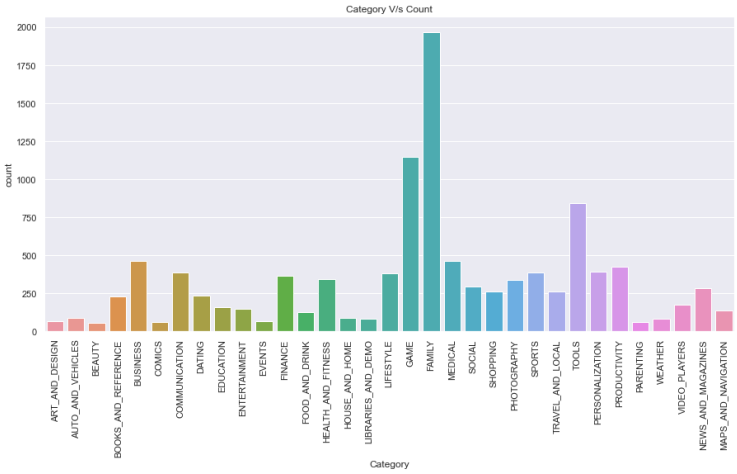
Analysis Tasks List:

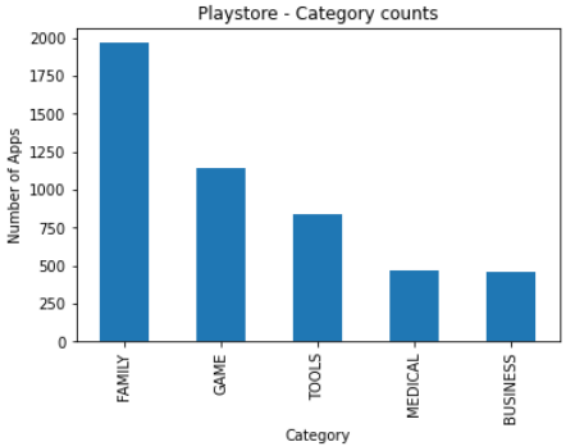
1. **Most Popular Categories**
2. **Applications with the Largest Size**
3. **Applications with the Largest Installations**
4. **Most Installed and Rated Applications**

   Most Installed and Reviewed Applications
5. **Content Rating Analysis**
6. **Price Analysis**

Most Popular Categories

There is a total of 33 application categories in the dataset, the top five categories with respect to value count of applications are Family, Game, Tools, Medical, and Business throughout years.





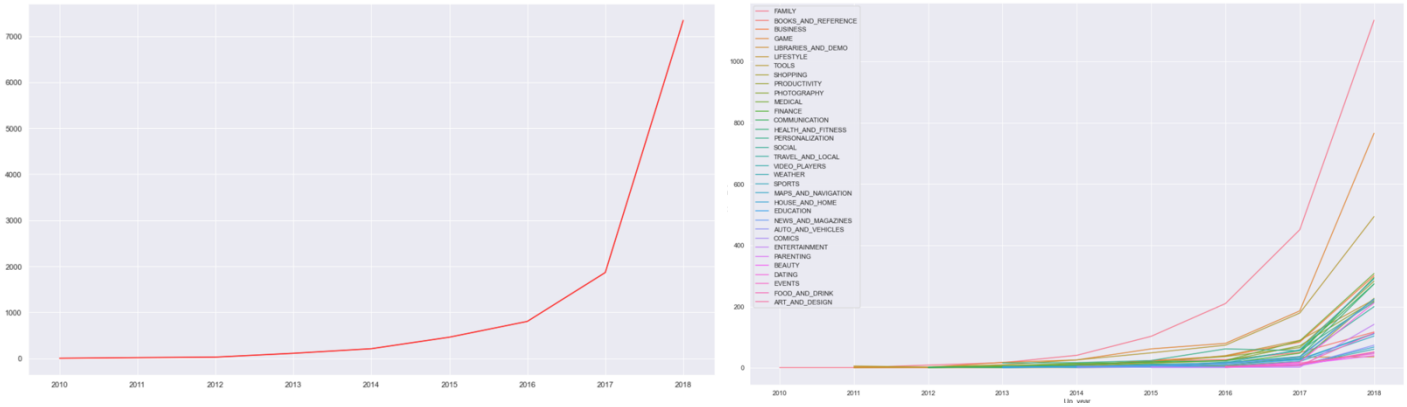
Getting the measures of central tendency for all the installation grouped by "Category":

Category	min	mean	median	max
ART_AND_DESIGN	0	1.942705e+06	100000.0	50000000
AUTO_AND_VEHICLES	1	6.250613e+05	100000.0	10000000
BEAUTY	50	5.131519e+05	50000.0	10000000
BOOKS_AND_REFERENCE	1	8.354211e+06	10000.0	1000000000
BUSINESS	0	2.178076e+06	1000.0	100000000
COMICS	50	9.347692e+05	100000.0	10000000
COMMUNICATION	1	8.435989e+07	1000000.0	1000000000
DATING	1	1.129533e+06	100000.0	10000000
EDUCATION	1000	5.586231e+06	1000000.0	100000000
ENTERTAINMENT	10000	1.925611e+07	5000000.0	1000000000
EVENTS	1	2.495806e+05	1000.0	5000000
FAMILY	0	5.212502e+06	100000.0	1000000000
FINANCE	0	2.395215e+06	50000.0	100000000
FOOD_AND_DRINK	1	2.156683e+06	500000.0	10000000
GAME	1	3.066960e+07	1000000.0	1000000000
HEALTH_AND_FITNESS	1	4.642441e+06	500000.0	500000000
HOUSE_AND_HOME	1	1.917187e+06	500000.0	10000000
LIBRARIES_AND_DEMO	10	7.380465e+05	10000.0	10000000
LIFESTYLE	0	1.407444e+06	10000.0	100000000
MAPS_AND_NAVIGATION	10	5.286729e+06	100000.0	100000000
MEDICAL	0	1.150269e+05	1000.0	5000000
NEWS_AND_MAGAZINES	0	2.648876e+07	100000.0	1000000000
PARENTING	10	5.253518e+05	100000.0	10000000
PERSONALIZATION	0	5.962779e+06	10000.0	100000000
PHOTOGRAPHY	5	3.011417e+07	5000000.0	1000000000

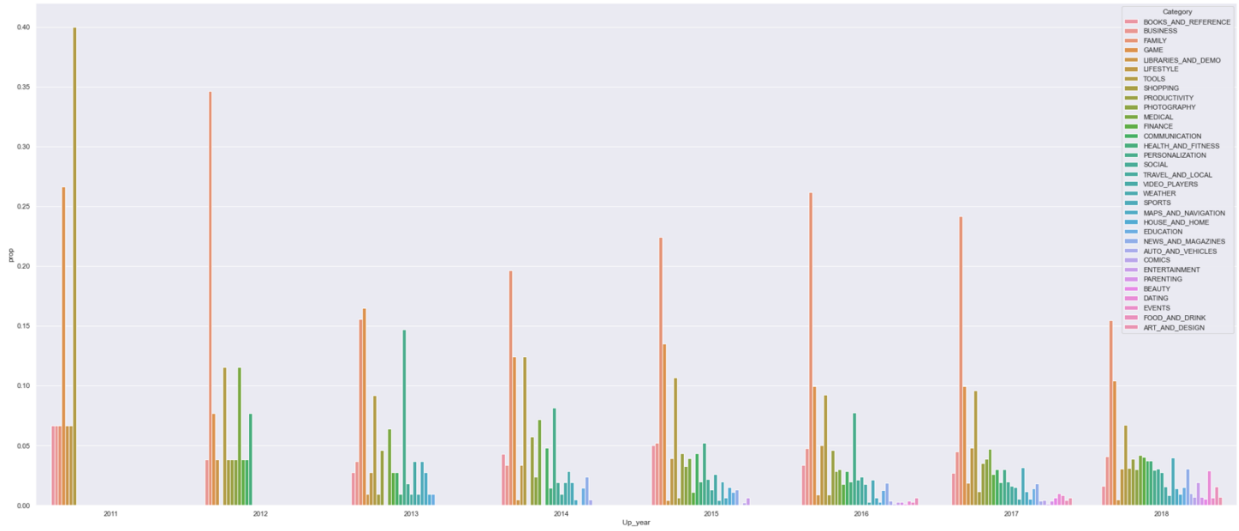
What interesting here, we found out that Education and Entertainment applications are having specific users that the minimum of installations are extremely high. In all categories, it shows Family applications

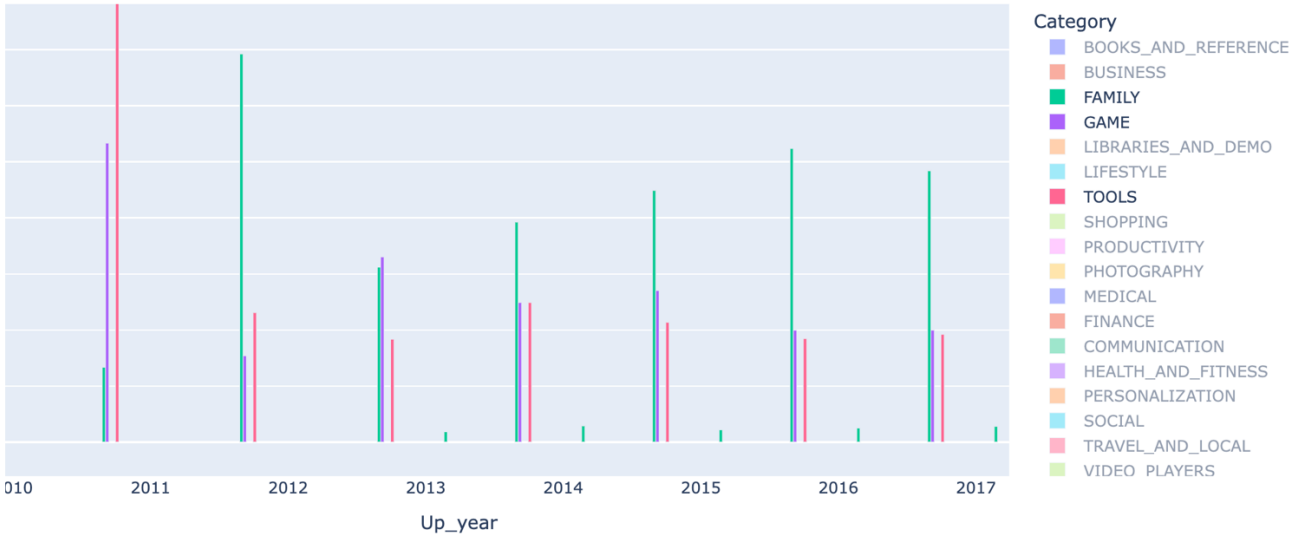
occupied the first place of installations. However, how each category of applications published on Google Play Store through years and how varied by year.

From these two time-series graphs, it shows year and applications updated quantity on Google Play Store are positively correlated as year increases applications on the platform updated increases especially in from 2016 to 2018.



The bar plot shows that categories of updated applications by year between 2011 to 2018 to figure out proportion of categories of applications updated in each year. Here we decided to drop year 2010, because there was only one category application which will cause 100% proportion in 2010. The main point from this graph to compare how different among different categories of applications updated their app on Google Play Store platform by year. It demonstrates that from 2012 to 2018, family applications were more likely to update their applications each year. Game applications were on second place by year. It also seems like the Education applications were the least to update their applications through years.





App with the largest size

We will simply sort our dataset to find out, which app has the biggest size.

```
cols = ["App", "Size"]
size_data = app_data[cols]
size_data = size_data.sort_values(by="Size")
size_data
```

	App	Size
3596	Family GPS Tracker and Chat + Baby Monitor Online	-1.0
5647	Audiobooks from Audible	-1.0
1467	realestate.com.au - Buy, Rent & Sell Property	-1.0
1466	Domofond Real Estate. Buy, rent an apartment.	-1.0
1464	Trulia Rent Apartments & Homes	-1.0
...	...	...
5858	Miami crime simulator	100.0
1563	Talking Babsy Baby: Baby Games	100.0
8400	Car Crash III Beam DH Real Damage Simulator 2018	100.0
8838	Draft Simulator for FUT 18	100.0
1791	Mini Golf King - Multiplayer Game	100.0

The result of App with the largest size

We notice that there may be multiple apps with the same Size, so we find the app with the maximum Size:

```
In [15]: max_inst = size_data[size_data["Size"] == size_data.Size.max()].App.to_list()
         max_inst
```

Get the following apps: 'Hungry Shark Evolution', 'The Walking Dead: Our World', 'Gangster Town: Vice District', 'SimCity BuildIt', 'Ultimate Tennis', 'Hungry Shark Evolution', 'Navi Radiography Pro', 'Hungry Shark Evolution', 'Post Bank', 'Vi Trainer', 'Stickman Legends: Shadow Wars', 'Miami crime simulator', 'Talking Baby: Baby Games', 'Car Crash III Beam DH Real Damage Simulator 2018', 'Draft Simulator for

FUT 18', 'Mini Golf King - Multiplayer Game'

App with the largest num of installs

First, we sort the column of the mount Installs:

```
In [19]: installs_data = installs_data.sort_values("Installs")
         installs_data
```

Out[19]:

	App	Installs
9709	EP Cook Book	0
8072	CX Network	0
5303	Ak Parti Yardım Toplama	0
8605	Sweden Newspapers	0
9327	EG | Explore Folegandros	0
...	...	...
4168	Google Drive	1000000000
5391	Google Photos	1000000000
3902	WhatsApp Messenger	1000000000
340	Hangouts	1000000000
3563	Google Drive	1000000000

As we can see, there are many apps with almost the same amount, so we found all apps that have the same number of installs as the maximum:

```
In [20]: result = installs_data[installs_data["Installs"] == installs_data.Installs.max()]
         result
```

	App	Installs
3115	Maps - Navigate & Explore	1000000000
3941	Facebook	1000000000
3521	Google Drive	1000000000
3125	Google Street View	1000000000
3994	Google Chrome: Fast & Secure	1000000000
2542	Facebook	1000000000
2851	Google Photos	1000000000
9834	Google News	1000000000
1698	Subway Surfers	1000000000
3926	YouTube	1000000000
151	Google Play Books	1000000000
4102	Messenger – Text and Video Chat for Free	1000000000
2882	Google Photos	1000000000
4096	Maps - Navigate & Explore	1000000000
4094	Gmail	1000000000
5852	Google Play Games	1000000000
1652	Subway Surfers	1000000000
2543	Instagram	1000000000
3663	YouTube	1000000000
3907	Instagram	1000000000
2552	Google+	1000000000
3814	Google News	1000000000
380	WhatsApp Messenger	1000000000
381	Messenger – Text and Video Chat for Free	1000000000
3763	Google News	1000000000
385	Hangouts	1000000000
2602	Instagram	1000000000

As we can see, there are several duplicates. So, we got rid of duplicate data items by using the drop_duplicates () method and get the result.

The result of App with the largest num of installs

```
3115                  Maps - Navigate & Explore
3941                                   Facebook
3521                              Google Drive
3125                         Google Street View
3994               Google Chrome: Fast & Secure
2851                             Google Photos
9834                               Google News
1698                             Subway Surfers
3926                                   YouTube
151                           Google Play Books
4102    Messenger – Text and Video Chat for Free
4094                                     Gmail
5852                           Google Play Games
2543                                 Instagram
2552                                   Google+
380                          WhatsApp Messenger
385                                  Hangouts
390                Skype - free IM & video calls
3685             Google Play Movies & TV
3232                                   Google
```

Most installed and most rated apps and most installed and reviewed apps

First, we sort the data using two fields "Installs" and "Rating":

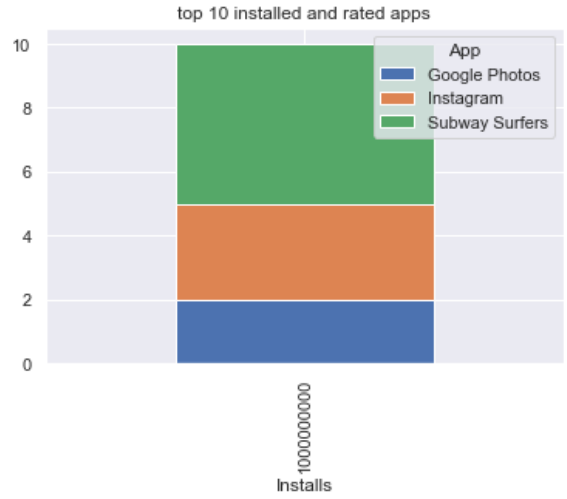
In [22]:
```
top_installed_and_rated_apps = app_data.sort_values(by=["Installs", "Rating"], ascending=False)
top_installed_and_rated_apps.head() # main top apps
```

Out[22]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
1652	Subway Surfers	GAME	4.5	27722264.0	76.0	1000000000	Free	0.0	Everyone 10+	Arcade	July 12, 2018	1.90.0	4.1 and up
1698	Subway Surfers	GAME	4.5	27723193.0	76.0	1000000000	Free	0.0	Everyone 10+	Arcade	July 12, 2018	1.90.0	4.1 and up
1748	Subway Surfers	GAME	4.5	27724094.0	76.0	1000000000	Free	0.0	Everyone 10+	Arcade	July 12, 2018	1.90.0	4.1 and up
1870	Subway Surfers	GAME	4.5	27725352.0	76.0	1000000000	Free	0.0	Everyone 10+	Arcade	July 12, 2018	1.90.0	4.1 and up
1915	Subway Surfers	GAME	4.5	27725352.0	76.0	1000000000	Free	0.0	Everyone 10+	Arcade	July 12, 2018	1.90.0	4.1 and up

Next, we sliced the top 10 apps that are installed and rating.

In [24]:
```
top_10_installed_and_rated_apps= top_installed_and_rated_apps.head(10)
app1= top_10_installed_and_rated_apps
top_apps=app1.groupby(["Installs", "App"]).size().unstack()
top_apps.plot(kind="bar",stacked=True)
plt.title('top 10 installed and rated apps')
ax=plt.gca()
plt.show()
```



What's more, we sorted the data using two fields "Installs" and " Reviews":

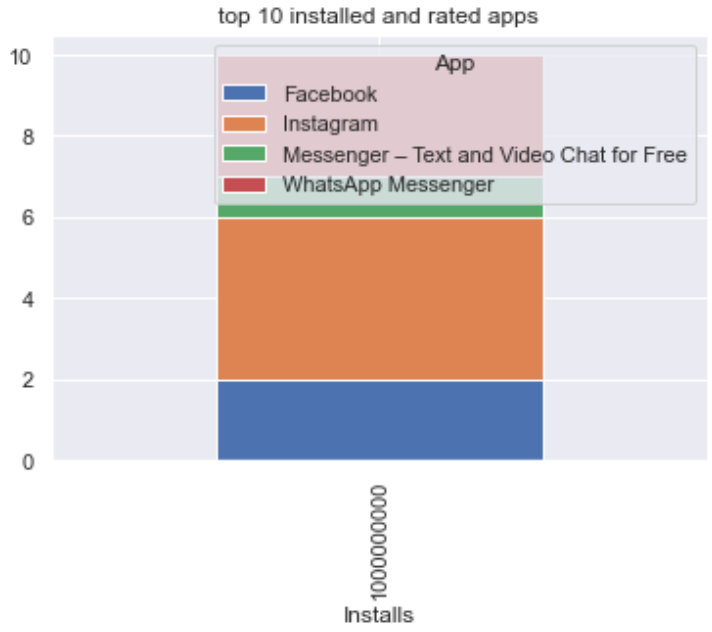
```
In [23]: top_installed_and_reviewed_apps = app_data.sort_values(by=["Installs", "Reviews"], ascending=False)
         top_installed_and_reviewed_apps.head()
```

Out[23]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
2542	Facebook	SOCIAL	4.1	78158306.0	-1.0	1000000000	Free	0.0	Teen	Social	August 3, 2018	Varies with device	Varies with device
3941	Facebook	SOCIAL	4.1	78128208.0	-1.0	1000000000	Free	0.0	Teen	Social	August 3, 2018	Varies with device	Varies with device
335	WhatsApp Messenger	COMMUNICATION	4.4	69119316.0	-1.0	1000000000	Free	0.0	Everyone	Communication	August 3, 2018	Varies with device	Varies with device
380	WhatsApp Messenger	COMMUNICATION	4.4	69119316.0	-1.0	1000000000	Free	0.0	Everyone	Communication	August 3, 2018	Varies with device	Varies with device
3902	WhatsApp Messenger	COMMUNICATION	4.4	69109672.0	-1.0	1000000000	Free	0.0	Everyone	Communication	August 3, 2018	Varies with device	Varies with device

We got the top 10 apps that are Installed and Reviews.

```
In [25]: top_10_installed_and_reviewed_apps = top_installed_and_reviewed_apps.head(10)
         app2= top_10_installed_and_reviewed_apps
         top_apps=app2.groupby(["Installs", "App"]).size().unstack()
         top_apps.plot(kind="bar",stacked=True)
         plt.title('top 10 installed and rated apps')
         ax=plt.gca()
         plt.show()
```

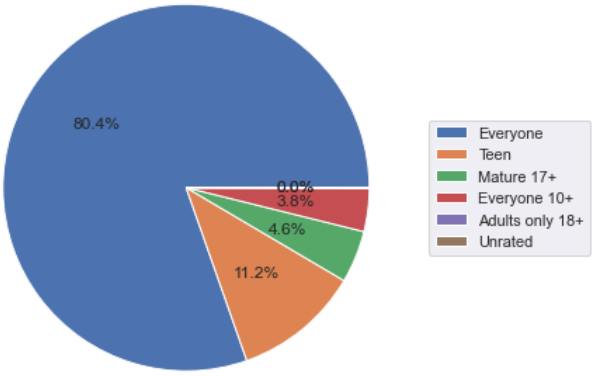


Content Rating Analysis

Firstly, we count for content rating in each of the category:

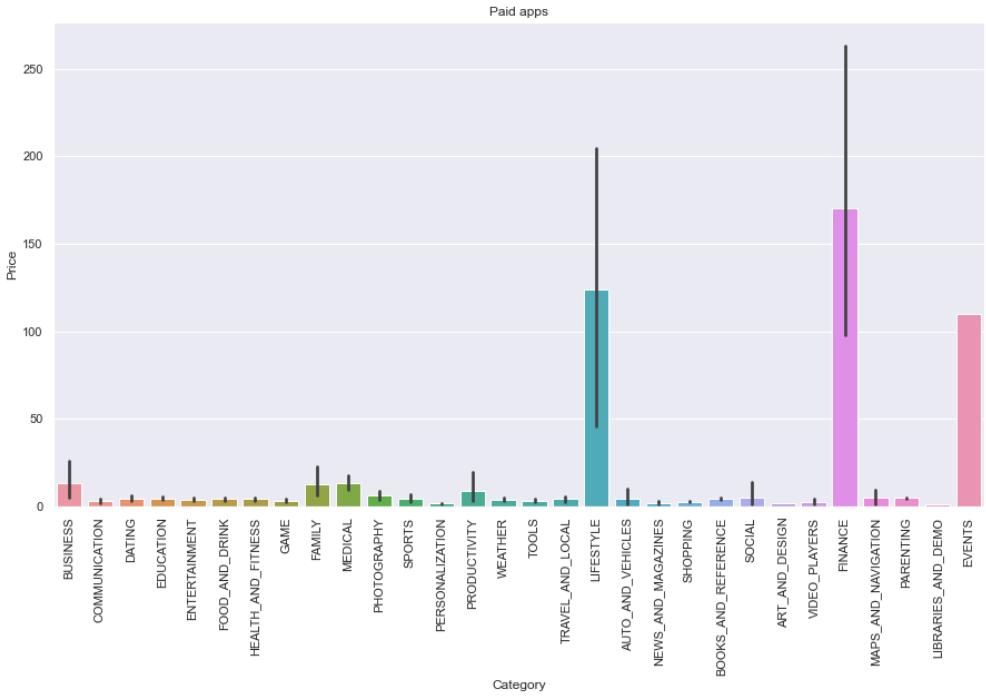
```
In [26]: app_data["Content Rating"].value_counts(ascending=False)

Out[26]: Everyone           8704
         Teen               1208
         Mature 17+          499
         Everyone 10+        413
         Adults only 18+       3
         Unrated               2
         Name: Content Rating, dtype: int64
```

Basic pie chart to view distribution of apps across various categories:



Price Analysis

In the previous analysis, we noticed that the "Family" had a lot of apps but didn't have the highest number of installs. Price may be a factor. We visualize this in the form of a simple bar graph.

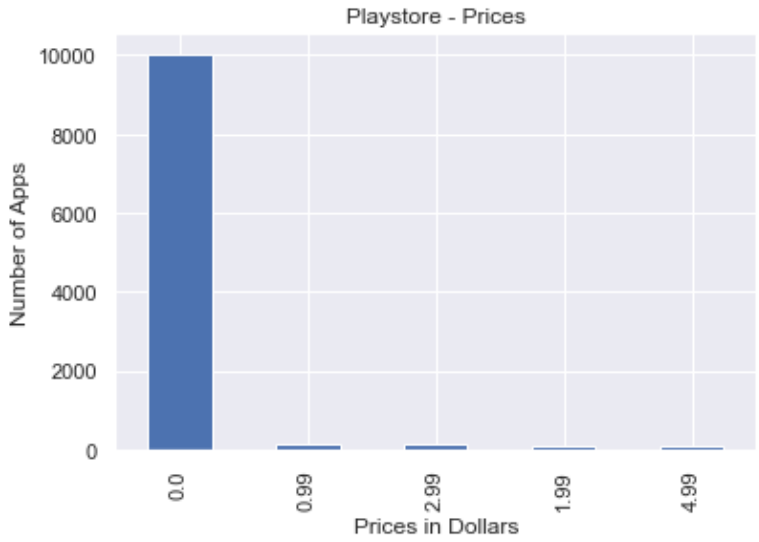


Above plot shows that the apps in the category Lifestyle and Finance have the highest prices. Here we look for the top five prices and the number of apps.

```
app_data["Price"].value_counts().nlargest(5).sort_values(ascending=False)
```

```
Out[30]: 0.00    10032
         0.99      146
         2.99      129
         1.99       73
         4.99       72
         Name: Price, dtype: int64
```

Now we visualize it by using bar plot, this plot shows that data has more than 90% apps that are free.

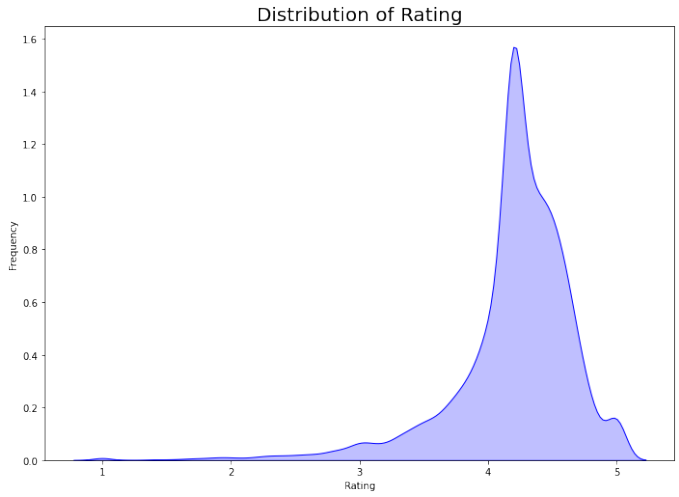


Rating on Play Store

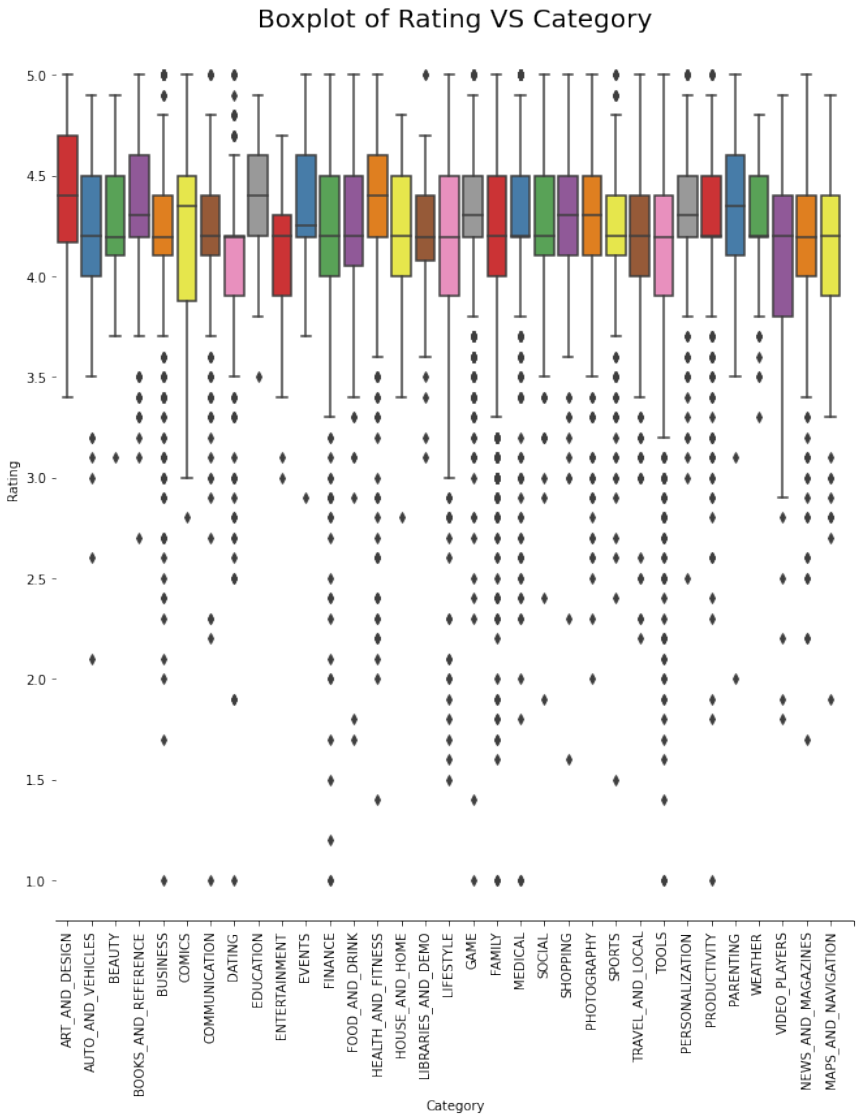
The Rating describe:

```
In  [8]: app_data['Rating'].describe()

Out[8]: count    10829.000000
        mean         4.192041
        std          0.479038
        min          1.000000
        25%          4.100000
        50%          4.200000
        75%          4.500000
        max          5.000000
        Name: Rating, dtype: float64
```

We visualized the distribution of rating; it can be clearly seen from the picture that the average of rating of application in store is around 4 which is very high.



According to the plot below, it illustrates that rating of application in each category does not show obvious differences.



Semi-Structured Data Analysis

We analyze data sets of application reviews from users:

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity
0	10 Best Foods for You	I like eat delicious food. That's I'm cooking ...	Positive	1.00	0.533333
1	10 Best Foods for You	This help eating healthy exercise regular basis	Positive	0.25	0.288462
2	10 Best Foods for You	NaN	NaN	NaN	NaN
3	10 Best Foods for You	Works great especially going grocery store	Positive	0.40	0.875000
4	10 Best Foods for You	Best idea us	Positive	1.00	0.300000

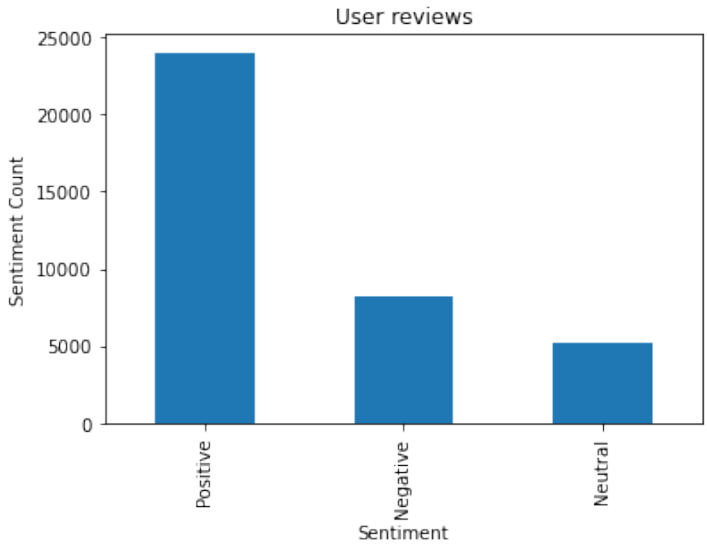
Firstly, we dropping the rows corresponding to missing values, and then we get the number of unique categories in the dataset.

```
In [3]: # the number of unique categories in the dataset
        reviews_data.App.nunique()

Out[3]: 865
```

For the column sentiment, we find the count of each sentiment in the dataset：

```
In [6]: # For the column sentiment, we find the count of each sentiment in the dataset
        reviews_data.Sentiment.value_counts()

Out[6]: Positive    23998
        Negative     8271
        Neutral      5158
        Name: Sentiment, dtype: int64
```

The bar plot of the sentiment count shows that more than 60% of the sentiment is positive.



Each app has different sentiments. Thus, we grouped by category:

App	Sentiment_Polarity	Sentiment_Subjectivity
10 Best Foods for You	0.470733	0.495455
104 找工作 - 找工作 找打工 找兼職 履歷健檢 履歷診療室	0.392405	0.545516
11st	0.185943	0.455340
1800 Contacts - Lens Store	0.318145	0.591098
1LINE – One Line with One Touch	0.196290	0.557315
...	...	...
Hotels.com: Book Hotel Rooms & Find Vacation Deals	0.101622	0.545444
Hotspot Shield Free VPN Proxy & Wi-Fi Security	0.251765	0.393284
Hotstar	0.038178	0.493964
Hotwire Hotel & Car Rental App	0.187029	0.459717
Housing-Real Estate & Property	-0.021427	0.378532

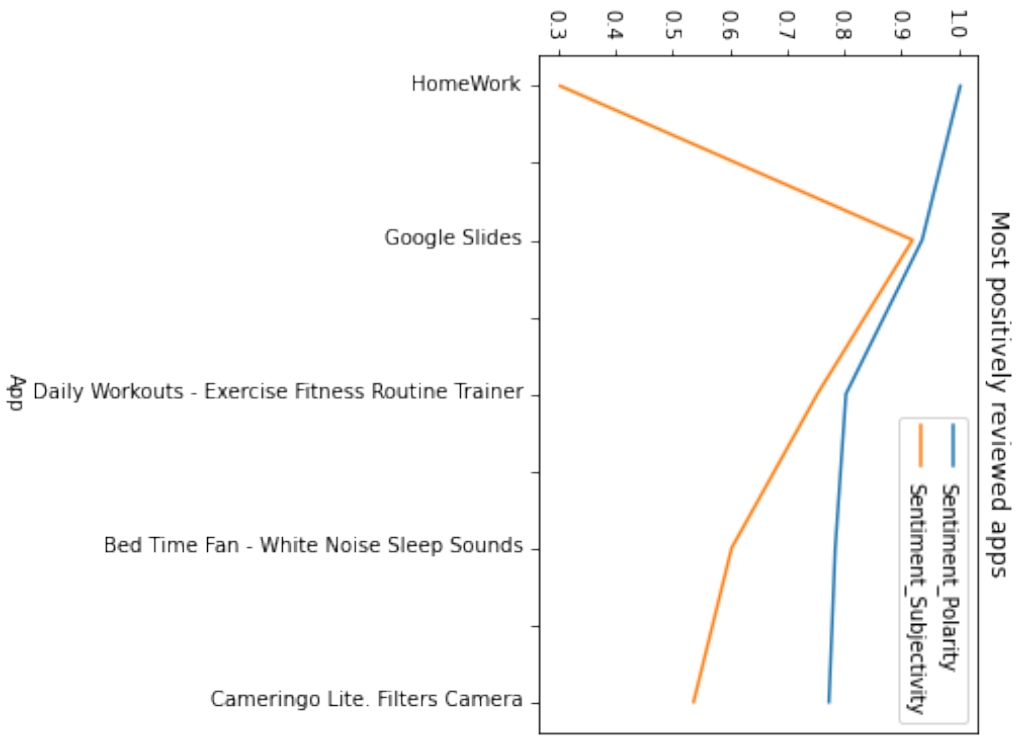
Reading maximum value of each of sentiment, sentiment polarity and sentiment subjectivity for each app:

App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity
10 Best Foods for You	nice super get	Positive	1.000000	1.00
104 找工作 - 找工作 找打工 找兼職 履歷健檢 履歷診療室	nice	Positive	0.910000	1.00
11st	good good	Positive	1.000000	1.00
1800 Contacts - Lens Store	great	Positive	0.838542	1.00
1LINE – One Line with One Touch	yoko	Positive	1.000000	1.00
...	...	...	...	...
Hotels.com: Book Hotel Rooms & Find Vacation Deals	would actually load anything. I tried uninstal...	Positive	0.700000	0.95
Hotspot Shield Free VPN Proxy & Wi-Fi Security	ɞmoƨ naffgggg	Positive	1.000000	1.00
Hotstar	running	Positive	1.000000	1.00
Hotwire Hotel & Car Rental App	Works better	Positive	0.800000	0.90
Housing-Real Estate & Property	Worst app. We get nothing Time waste . They up...	Positive	0.800000	1.00

We sort the data corresponding to mean of the data in accordance with sentiment polarity and sentiment subjectivity. And we look for the top positively user reviewed apps.

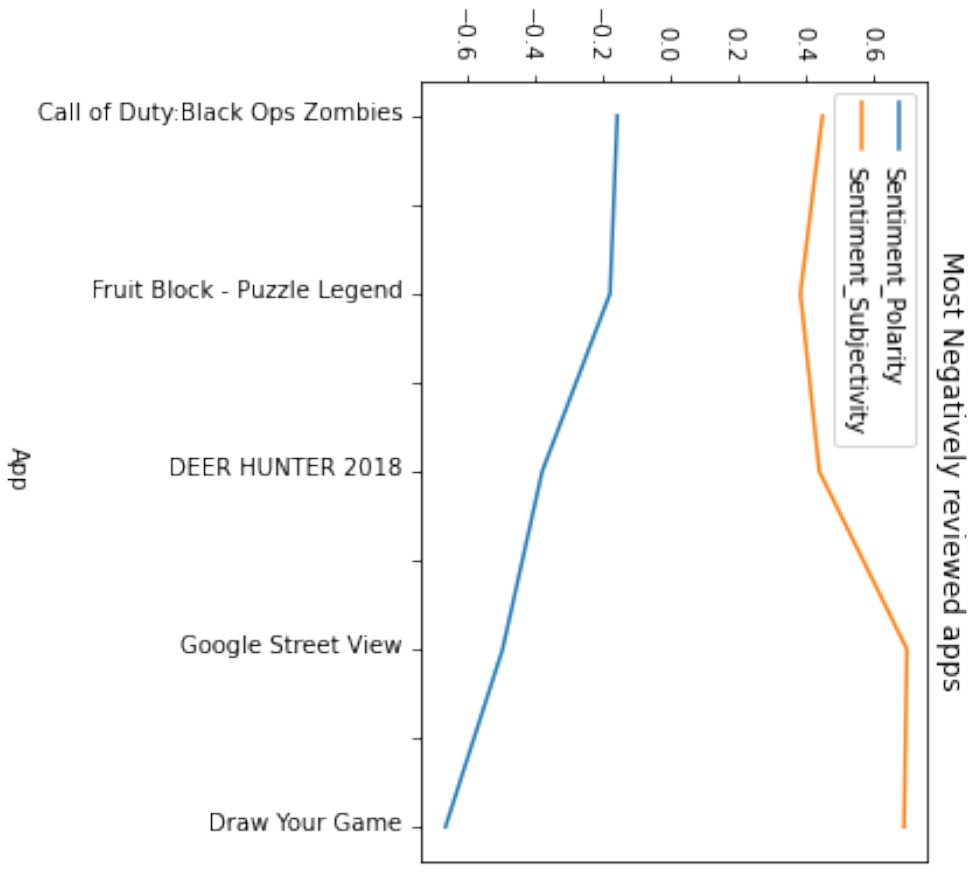
App	Sentiment_Polarity	Sentiment_Subjectivity
HomeWork	1.000000	0.300000
Google Slides	0.933333	0.916667
Daily Workouts - Exercise Fitness Routine Trainer	0.800000	0.750000
Bed Time Fan - White Noise Sleep Sounds	0.781250	0.600000
Cameringo Lite. Filters Camera	0.770269	0.533333

We plot the line graph for the above result:



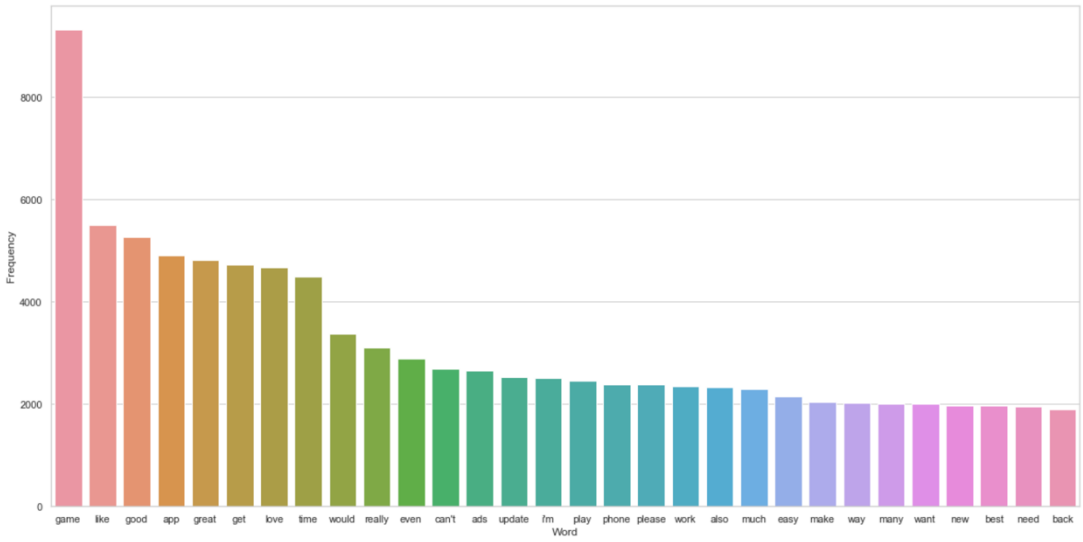
Here we look for the most negatively reviewed apps:

App	Sentiment_Polarity	Sentiment_Subjectivity
Call of Duty:Black Ops Zombies	-0.162120	0.442716
Fruit Block - Puzzle Legend	-0.183333	0.377778
DEER HUNTER 2018	-0.383333	0.433333
Google Street View	-0.500000	0.691667
Draw Your Game	-0.668490	0.683333

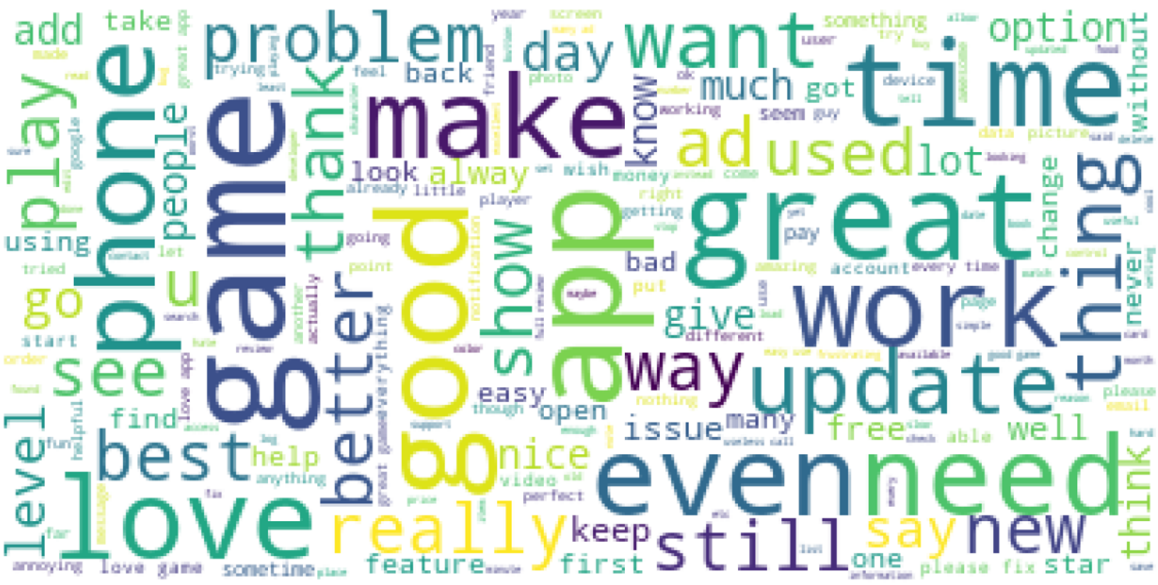


Word Cloud

We added a word cloud to figure what are the most common words in users' reviews on all applications on Good Play Store. Therefore, we might extract some insights based on the most common words people were talking about. No surprised, Game applications on Google Play Store were the most common word "game" that users mentioned. More importantly, when go through the most top50 common words, the top10words are mostly positive words, which means applications on Google Play Store were satisfied most users in different perspectives.



Here is the Top50 common words in a word cloud visualization, as words with larger size means more users mentioned or cared about in the reviews.



Summary

Collected data from the Google Play Store for our research. From our exploratory data analysis, we found several issues with the Google Play Store. If users are unhappy with app update, they can communicate with the developers so that low-install apps will not fall flat. There is also a smaller-scale rating mismatch problem. If this problem can be achieved, the Google Play Store will be able to more accurately reflect user sentiment which helps developers adjust accordingly.

Users prefer paying for lightweight apps. As a result, bulky paid apps may not perform well in the market. The optimal size for most top-ranked apps is between 2MB and 40MB — neither light nor heavy. Medical and home apps are most expensive. Users are more likely to download an app with more reviews.

Insights from word cloud visualization, as we mentioned, game applications are the most users mentioned and cared. Therefore, the developers of game applications could increase frequency of update on version of their applications to give more new content or inspiration for their users. Besides, they can also learn from their users' sentiment analysis result and most common words that they cared about to make improvement on what the users dislike to increase user retention rate.

Recommendations for Google Play Store platform, they can push more game applications on their platform based what we learned from this project that most users installed game applications from Google Play Store. In addition, they could analyze the features of internal game application users, then based on the result, A/B testing would be a better choice to figure out how to establish their recommendation system on their platform. From A/B result to decide if this recommendation works for Google Play Store platform. If works, this recommendation system will bring more traffic and application developers.