Project Report – MSBA Internet Analytics

Vyshak Sugunan

Project Overview:

Perform sentiment analysis on user reviews for mobile games having 1000+ comments

Data Source:

Identified a mobile game with 1000+ reviews: “Among Us!” by InnerSloth

The reviews were fetched using the python package “app_store_scraper” for the reviews of the game from the Apple AppStore.

For the purpose of the project, we extracted around 2000 comments to get a good mix of good and bad comments. The game has a very high rating, so we expect the comments to be skewed positively.

The reviews we extracted would have the following information:

- isEdited: flag to indicate if the review has been edited
- title: Review title given by user
- rating: Rating given by user for the game in addition to the comment
- username: Gamer’s username in Appstore
- review: The entire unclean review text
- date: Date of review in datetime format

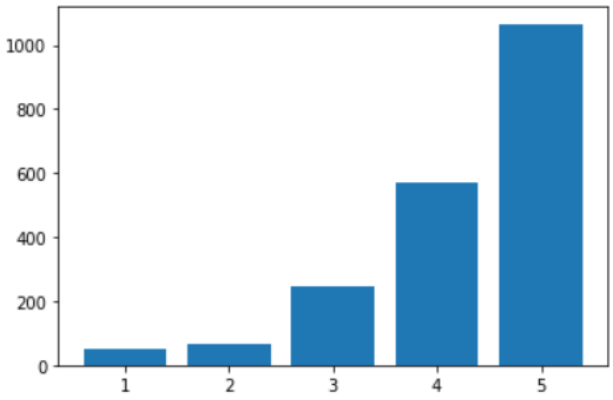
The majority of our analysis would be on the review text

Insights Objectives:

- Identify rating distribution
- Find keywords, bi-grams and trigrams most frequent in the reviews
- Identify the general sentiment of the users through topic modelling of the good and bad reviews separately

Rating distribution:

We analyzed the ratings of all the extracted reviews and came up with the below distribution:



From the distribution, we found that the ratings are always a whole number. Split the reviews as good and bad using 3 as the threshold => Good reviews are ones with rating >=3 and bad ratings < 3.

We create clean versions of the documents through preprocessing functions.

Frequently used keywords

To understand the frequently phrases and keywords used in the comments, we created a single function in which we could specify the number of top words/terms we need as well as if we want just single words, bi-grams or trigrams.

A much more elaborate information can be taken out from the trigrams, so including them in the report, but the top bi-grams and keywords can be seen in Jupyter notebook.

For good reviews

```
Top 20 trigrams and count
[('play among us', 91),
 ('love game much', 68),
 ('absolut love game', 59)
 ('game realli fun', 53),
 ('best game ever', 52),
 ('love among us', 51),
 ('keep good work', 40),
 ('realli love game', 40),
 ('among us great', 38),
 ('would make game', 35),
 ('call emerg meet', 35),
 ('game ever play', 35),
 ('game play friend', 34),
 ('fun play friend', 33),
 ('realli like game', 32),
 ('love game play', 31),
 ('game much fun', 30),
 ('realli good game', 29),
 ('love game fun', 28),
 ('would love see', 27)]
```

For bad reviews

```
Top 20 trigrams and count
[('game realli fun', 21),
 ('play among us', 18),
 ('tri play game', 12),
 ('version among us', 12),
 ('game play friend', 11),
 ('old version among', 10),
 ('realli fun game', 9),
 ('game super fun', 9),
 ('everi singl time', 9),
 ('absolut love game', 8),
 ('want play game', 8),
 ('love game play', 7),
 ('get kick game', 7),
 ('peopl leav game', 7),
 ('join anoth game', 7),
 ('realli love game', 7),
 ('get disconnect server', 7)
 ('kick peopl game', 6),
 ('realli fun play', 6),
 ('among us great', 6)]
```

Even in the relatively negative reviews, we can see people praising the game and how fun it is to play.

Some of the phrases that do stand out in the bad reviews are the version issues and server issues causing players to be kicked out. Some other issues seem to be people leaving and joining other rooms mid-game.

Topic Modelling

We make use of the genism packages LSI- and Coherence-model for topic modelling based on the good and bad reviews.

For good reviews

After testing out, the optimal number of topics was found to be 3. We modelled 3 topics of 10 words.

```
[(0,
  '-0.691*"game" + -0.266*"play" + -0.202*"like" + -0.198*"peopl" + '
  '-0.167*"would" + -0.152*"friend" + -0.150*"get" + -0.125*"love" + '
  '-0.123*"realli" + -0.122*"impost"'),
 (1,
  '-0.568*"game" + 0.391*"would" + 0.373*"like" + 0.189*"friend" + '
  '0.188*"could" + 0.175*"add" + 0.165*"peopl" + 0.147*"impost" + '
  '0.143*"think" + 0.135*"make"'),
 (2,
  '-0.630*"impost" + 0.442*"friend" + -0.253*"get" + 0.252*"play" + '
  '-0.222*"kill" + 0.162*"would" + -0.141*"task" + 0.122*"add" + -0.103*"like" '
  '+ -0.099*"crew"')]
```

The topics indicate that the people are really happy with the game but would also like some added functionalities – They would like to add friends through the game and would also like to have some of the functionalities of the “imposter” to be provided to the “crew”

For bad reviews

After testing out, highest coherence was for 4 topics:

```
[(0,
  '-0.726*"game" + -0.295*"play" + -0.200*"peopl" + -0.188*"get" + '
  '-0.137*"like" + -0.123*"time" + -0.113*"say" + -0.105*"impost" + '
  '-0.103*"would" + -0.100*"realli"'),
 (1,
  '-0.572*"peopl" + 0.423*"game" + -0.335*"like" + -0.317*"impost" + '
  '-0.154*"also" + -0.142*"make" + -0.115*"kill" + -0.110*"thing" + '
  '-0.088*"vote" + 0.084*"work"'),
 (2,
  '0.562*"play" + -0.376*"game" + 0.369*"get" + -0.295*"peopl" + 0.219*"time"
  '+ 0.123*"tri" + 0.122*"disconnect" + 0.109*"impost" + 0.107*"friend" + '
  '0.102*"fix"'),
 (3,
  '0.614*"impost" + -0.309*"peopl" + -0.302*"play" + 0.249*"get" + '
  '0.171*"game" + -0.146*"friend" + 0.122*"like" + -0.118*"would" + '
  '-0.116*"say" + 0.109*"time"')]
```

In the bad reviews, we can see that people complain about other players exposing themselves as the imposters, and the troubles with the voting system. They also ask for fixes on the disconnectivity issues and the wait time to get people into a room.

Conclusion

Overall, by performing analysis on the review comments, we can see the public sentiment towards Among-Us is really good. There is a general conception for added features and functionalities to make the game more social and engaging that is a helpful tool to let the developers know what the audience want.

We could implement the same process on other games as well to see how they fare and what issues plague them from a user perspective