Google Play Store Reviews Sentiment Analysis

Team 17 Zuoyin Li, Alice Jiang, Yi Ming, Zhiyuan Sun

Table of Contents

I. Business Problem

The problem that motivates us is the prospect of being able to analyze the sentiment of each app's reviews and ratings. The rationale behind this is that the mobile landscape is ever-changing in nature and has become a very challenging space to maneuver. With growing access to mobile phones and the ubiquitous presence of the internet, innumerable apps have been developed and many have perished in the process. Highly rated apps often bring lots of profits to the developer. However, the fact is that many apps have not been widely welcomed by the audience, so it is important to find out the key attributes of those negative feedback.

Mobile app analytics is a good way to understand how to implement a strategy to drive growth and retention of future users. With millions of apps around nowadays, the following data set has become very key to getting top trending apps in Google Play Store.

II. Dataset

The dataset was collected from Google Play Store which contains 12,495 reviews of different applications by real users. The dataset was downloaded from Kaggle (**https://www.kaggle.com/prakharrathi25/google-play-store-reviews**). There are 12 columns describing features of the reviews. Several sample features are listed in Table 2.1.

Table 2.1

Variable Name	**Data Type**	**Description**
content	factor	Review Text
score	int	Rating given to the application by the user (1 - 5) where 5 is the most positive and 1 is the most negative
thumbsUpCount	int	Number of users who upvoted the review
at	factor	Date and Time when the review was posted

We mainly focused on analysis on review content, score, and thumbUpCount. In the high score reviews, we investigated the top words from the reviews and found top functions favored by users. For the low score reviews, we explored the common problem and reasons. Moreover, we also provide actionable suggestions to Google Play Store.

III. EDA & Preliminary Analysis

Exhibit 3.1 shows the review frequency grouped by date of the post. The trend starts to increase dramatically from February 2020. We assume that the increased reviews are related to COVID-19. Due to the pandemic, people are more likely to use electronic devices. Hence, the feedback of the apps were rising.

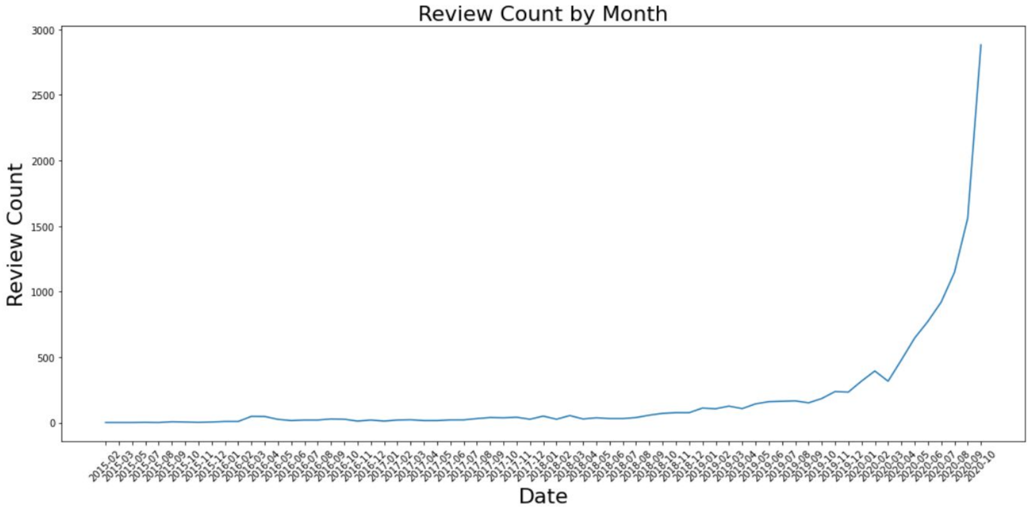


Exhibit 3.1

By looking at the distribution of review scores, we found that there are slight differences between frequencies of the scores (shown in Exhibit 3.2).

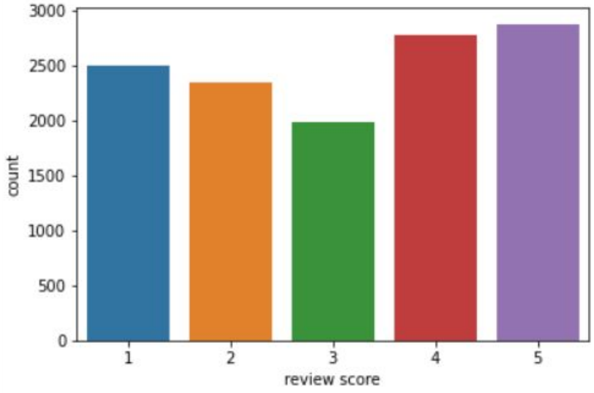
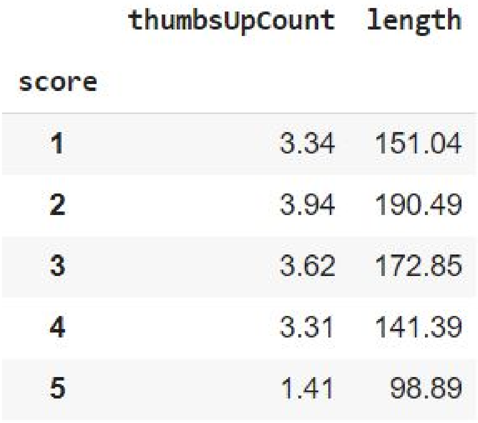


Exhibit 3.2

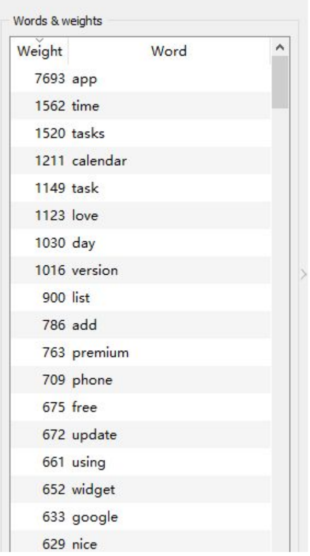


	thumbsUpCount	length
score		
1	3.34	151.04
2	3.94	190.49
3	3.62	172.85
4	3.31	141.39
5	1.41	98.89

Exhibit 3.3

We dived deeper into details about thumbsUp and length of views grouped by scores (Exhibit 3.3). It's interesting that customers are more likely to reach the census with long negative comments, because when the score is 2, thumbsUp is nearly 4. On the other

hand, the most positive sentiment (score = 5) has the lowest thumbsUp with the shortest reviews. It might indicate that there is a large room for apps to improve to reach the highest score and it’s necessary to dive even deeper to the negative reviews to see how to achieve that.
We also explored the relationship between sentiment and the number of thumbsUp. The calculated correlation between the number of thumbsUp and sentiment is -0.04167. In other words, there is no strong relationship shown between these features.



Weight	Word
7693	app
1562	time
1520	tasks
1211	calendar
1149	task
1123	love
1030	day
1016	version
900	list
786	add
763	premium
709	phone
675	free
672	update
661	using
652	widget
633	google
629	nice

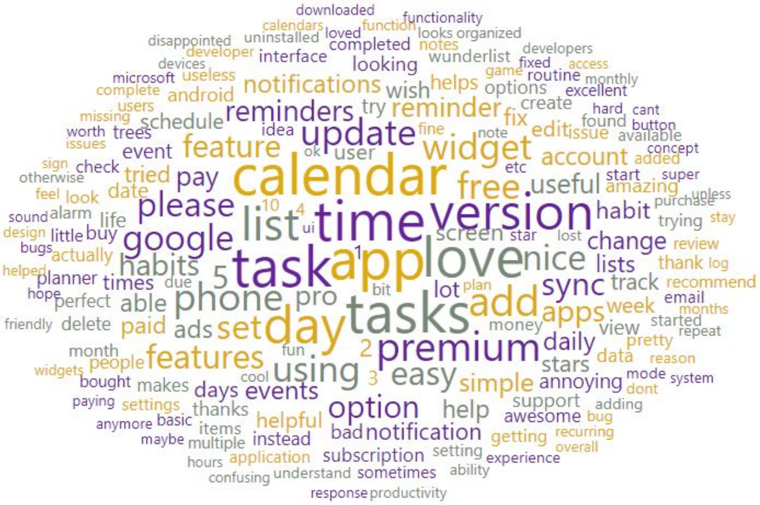


Exhibit 3.4

We generated Word Cloud based on the whole dataset (Exhibit 3.4), a collection of words that are depicted in different sizes. The bigger and bolder the word appears, the more often it is mentioned within our corpus and the more important it is. This word could show that the word “app”, “time”, “tasks”, “love”, “calendar”, “day”, and “version” are the words that have more than 1000 weight, which means it is the most significant word of the corpus.

IV. Sentiment Analysis

Sentiment analysis is an important part of our research, because there is a variable called score in the dataset. It represents part of sentiment's characteristics because it is the customer's evaluation of an app.

Firstly, we tried to use afinn to describe sentiment. Exhibit 4.1 drawn using the afinn package. In this figure, the positive values on the y-axis represents positive sentiment, and the negative values represents negative sentiment. The line at y=0 to separate the positive and negative sentiment. According to Exhibit 4.1, the positive sentiment is more than the negative sentiment.

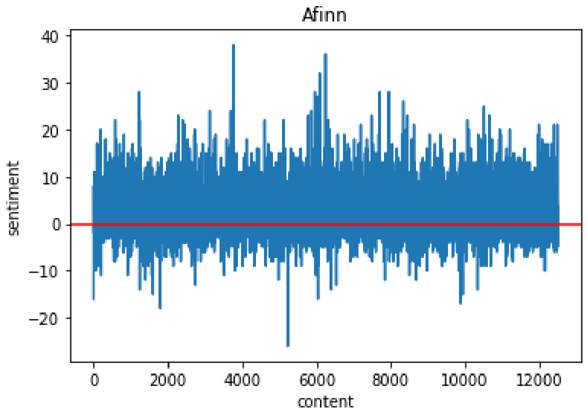


Exhibit 4.1

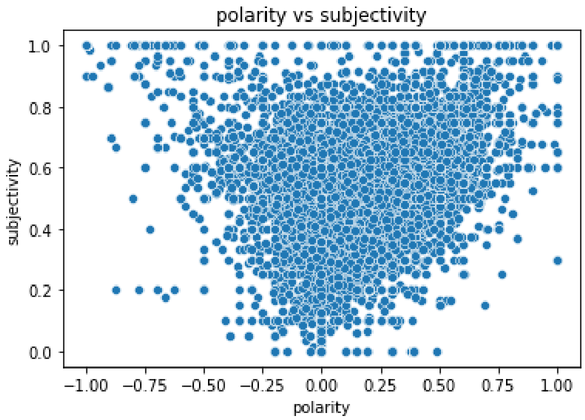


Exhibit 4.2

The sentiment attribute returns a named tuple in the form of sentiment including polarity and subjectivity. The polarity score floats in the range of -1 to 1. Positive polarity numbers indicate positive sentiment, while negative numbers indicate negative sentiment. Subjectivity is within the range of 0 to 1, where 0 is extremely objective and 1 is extremely subjective.

We created a scatterplot to measure the relationship between these two variables by setting subjectivity on the y-axis and polarity on the x-axis. From Exhibit 4.2, we can see that the points with lower subjectivity are all clustered near 0 in polarities. The points with higher subjectivity are distributed in all areas of polarity. The blanks in the figure are those areas where the polarity is close to 1 or -1, and the subjectivity is low. This is not hard to understand, because comments with strong positive or negative sentiment are often difficult to make more objective judgments.

Then we continue to visualize the relationship between sentiment and polarity. We set the x-axis to polarity and the y-axis to sentiment. Exhibit 4.3 has drawn a regression line for scatter points. This line describes the positive relationship between sentiment and polarity. Higher polarity often corresponds to higher sentiment. The line also fits our expected relationship between these two variables.

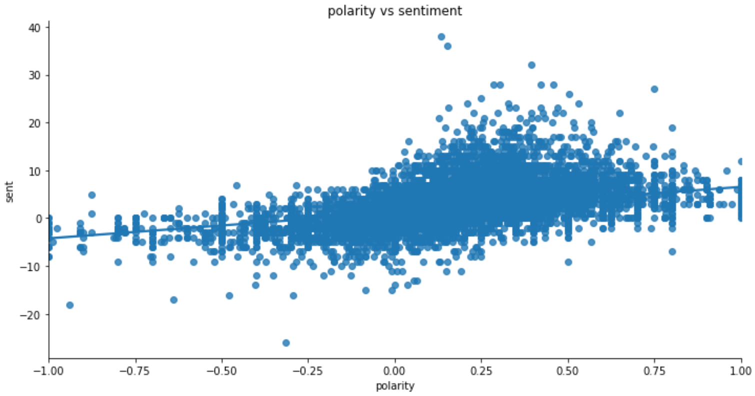


Exhibit 4.3

V. Topic Modeling LDA

Topic Modeling is a type of statistical modeling for discovering the abstract “topics” that occur in a collection of documents. Latent Dirichlet Allocation (LDA) is an example of topic models and used to classify text in a document to a particular topic. In this project, we applied LDA to reviews of applications in the Google Play Store reviews. Our business goal is to find the top reasons for the success and failure apps. So, it is important to find the topics that have a big impact on the popularity of these apps. Firstly, we generated two CSV files containing the text review content on Score 5 and Score 1, respectively. Score 5 represents the success of the software. On the contrary, Score 1 indicates that the app is not popular. The LDA model was applied to both Score 1 and Score 5 reviews and set the topic amounts to 6 interpretable topics. The output for Score1 is below:

LDA Topic Modeling for Score1

1: version, pro, list, paid, feature, settings, add, pay, bought, trees
2: task, set, screen, able, option, bug, hate, mode, options, star
3: subscription, cant, play, complicated, crashes, sounds, button, somehow, blocked, cause
4: la, wunderlist, Microsoft, downloaded, hard, de, que, en, miss, forced
5: update, data, notifications, bad, users, pop, event, 10, ad, android
6: app, time, premium, sync, tasks, calendar, ads, day, free, phone

This result clearly shows that certain features are disliked by customers, and some of the features have potential relationships. For example, in Topic 3, the words “subscription”, “complicated”, “somehow crashes”, “blocked” “cant play”, are all clearly expressing the dislike by users. By looking at Topic 3 and 1, we could know users do not like the app that has “complicated functions”, is always “somehow crashes” or “cant play”, and has to “pay” for “subscriptions” to “bought” more “pro features”. In Topic 5 and 6, we could know users do not like the apps that are always required to “update”, have many “pop notifications”, have too many “ads”, and number “10” might be the threshold of how many advertisements that the customers can tolerate.

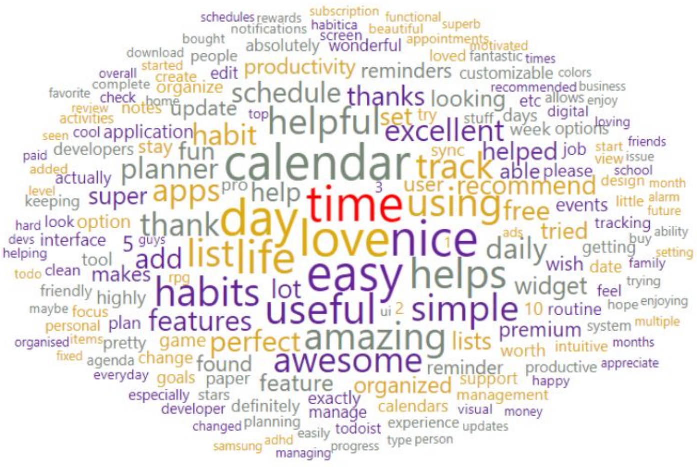


Exhibit 5.1 Word Cloud for Score 1

LDA Topic Modeling for Score 5

1: easy, simple, task, helpful, daily, features, calendars, etc, colors
2: app, love, calendar, time, nice, useful, amazing, thanks, widget, schedule
3: version, track, fun, free, user, friendly, paid, lots, understand, job
4: day, list, intuitive, cool, una, install, cute, looks, learning, finally
5: tasks, apps, awesome, help, reminders, makes, game, feel, download, style
6: helps, thank, life, recommend, lot, excellent, perfect, helped, create, stay

As for the good players, we could see many positive words such as “easy”, “simple, “helpful”, “nice”, “useful”, “free”, “fun”, “thank” in topics. There are some interesting findings when combining the words in the same model. For example, in Topic 6, the group of customers may want to express their appreciation of how the app helped their life and they are willing to recommend this app to others. Topic 1 and 2 might be a group of calendar apps that have some helpful features and help customers’ daily tasks effectively. In the EDA part, we found “Calendar” is one of the largest words, which means there are a large number of calendar apps reviews.

In order to better visualize the hot words of Score 1 and Score 5, the word clouds are shown in Exhibit 5.1 and Exhibit 5.2, respectively.

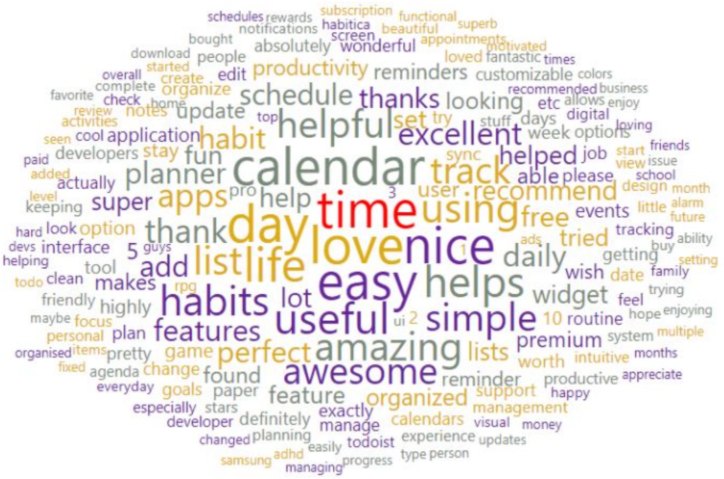


Exhibit 5.2 Word Cloud for Score 5

VI. Clustering Analysis

To better understand features and feedback from users regardless of the scores, we decide to apply clustering analysis in order to segment users into distinct groups and label these groups according to tokens reviews. We tokenize the lower capitalized reviews and select the top 50 unigrams after removing stopwords. Besides applying the default stopwords list, we also added some words specific for the dataset. For example, “app”, “phone”, “features”, etc.

1. Hierarchical Clustering

Hierarchical Clustering (Hcluster) is the one of the clustering approaches to explore the appropriate groups. We tried multiple combinations of distance metrics and linkage methods. As a result, we concluded that ward distance and cosine linkage are the best choices. In this case, the appropriate number of clusters is 7. The dendrogram reflects the hierarchical relationship (shown in Exhibit 6.1).

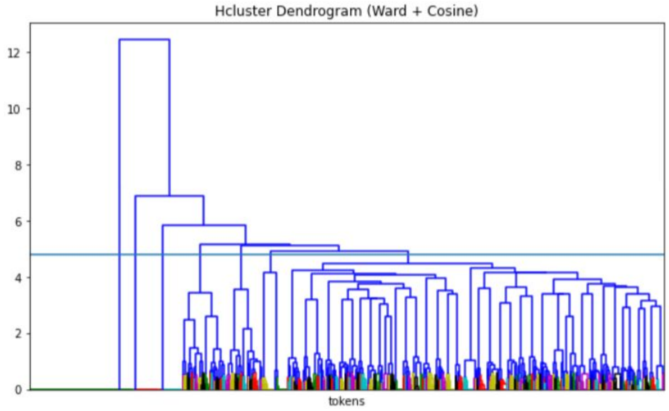


Exhibit 6.1

2. K-Means Clustering

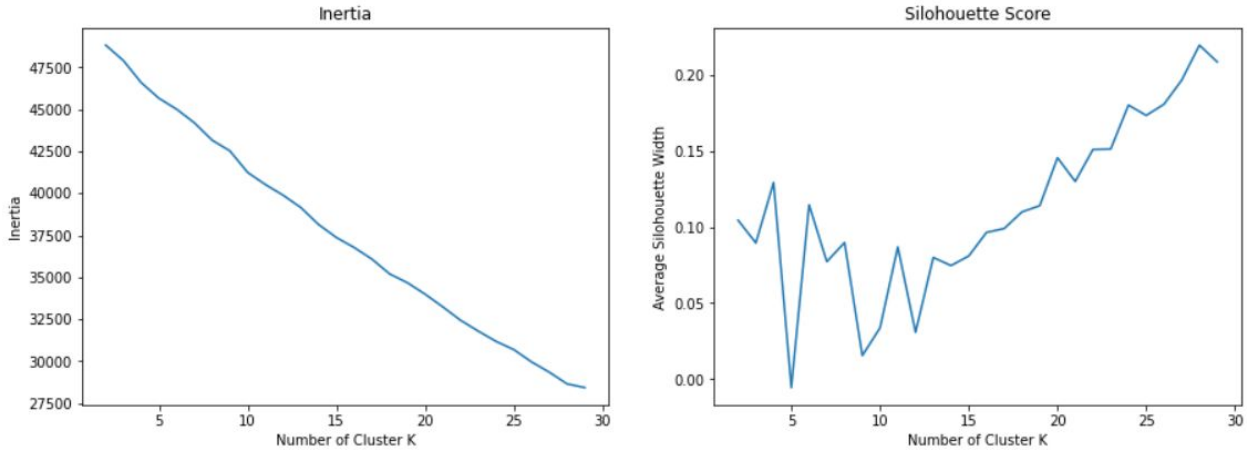


Exhibit 6.2

K-Means is the alternative method for clustering. In Exhibit 6.2 the left graph shows the inertia decreases by increasing the number of clusters. As for the right plot, it demonstrates how the average silhouette score varies as the number of clusters increases. The optimal K is chosen by keeping a high average silhouette score and minimizing the inertia value. Based on the results from Exhibit 6.2, the best K value is 6.

3. Clustering Methods Comparison

In order to select the best clustering model, we compared distribution and silhouette scores of the two clustering methods. Exhibit 6.3 visually represents how Hcluster and K-Means clustering methods fit the dataset. The K-Means generates a higher silhouette score and better distribution among the clusters. As for the plot of Hcluster, it contains a large number of negative values, which is not an ideal choice. Hence, we select K-Means as the clustering method to group the data into 6 segments.

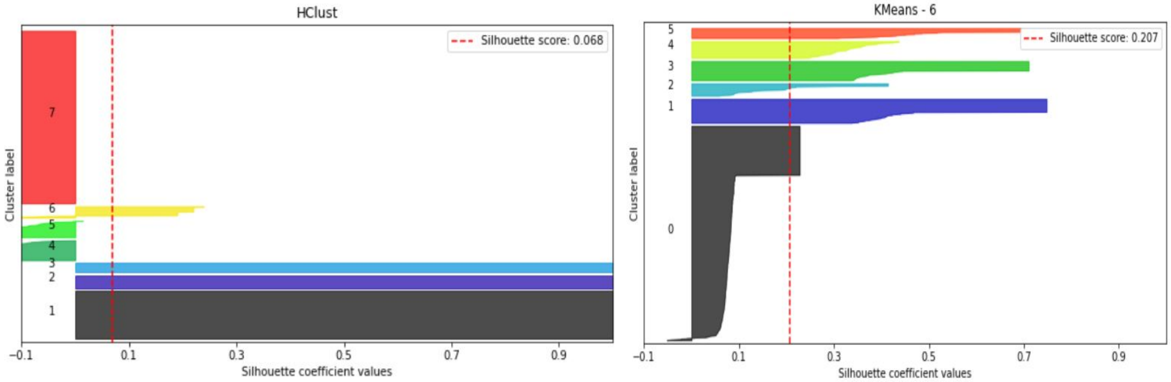


Exhibit 6.3

By adding cluster labels into the reviews, we got some observations from the clusters. The Exhibit 6.4 and Exhibit 6.5 indicate the average score and average thumb up in each cluster. The cluster 1 has the highest score but lowest thumbs up. As for cluster 0, it has a low average score but got the high thumbs up. The result emphasizes that there is no obvious positive correlation between the score and number of thumbs up in the clusters.

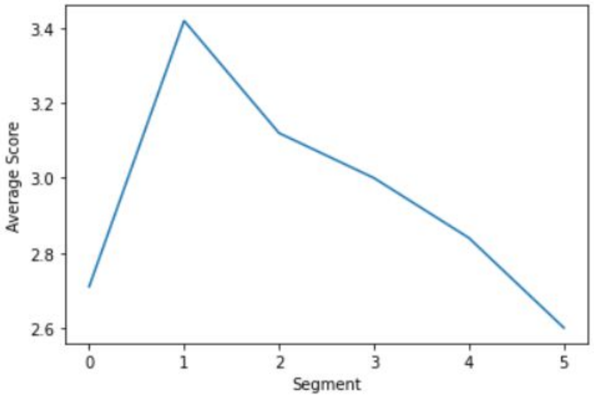


Exhibit 6.4

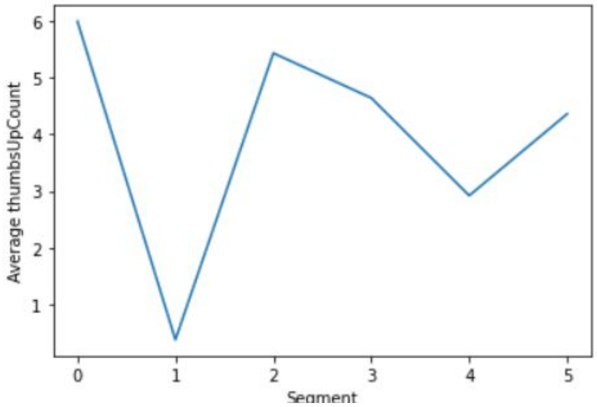


Exhibit 6.5

We dived into the content in the clusters, specifically for Cluster 1. Most of the reviews in Cluster 1 are related to versions of apps. We created a version word list containing “pro”, “premium”, “free”, “version”, “money”, and “price”. By counting words frequency from the version word list, we noticed that Cluster 1 has 159 counts of these tokens, which is extremely higher than other clusters. Therefore, we believe that there is a potential connection between review content within clusters. Recall that Cluster 1 has the high average score but low average thumbs up count. By combining the facts with the version related words, we conclude that most of the reviews in Cluster 1 are sharing fancy features in the premium/pro version. However, the low thumb up is possible due to the pricing.

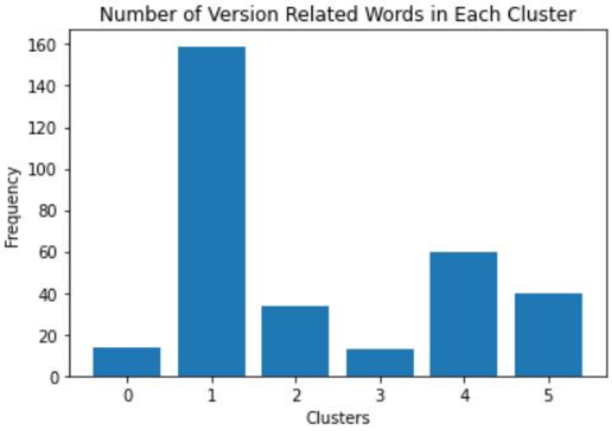


Exhibit 6.6

4. Dimensionality Reduction

After completion of finding out the proper number of clustering, dimensionality reduction is the next step we applied for further visualization and text analysis. The method being used is PCA and the number of components is 2, because we want to build a two dimensional map with one principal component value as x-axis and another one as y-axis. Drawing a scatter plot with principal component value as axises and grouped by segment. Exhibit 6.7 shows a good mapping of 6 clusters, because each cluster is significantly different from each other, which means that clustering analysis on tokenization can provide a nice grouping of reviews.

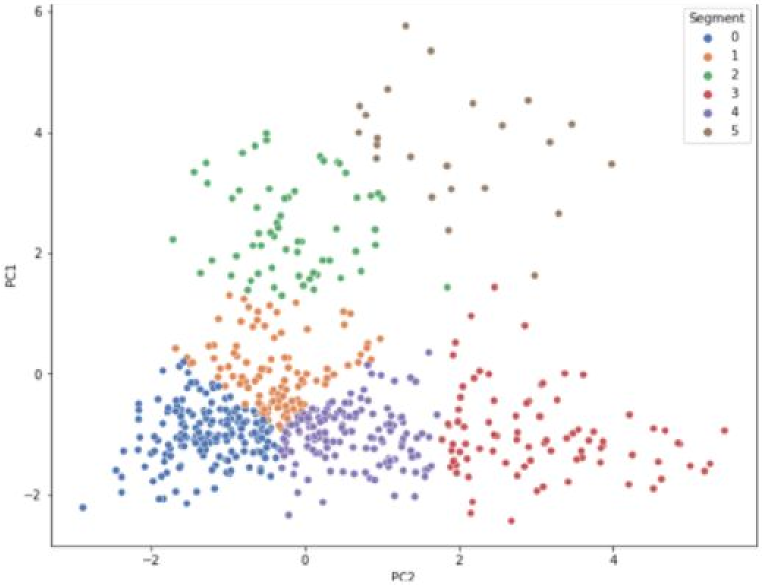


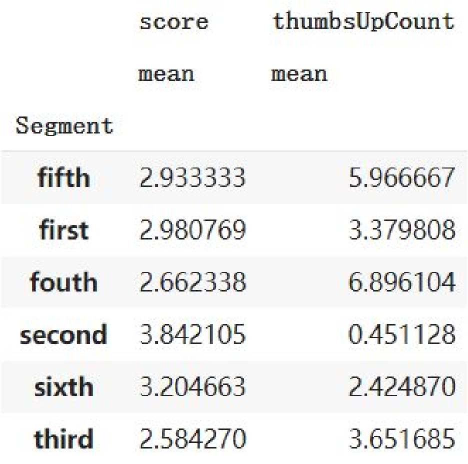
Exhibit 6.7

Exhibit 6.8 indicates that the thumbs up count of the fourth and fifth group is dramatically larger than that of other clusters. We assume that there might be some words that customers of those groups are likely to reach consensus. Then we tokenize content from those two groups and select meaningful words that come frequently. Here is the result:

Fourth group: work, version, tick, sync, support, starts, premium

Fifth group: work, widget, track, time, tasks, reminders, new

It's confident to say that customers are likely to give a thumbs up when they notice these words occur in the content.



	score	thumbsUpCount
	mean	mean
Segment		
fifth	2.933333	5.966667
first	2.980769	3.379808
fouth	2.662338	6.896104
second	3.842105	0.451128
sixth	3.204663	2.424870
third	2.584270	3.651685

Exhibit 6.8

VII. Conclusion & Recommendation

It is inaccurate to conduct customer sentiment analysis simply based only on the customer's rating of the product. We found that the plots of the afinn and textblob analysis methods tell us that the attitude of most customers is neutral (more than 70% of points are near 0), even if the scores given by customers are roughly evenly distributed. Therefore, when doing text mining, focus more on reviews with strong positive or negative attitudes, or reviews with scores of 1 or 5. The reviews will be more specific to the product. By mining the text data of score 1 and 5, we are able to transfer the complicated and wordy reviews to valuable business intelligence.

As recommended, easy-to-use, free, simple but functional apps are more easily to get popular in Google App Store. Although the premium that customers pay for gaining an enhanced version is one of the main ways to make profit for the freemium business(such as, apps development), app developers should still avoid placing too many ads in the free version, otherwise, the virtuous business circle will be broken.