Gaming Industry - STEAM and NLP Applications

Introduction

The gaming industry is worth approximately 140 billion in revenue across the globe. Gamers connect across the globe and transcend cultural, religious, location differences to play the same games, be it collaboratively in multi-player settings or individually, or in different genres such as role playing or action.

When a new game is released, game product managers want to know how players feel about the new game. While sales and ranking are the baseline KPIs, the gaming industry has moved to a point of sophistication where competition between games means the product managers have to dive deeper into players' feelings and feedback to differentiate their game development and offerings. Similarly, when new content is released to update a game, product managers want to know the contribution of the new content towards sales, vs other factors such as seasonality or different influences.

Before data science and NLP, all product managers could do was scan reviews with their eyes and survey a small number of players to gauge player sentiment. Obviously, this manual method is not representative of the whole player population. This manual method is time consuming and the time lag between start of player research to presentation of research insights prevents product managers from making timely changes.

Using NLP and data science techniques, we built a model that can provide updated dashboards and a visualization of players' sentiments whenever it is run. In this paper, we will demonstrate our methodology on a game released on May 23, 2019, so half a year into its launch - Total War: Three Kingdoms. It's a popular game right now and on the bestselling list at STEAM for its game categories: turn-based strategy, real time, single/multi player game. Analysis of this game at this stage can aid game developers in refining this game for updates and improved/future version releases as well as help gamers select and tryout similar games that are highly rated and fulfil their entertainment needs.

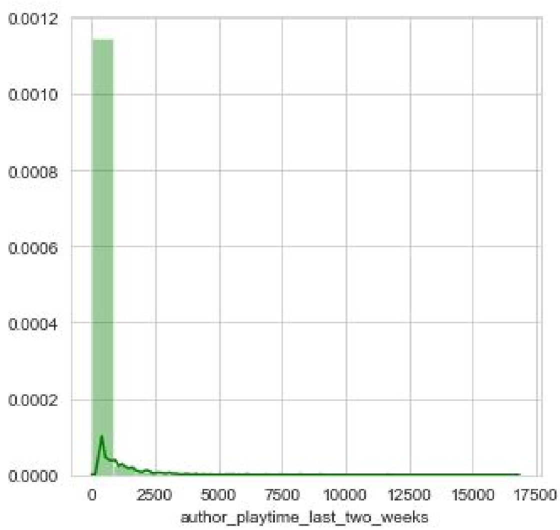
Preliminary Steps

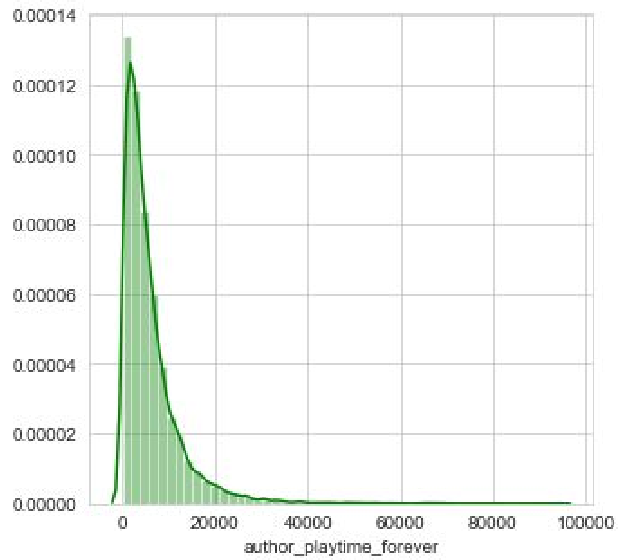
We obtained our data using an API and stored the data in a json file. Basic data exploration include reviewing the correlation between numerical variables and comparing graphically the statistical features.

	author num games owned	author num reviews	author playtime forever	author playtime last two weeks	author last played	timestamp created	timestamp updated
author num games owned	1.000000	0.287392	-0.074896	-0.031282	-0.085035	-0.033253	-0.030844
author num reviews	0.287392	1.000000	-0.080439	-0.013017	-0.067750	-0.055304	-0.021615
author playtime forever	-0.074896	-0.080439	1.000000	0.374205	0.384235	0.074799	0.089645
author playtime last two weeks	-0.031282	-0.013017	0.374205	1.000000	0.283315	0.207397	0.202133
author last played	-0.085035	-0.067750	0.384235	0.283315	1.000000	0.213381	0.225296
timestamp created	-0.033253	-0.055304	0.074799	0.207397	0.213381	1.000000	0.938900
timestamp updated	-0.030844	-0.021615	0.089645	0.202133	0.225296	0.938900	1.000000

From the correlation matrix shown above, we find that author_playtime_forever is positive relative to author_playtime_last_two_weeks and author_last_played. This is as expected. If a player has spent much time on the game, he or she is more likely to play it in the last two weeks.

Also, we see that timestamp_created and timestamp_updated are highly positively related. This indicates that when a person creates a review, he tends to update it as he continues playing and exploring other features. The time spans are not long apart. At the same time, when we make a regression model on these two features, we know to be attentive to the variances caused by the linear dependency.

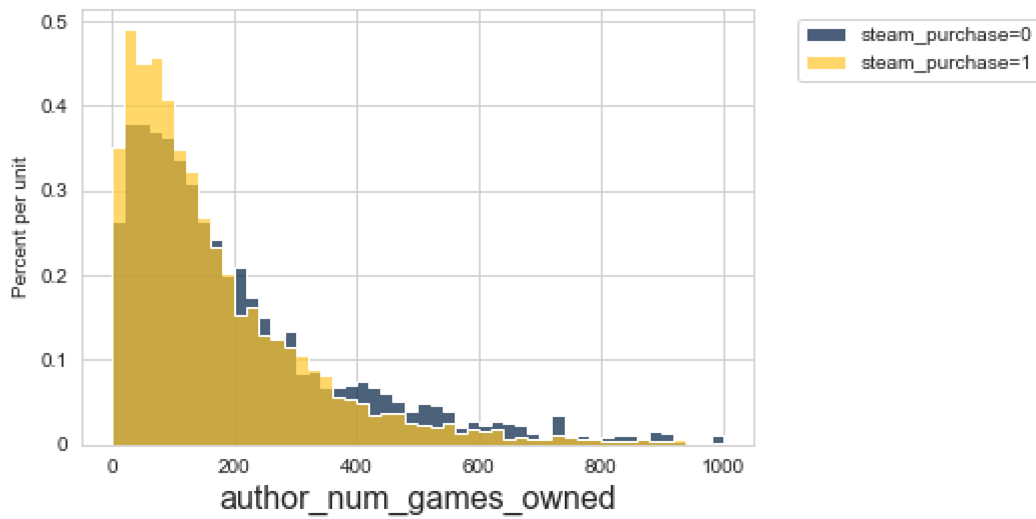




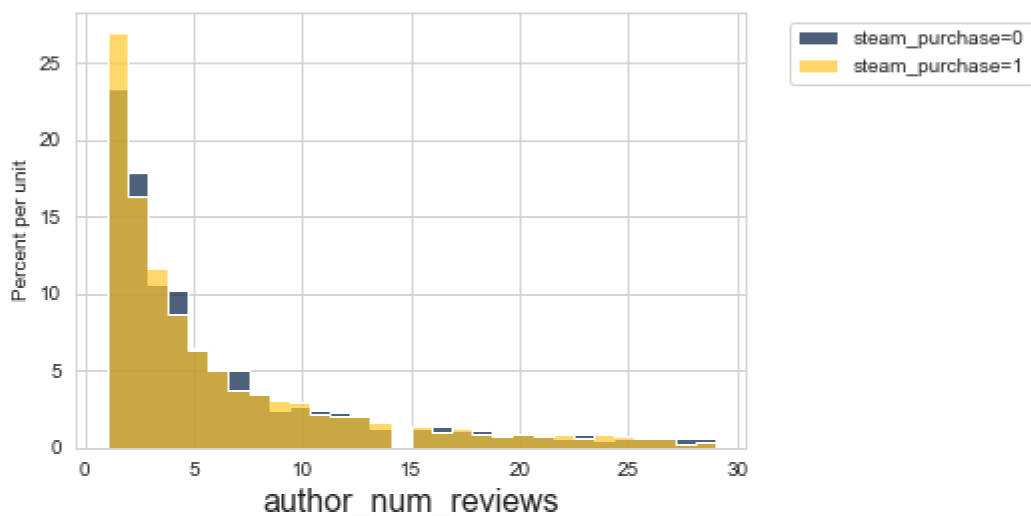
From the histogram above, we find that the distributions of author_playtime_forever and author_playtime_last_two_weeks are similar. We can learn the relationship between these variables later and make reasonable estimations basing off one variable for the other.

On Steam, we are also able to conduct some analyses on if the reviewer purchased the game. Obviously a reviewer has higher credibility if he purchased the game.

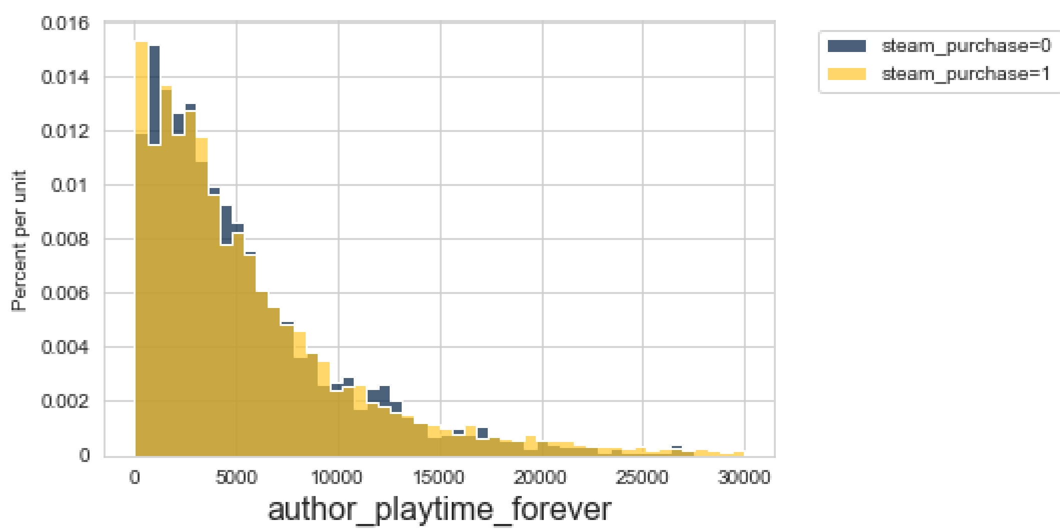
From the results of the A/B test, we find that people of different groups of steam_purchase aren't affected by games_owned_buy and author_num_reviews. So for STEAM/Three Kingdom developers, these two variables are not very valuable when they want to analyses whether the customers would buy the game. However,the author_playtime_forever is meaningful because the p-value is 0.968, showing that these two groups are really different. For STEAM, they should pay attention to the author_playtime_forever and target those customers who are more likely to buy the product. See graphical visualization below.



Percent per unit
0
0.1
0.2
0.3
0.4
0.5
0
200
400
600
800
1000
author_num_games_owned
steam_purchase=0
steam_purchase=1



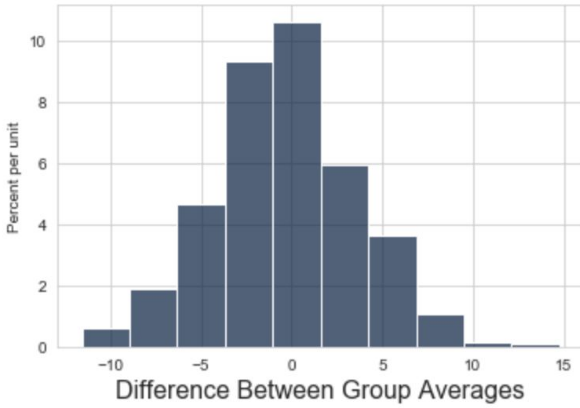
Percent per unit
0
5
10
15
20
25
0
5
10
15
20
25
30
author_num_reviews
steam_purchase=0
steam_purchase=1



Percent per unit
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0
5000
10000
15000
20000
25000
30000
author_playtime_forever
steam_purchase=0
steam_purchase=1

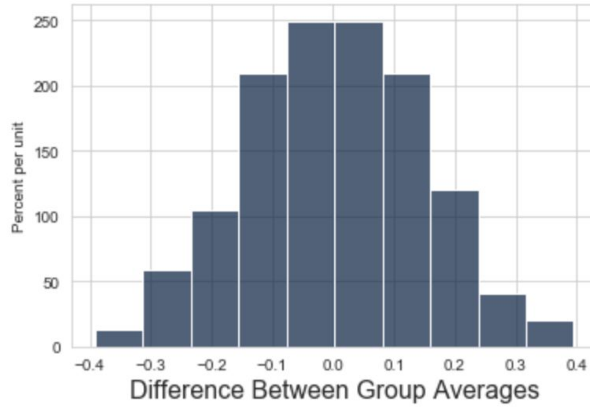
```
# A/B test on author_num_games_owned
AB_test(games_owned_buy_or_not_buy,500)
```

```
Observed value: -40.211385555467245
p value: 0.0
```



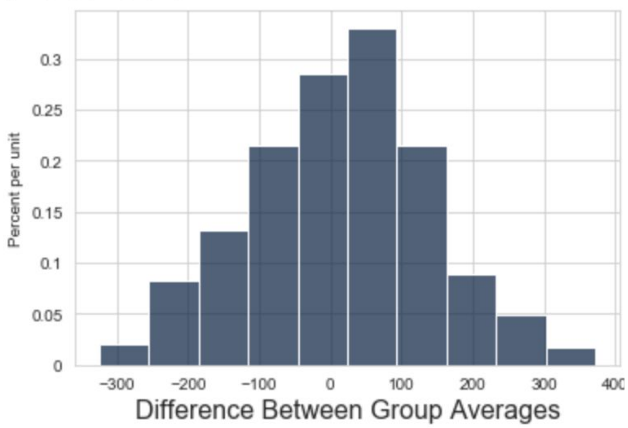

```
# A/B test on author_num_reviews
AB_test(author_num_reviews_buy_or_not_buy,500)
```

```
Observed value: -0.25645377730389907
p value: 0.034
```



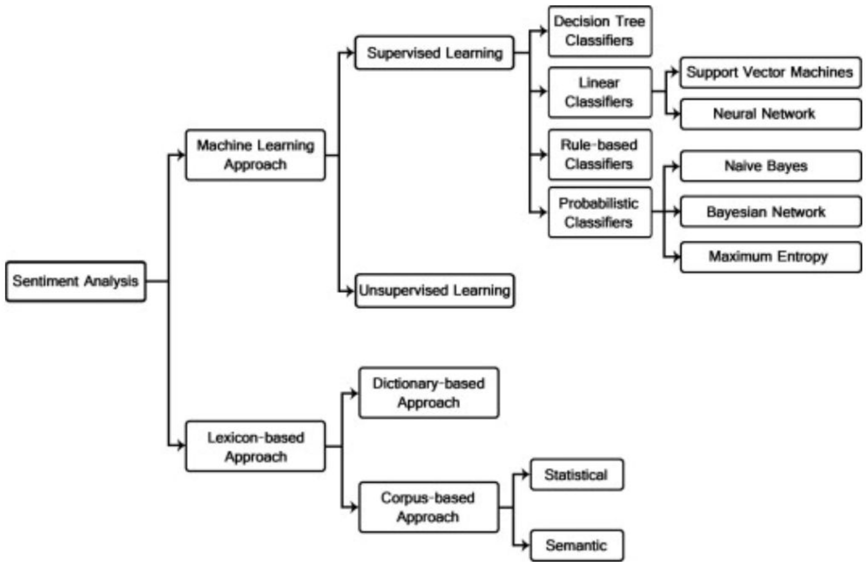

```
# A/B test author_playtime_forever
AB_test(author_playtime_forever_buy_or_not_buy,5
```

```
Observed value: 219.78640426897618
p value: 0.948
```



Sentiment Analysis

Sentiment Analysis can be considered a classification process. The target of sentiment analysis is to find opinions, identify the sentiments they express, and then classify their polarity. There are many applications and enhancements on sentiment analysis algorithms that were proposed in the last few years. These enhancements are categorized below to illustrate the main algorithms and techniques. We explored the naive and vader methods in the NLTK package for this paper.



Naive Approach

- Motivation: Try the naive approach based on positive/negative keywords to determine sentiment and analyze its performance
- For each document
    - We use our tokenize function on each document The function "tokenize" is defined as follows:
        - takes a string as an input
        - converts the string into lower case
        - tokenizes the lower-cased string into tokens. A token is defined as follows:
            - a token has at least 2 characters
            - a token must start with an alphabetic letter (i.e. a-z or A-Z),
            - a token can have alphabetic letters, "-" (hyphen), "." (dot), "'" (single quote), or "_" (underscore) in the middle
            - a token must end with an alphabetic letter (i.e. a-z or A-Z)
        - removes stop words from the tokens (use English stop words list from NLTK)
        - returns the resulting token list as the output
    - Counts positive words and negative words in the tokens using the positive/negative words lists. With a list of negation words (i.e. not, no, isn't, wasn't, aren't, weren't, don't didn't, cannot, couldn't, won't, neither, nor), the final positive/negative words are defined as follows:
        - Positive words:
            - a positive word not preceded by a negation word
            - a negative word preceded by a negation word
        - Negative words:
            - a negative word not preceded by a negation word
            - a positive word preceded by a negation word
        - determined the sentiment of the string as follows:

        - 2: number of positive words > number of negative words
        - 1: number of positive words <= number of negative words
- Please note:
    - voted_up - true means it was a positive recommendation

The accuracy of our sentiment analysis based on the naive approach is 0.58, which is not ideal. We then explored the vader approach below.

Vader Approach

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. It is fully open-sourced under the MIT license.

- Motivation: Improve the accuracy of sentiment analysis
- We use nltk.sentiment.vader(in package nltk) to classify the reviews in 3 sentiments: positive, neutral and negative
- Specific steps:
    1. Preprocess the data and import the packages:
       Link the vote_up feature, time feature and the review feature of every piece of reviews and transfer the Boolean value of vote_up to int value 1 or 0 (Picture 1.1 and 1.2).

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
from nltk.corpus import wordnet as wn
from sklearn.metrics import accuracy_score
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk import tokenize
```

Picture 1.1

```
text = list(text)
time = list(time)
vote = list(vote)
for i in range(len(vote)):
    if vote[i] == True:
        vote[i] = 1
    else:
        vote[i] = 0
```

Picture 1.2

    2. Use function SentimentIntensityAnalyzer to calculate the compound value for sentiment of all reviews. Here are some parts of the results in picture below (Picture 2.1).

```
Definitely worth a buy for TW fans.
compound: 0.5574,
neg: 0.0,
neu: 0.465,
pos: 0.535,
this game is ok, but compared to the content we get in the warhammer franchise, its not on par.
compound: 0.1531,
neg: 0.0,
neu: 0.918,
pos: 0.082,
```

Picture 2.1

3. Calculate the quantity of the reviews in three sentiments and the accuracy with the vote up value. And here are the results in picture below (Picture 3.1).

```python
print("The quantity of positive sentiment is",len(Pos_text))
print("The quantity of neutral sentiment is",len(Neu_text))
print("The quantity of negative sentiment is",len(Neg_text))
print("The accuray of the our sentiment analysis is %.2f" % acc_Pos)
```

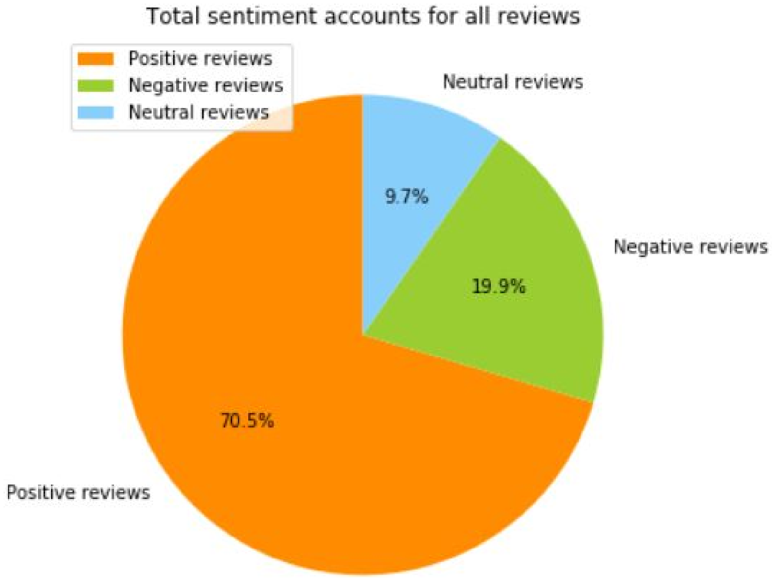
```
The quantity of positive sentiment is 7606
The quantity of neutral sentiment is 1044
The quantity of negative sentiment is 2146
The accuray of the our sentiment analysis is 0.93
```

Picture 3.1

4. Paint a pie chart to show the account of three sentiments in total reviews (Picture 4.1). And here is the result of the total sentiment accounts for all reviews (Picture 4.2).

```python
plt.figure(figsize=(6,9))
labels = [u'Positive reviews',u'Negative reviews',u'Neutral reviews']
sizes = [len(Pos_text),len(Neg_text),len(Neu_text)]
colors = ['darkorange','yellowgreen','lightskyblue']
patches,l_text,p_text = plt.pie(sizes,labels=labels,colors=colors,
                                labeldistance = 1.1,autopct = '%3.1f%%',shadow = False,
                                startangle = 90,pctdistance = 0.6)
plt.title('Total sentiment accounts for all reviews')
plt.legend()
plt.show()
```

Picture 4.1



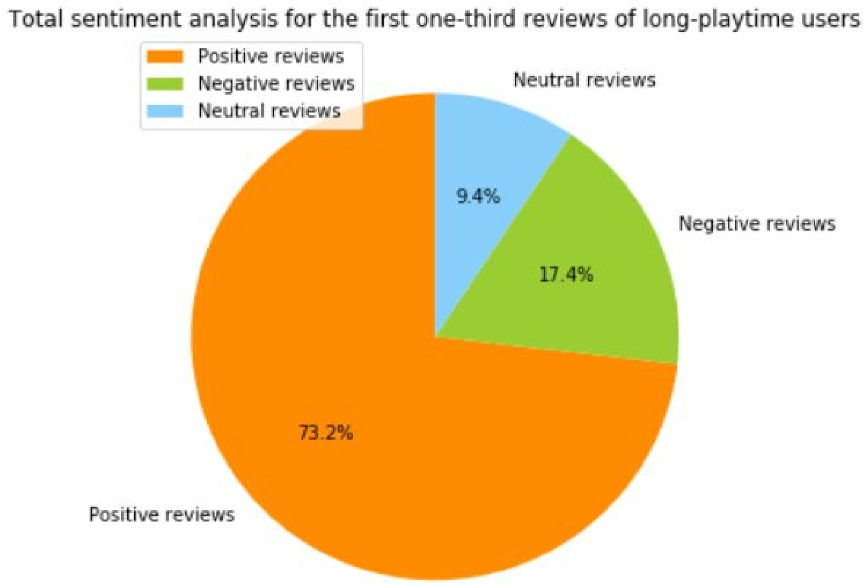
Picture 4.2

5. Sort the data according by the play-time and split them into three pieces (Picture 5.1).

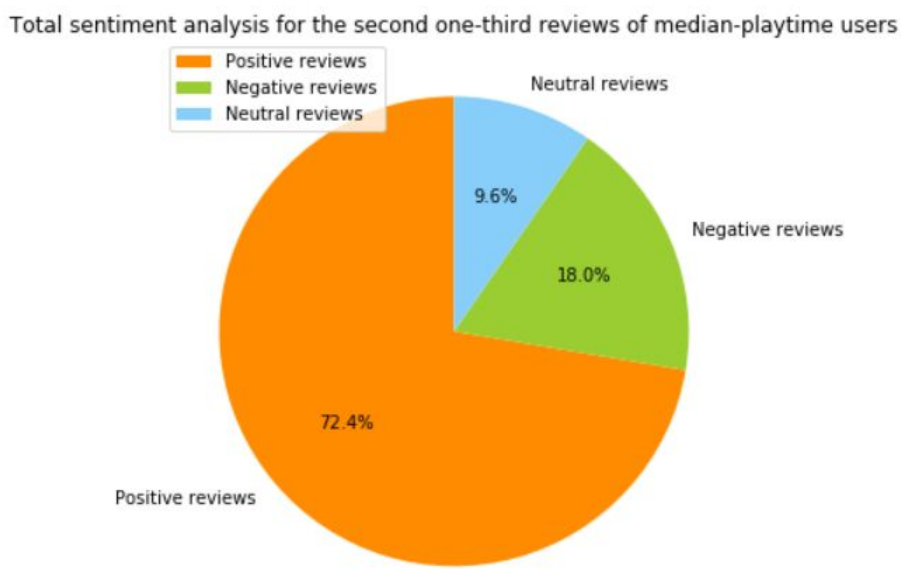
```
dic2 = sorted(dic.items(), key=lambda e:e[1], reverse=True)
a = int((len(dic2))/3)
b = int(a*2+1)
d1 = dic2[0:a+1]
d2 = dic2[a+1:b+1]
d3 = dic2[b+1:]
```

Picture 5.1

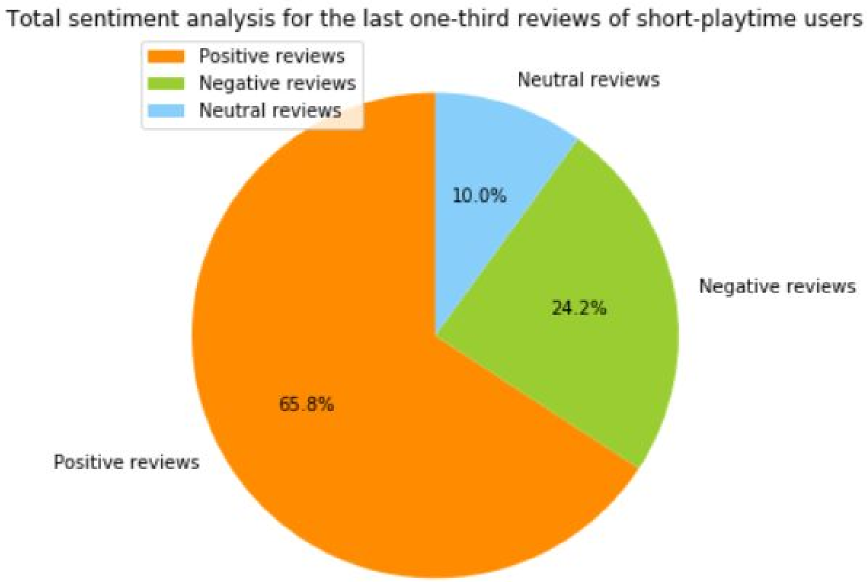
6. Repeat step 2 and 4 for three pieces. Here are the results for the three pie charts. The picture 6.1 shows the account in the first part (more play-time part). The picture 6.2 shows the account in the second part (median play-time part). The picture 6.3 shows the account in the last part(less play-time part)



Picture 6.1



Picture 6.2



Picture 6.3

Analysis

1. From the 0.93 accuracy, we could find that the classification is more accuracy than before.
2. In order to know more specifically about the effectiveness, we need to know the account of three types of people according to the total play-time, because some people who play more time just due to the interest of the game, in that case, their review could more trusty.
3. From the chart (Picture 4.2), we can find that the positive reviews take most account of total reviews(70.5%), negative reviews take about 20% percentage(19.9%) of the total reviews, and the neutral reviews take least account of total reviews(9.7%). The situation could besically match the reality, because most players would write a review after they won or lost. At that time, most people will write a review in positive emotions (most are winners) and negative emotions (most are losers).
4. From the chart (Picture 6.1), we can find that the account is similar to the total situation. Positive reviews take more account and negative reviews take less account. Neutral reviews account is nearly that in total reviews. Here are some slight differences. That verified the opinion just now that who play more time for the games would write better reviews.
5. From the chart (Picture 6.2), we can find that comparing with the people who play the game more time, the account of the positive reviews has decreased a little and the account of the negative reviews has increased a little. However, comparing with the total reviews, the account of the positive reviews is still more. Therefore, the most of players who play the game in the median part are also basically satisfied for the game.
6. From the chart (Picture 6.3), we can find that the account of the positive reviews decreased more clearly comparing with that of the more play-time players and the median play-time players. And similarly, the negative reviews account increased more clearly than in those charts. Therefore, it could show that the players who play less time are not satisfied with the game so that they will give worse reviews.

In conclusion, firstly, about the result, we can get an opinion that people will pay more attention on what you like, so the play-time could partly reflect the sentiment of the game. Secondly,
The vader could more accurately classify the sentiments.

About some insufficient points, maybe because the lack of the corpus for other languages, the package couldn't judge reviews in other languages except English, and both think those are neutral, that would influence the accuracy. And also, if the reviews are too long, the judgement by every single sentence would also account for the wrong judgement such as a "However" in last sentence. If in the future, we have more time, we will try to improve those insufficient points.

Topic Modeling

Topic modeling is a frequently used text-mining tool for discovery of hidden semantic structures in a text body. Intuitively, given that a document is about a particular topic, one would expect particular words to appear in the document more or less frequently. Topic models can help to organize and offer insights for us to understand large collections of unstructured text bodies. In this paper, we used Latent Dirichlet Allocation (LDA) and Biterm Topic Model (BTM).

LDA is an example of topic model and is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions.

BTMis a word co-occurrence based topic model that learns topics by modeling word-word co-occurrences patterns (e.g., biterms)

- A biterm consists of two words co-occurring in the same context, for example, in the same short text window.
- BTM models the biterm occurrences in a corpus (unlike LDA models which model the word occurrences in a document).
- It's a generative model.

LDA Analysis
Methodology: split the data basing on vote_up
The aim in this process is to extract information from different group of people. As we have mentioned before, voted_up with true means it was a positive recommendation. Basing on this variable, we Split the whole people into two group:
--- Group1 - 'voted_up' == True
--- Group2 - 'voted_up' == False

```
df_raw['voted_up'].value_counts()
```

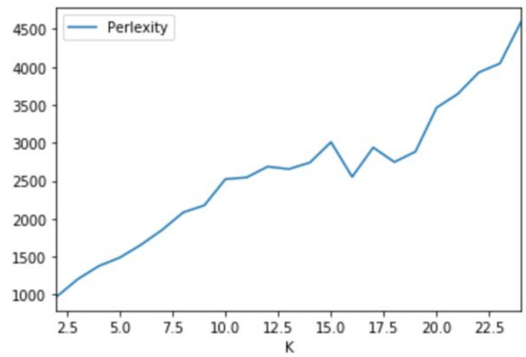
```
True     8309
False    1464
Name: voted_up, dtype: int64
```

From the above we see that among all 9773 reviews. 8309 hold positive recommendation, 1464 hold negative. This result also shows us that most reviewers think this game is worth being recommended. So, it indicates that we should focus more on group1 instead of group2 when we are making business plan.

Use LDA to find topic

For Group1 with 'voted_up' == True

Firstly, we need to find the number of topics. Perplexity may be one way for you to find the number of topics. Typically, the best number of topics should be around the lowest perplexity. However, in practice, a few factors need to be considered: It is usually difficult for human to understand a big number of topics. Usually, after LDA, we need manually inspect each discovered topic, merge or trim topics to get semantically coherent but distinguishable topics.



From the perplexity-K plot we choose the number of topics as 16. Because 16 is a local minimal. Moreover, we know for the real word data, it can't be perfect. For instance, we use K to divide the documents into K topics, it doesn't mean that the origin documents have K topics. This in not happening in the real world. So, what we need to do is to find some significant topics among these K topics.

The result is bellowing:

Topic 0:
[('strategy', 41.963425489039714), ('fantastic', 38.11606291106211), ('long', 34.88380574020764), ('graphics', 22.656805679215093), ('release', 17.488618784445862)]

Topic 1:
[('epic', 26.52328259280747), ('simply', 20.186462703958014), ('story', 17.84728686153461), ('easily', 14.998722208508017), ('era', 13.950685915085664)]

Topic 2:
[('finally', 24.84276111228602), ('shit', 22.178559218665686), ('job', 21.934862287396903), ('koei', 18.7195067181397), ('medieval', 17.54397483826273)]

Topic 3:
[('probably', 35.78160336515408), ('gud', 34.106549140364045), ('yuan', 27.464087889755433), ('shao', 26.78332186723511), ('sun', 21.295455720733724)]

Topic 4:
[('dlc', 57.59542401532326), ('blood', 48.30491206944922), ('worth', 45.24598529020593), ('need', 28.501579429046757), ('turn', 27.18623047027297)]

Topic 5:
[('nice', 125.22442140155626), ('buy', 53.88382210648463), ('excellent', 33.01697828731698), ('enjoying', 21.137107351575754), ('rice', 18.8404635764391)]

Topic 6:
[('dynasty', 56.341635951573316), ('warriors', 52.15313053263923), ('years', 26.731306424176864), ('cao', 25.669395559020156), ('creative', 25.58756161601824)]

Topic 7:
[('favorite', 38.08237458638031), ('period', 26.98848117484265), ('absolutely', 24.926267128808895), ('beautiful', 24.19799691928118), ('setting', 22.38804544482986)]

Topic 8:
[('runs', 30.436903978276842), ('needs', 21.551881937519404), ('optimization', 15.928834054612866), ('dream', 15.7635412486581), ('smooth', 15.346940139936757)]

Topic 9:
[('date', 96.18132912154302), ('cool', 23.541717564972803), ('ii', 20.38121772566235), ('title', 19.412756045263666), ('titles', 18.46500624156434)]

Topic 10:
[('awesome', 98.91175081515468), ('kingdom', 21.040886019576764), ('ok', 19.833680352048784), ('wanted', 17.431455470661422), ('brilliant', 15.21632950978395)]

Topic 11:
[('highly', 36.660137809831355), ('perfect', 34.04549439785453), ('recommended', 27.035712110554055), ('dong', 26.28266623694069), ('2019', 16.05365436936465)]

Topic 12:
[('yes', 60.92264094593182), ('damn', 18.331943832871406), ('entry', 17.727289075090525), ('makes', 13.826664479976165), ('tho', 13.271939385109546)]

Topic 13:
[('franchise', 46.670192021709994), ('enjoy', 32.69045477543598), ('wars', 29.008929253133605), ('addition', 26.22671055588695), ('masterpiece', 17.912096766319923)]

Topic 14:
[('chinese', 54.20635475129121), ('history', 47.01889721147863), ('solid', 27.144770779168045), ('grand', 15.33506570146484), ('strategy', 13.605907123094568)]

Topic 15:
[('bu', 78.41575917268662), ('lu', 71.87472562222608), ('faction', 53.98265414801434), ('army', 51.74624879144724), ('factions', 47.2983556957114)]

After fitting the data, we generate the document-topic matrix. The value in this matrix represent the percentage of each topic in each document.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0.018177	0.018177	0.392601	0.018177	0.352924	0.018177	0.018177	0.018177	0.018177	0.018177	0.018177	0.018177	0.018177	0.018177	0.018177
1	0.013388	0.013388	0.013388	0.013388	0.013388	0.013388	0.013388	0.013388	0.191546	0.013388	0.013388	0.013388	0.013388	0.013388	0.013388
2	0.031250	0.031250	0.031250	0.031250	0.031250	0.531250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250
3	0.727737	0.018151	0.018151	0.018151	0.018151	0.018151	0.018151	0.018151	0.018151	0.018151	0.018151	0.018151	0.018151	0.018151	0.018151
4	0.031250	0.531250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8304	0.020888	0.192327	0.020888	0.020888	0.020888	0.020888	0.020888	0.020889	0.020888	0.515235	0.020888	0.020888	0.020888	0.020888	0.020889
8305	0.031250	0.031250	0.531250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250
8306	0.062500	0.062500	0.062500	0.062500	0.062500	0.062500	0.062500	0.062500	0.062500	0.062500	0.062500	0.062500	0.062500	0.062500	0.062500
8307	0.031250	0.031250	0.031250	0.531250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250	0.031250
8308	0.016458	0.016458	0.127037	0.016458	0.016458	0.016458	0.016458	0.016458	0.016458	0.341919	0.016458	0.016458	0.016458	0.317095	0.016458

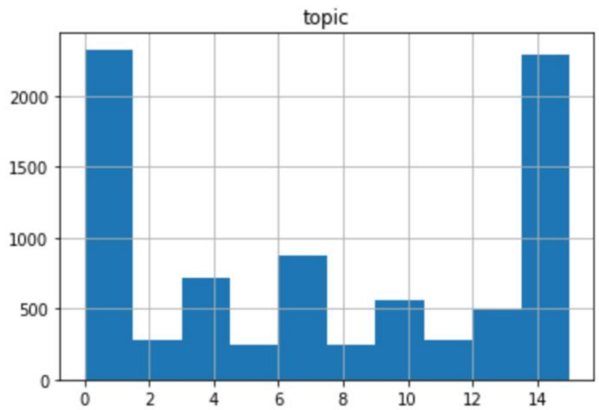
8309 rows × 16 columns

Then we can assign topic to each document

	text	topic
0	A good back to form for total war, so far a ga...	2
1	Awesome game , but my only problem is the camp...	15
2	nice	5
3	better than every Total war game i have played...	0
4	I Love Three Kingdoms Story	1
...	...	...
8304	The most engaging Total War game yet. I hope t...	9
8305	incredible	2
8306	A very elaborate game	0
8307	gud	3
8308	Fantastic latest entry in this great franchise...	9

8309 rows × 2 columns

We also draw the histogram of the topic:



We find that topic0 and topic15 have the most document. The others have a similar size.

Among these 16 topics, we manually check the top 10 documents for each topic. That is, we sort each column for the above document-topic matrix, and find the top 10 high percentage documents for each topic. We pick up some reasonable topic with representative documents to show our LDA model works:

Topic 14:
[('Chinese', 54.20635475129121), ('history', 47.01889721147863), ('solid', 27.144770779168045), ('grand', 15.33506570146484), ('strategy', 13.605907123094568)]

```
The top 10 review from topic 14:
I do like the game but not the graphics plus so many Chinese names and groups with
unfamiliar names to us Westoners perhaps the ability to who is who on the map as we
are asked to make decisions might help? your next one must be about the WW1!!

The top 10 review from topic 14:
SO happy this Total War title has Chinese audio. Good work guys!

The top 10 review from topic 14:
For those who like the Ancient Chinese History especially Three Kingdoms, please grab
this game and experience the feel of becoming an emperor of Ancient China.

The top 10 review from topic 14:
Like Total War?
Like Chinese History?
Want to be an Emperor?
Then you will love Total War: Three Kingdoms.
DO NOT HESITATE! I REPEAT. . . DO NOT HESITATE TO BUY THIS GAME!!
```

We assign this topic 14 to ‘Chinese history prefer’. As we see from the example, this topic talks about chinses, Chinese history. These customers show their passion on the Chinese culture. The reason they choose this game is because they are interested in Chinese or Chinese history. Also, from the histogram we know topic 14 has the most reviews. This means that many customers who recommend for this game is because of their preference on Chinese. As a result, what we can suggest to the game company is to optimize the modeling of Chinese elements in the game. Let foreign players better understand and integrate these Chinese elements.

Topic 13:
[('franchise', 46.670192021709994), ('enjoy', 32.69045477543598), ('wars', 29.008929253133605), ('addition', 26.22671055588695), ('masterpiece', 17.912096766319923)]

The top 10 review from topic 13:
This is a fantastic entry to the Total War series. The diplomacy system is amazing. The combat is as great as you can always expect from this franchise and by God is it a beautiful game.

The top 10 review from topic 13:
i enjoy this game very much. the diplomacy mechanic is very akin to that seen in GoT. THIS GAME IS SO AMAZING CA SHOULD ACQUIRE THE RIGHTS OF GAME OF THRONES FROM GRR TO MAKE A TOTAL WAR: GAME OF THRONES! DOUBLE HACK N SLASH POWAAAA!!!! and politics lol =3

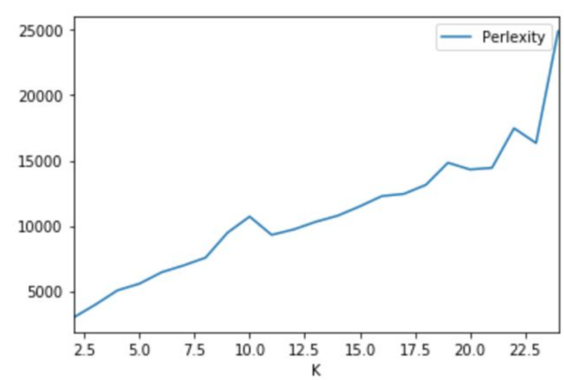
The top 10 review from topic 13:
I prayed for this game to be good, because too many recent Total War releases were pretty bad and I'm not the biggest fan of the Warhammer franchise and very surprising this game didn't disappoint! I wholly recommend to anyone that's a fan of the Total War franchise.

The top 10 review from topic 13:
Has the best end game mechanic out any of the total wars. Had an absolute blast when the I declared myself as Emperor.

We assign topic 13 to 'politic diplomacy'. As we see from the example, this topic talks about diplomacy, franchise. These customers show their interested in the political element of the game. For the game designer, especial for the plot design, they should get the feedback from this topic cluster and make the improvement on the game.

For Group1 with 'voted_up' == False

Remind that this group of people hold a negative recommend on the game. We follow the same process as above.



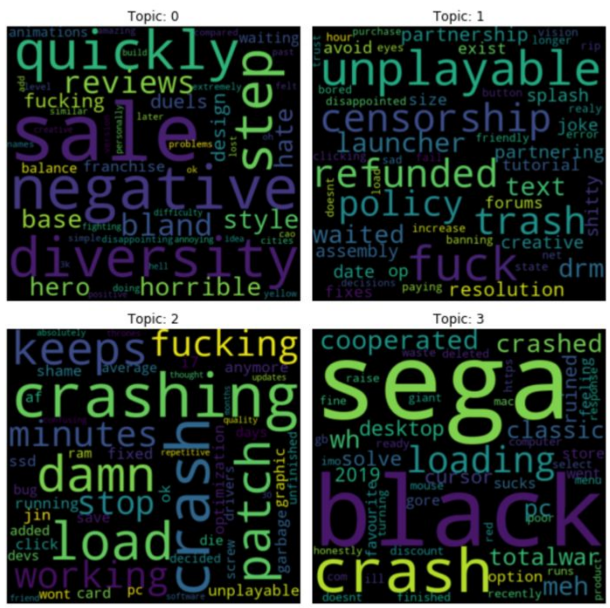
From this plot we can't directly get the K. So, we will take many try to determine the K. At last, we choose K as 4 and we get these four topics.

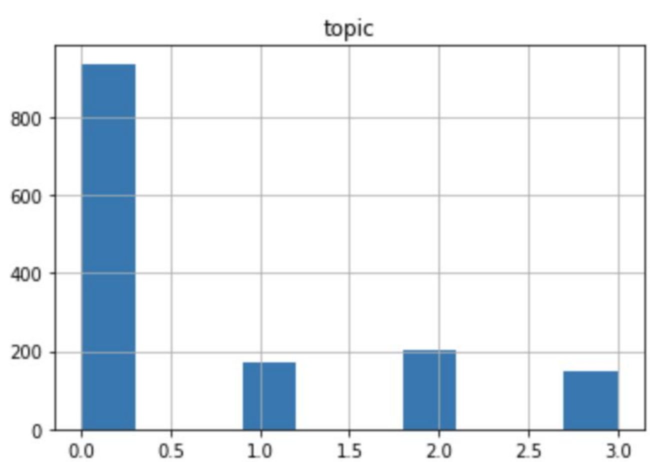
Topic 0:
[('sale', 8.60916113180274), ('negative', 8.229582948563916), ('diversity', 8.189723048132425), ('quickly', 8.037046758750055), ('step', 7.8323067818078504), ('reviews', 7.799226115860938), ('bland', 7.733778460545213), ('style', 7.573187212319218), ('base', 7.357653057070072), ('horrible', 7.23431616665337)]

Topic 1:
[('unplayable', 9.609854961765036), ('fuck', 9.45743985388559), ('refunded', 6.253663893523959), ('censorship', 6.187523616078877), ('trash', 5.9886105330778125), ('policy', 4.961448612418783), ('launcher', 4.644232380864038), ('drm', 4.042872921289739), ('text', 3.9572492404632325), ('waited', 3.695524332186118)]

Topic 2:
[('crashing', 8.709674489457079), ('crash', 8.64485528726209), ('load', 6.242042399676977), ('damn', 5.685498832679443), ('patch', 5.20686063127545), ('keeps', 4.764745016754615), ('working', 4.423377639013047), ('fucking', 4.121079708218993), ('minutes', 4.10632915457768), ('stop', 4.096002041221054)]

Topic 3:
[('sega', 6.430503769671124), ('black', 5.915012491617888), ('crash', 5.690826098962906), ('loading', 5.664613107922646), ('cooperated', 4.0176065535657255), ('meh', 3.9048020343864653), ('wh', 3.797188339174103), ('crashed', 3.7086226837905087), ('classic', 3.325155853279093), ('totalwar', 3.295381324948687)]





We follow the same rule to draw the example document from group 2:

Topic 1:
[('unplayable', 9.609854961765036), ('fuck', 9.45743985388559), ('refunded', 6.253663893523959), ('censorship', 6.187523616078877), ('trash', 5.9886105330778125), ('policy', 4.961448612418783), ('launcher', 4.644232380864038), ('drm', 4.042872921289739), ('text', 3.9572492404632325), ('waited', 3.695524332186118)]

```
From topic 1:
First there was the censoring of mods and mod creators.
Then there was the NetEase partnership.
And now it's the price hike of all the products in Creative Assembly's historical
catalog in various regions, some as much as 200%
Take your business elsewhere.

From topic 1:
The game was good, but the censorship and having to pay for gore when you could make a
mod that adds it is dumb.

From topic 1:
Fuck Netease, Fuck CCP censorship, fuck the CCP and fuck CA for bowing to fascists.

From topic 1:
Censorship of mods being main feature

From topic 1:
From one of my favourite games, to totally unplayable overnight.

From topic 1:
Terrible UI, refunded after an hour or so
```

We assign this topic1 to 'censorship problem, unplayable, refund'. Many reviews in this topic said they refund because they are not satisfied with the censorship or the game is unplayable. This topic should be attached a great importance with. Because the censorship problem may be a common problem in some cases.

Topic 2:
[('crashing', 8.709674489457079), ('crash', 8.64485528726209), ('load', 6.242042399676977), ('damn', 5.685498832679443), ('patch', 5.20686063127545), ('keeps', 4.764745016754615), ('working', 4.423377639013047), ('fucking', 4.121079708218993), ('minutes', 4.10632915457768), ('stop', 4.096002041221054)]

```
From topic 2:
love the game but crashes 50% of the time when I press the reform button

From topic 2:
Something is wrong with the resolution of the game when I play it. Everything looks
blurry and pixelated. The only way I can make this go away is by raising the
resolution all the way up to 3840x2160, where as my native resolution for my pc is
1920x1080. Was really excited to play this game and waited for it all month, only to
be highly disappointed by what is (for me) a game breaking bug.

From topic 2:
Can't recommend this buggy trash. I was having fun until I proclaimed myself emperor.
Then the game decided to crash every time I ended my turn or tried to save. Also the
game insists my version is modded despite never having downloaded a mod. Until this is
fixed, I won't be able to finish this game. Plz fix CA.

From topic 2:
Crash game every hour!

From topic 2:
```

Game is crashing all the time. Westing many hours of game because of that. Mostly when is enemy turn and you can't save afer a battle. I'm an adult and time is money. Path finding units in town annoing... "shift key" is not helping. Game is simple not ready yet.

From topic 2:
it was a very good game until it began to crash whenever the campaign map is loaded. no support at the SUPPORT-forum, neither from the sega support site. real shame.

We assign this topic2 as the ‘crash, bug problem’. The complaint in this topic is mainly about some technical problem in the game, like the game is always crashing or some bug for different system. This topic is also important, especially for the programmer of the game. Because they can get the feedback from this topic and fix the game in a proper way.

Our conclusion for the topic analysis is that the reason why we divide the people into two group is to get feedback from different attitude people. It helps us the extract the information in a proper way.

For group1, mostly are the praise for the game. We find two valuable topics that help the game designer know what most people are really interested for this game. As a result, they can make improvement on these features. Besides, for steam website, they can optimize their game recommend for this kind of customer. For instance, for each customer writes comment on the website. Steam can use our LDA model to assign a bunch of topics to this customer. Then give him the recommend basing on the topics assigned to him.

For group2, mostly are about the complaint of the game. We also find two valuable topics that tell the game designer the problem of the game. This is valuable because using our model will get rid of noisy complaint and let the designer focus on the really problem.

Conclusion

We obtained some unexpected interesting results. Our choice of game was unintentional, but we did choose a game with a cultural, historical perspective, not a fictional/magical background. It was interesting to us that part of the allure of this game is the cultural aspect as seen via topic modelling. We were also able to find bug issues and focus in on reviews that specifically state the issues, allowing game developers to look into the issues. Obviously, we’re only scratching the surface of NLP and in the future, we would want to train models that can parse phrases. Topic modelling simply based on keywords is jumbled mostly and not insightful.