Text Analytics Group Project - Text analysis of STEAM review dataset

Group members: Cheng, Yifan / Lee, Yun Cheng / Li, Edison / Li, Wenjun / Lin, Han Chun

1. Dataset

For the final project, we will analyze reviews of the games at the Steam Store. Unlike traditional products, digital products like games mostly can only be purchased online. Furthermore, game developers and publishers nowadays rarely release a demo before they launch the game. Therefore, content such as videos and reviews of gameplaying may be the important source that people can have a general idea about players' gaming experience before they buy the product. Reviews at Steam Store have a good metadata ecosystem for data analytics (e.g. reviews have a "helpful" score, "funny" score, number of hours the reviewer played the game…etc.).

The exact dataset we used is reviews of the game "Red Dead Redemption 2" at the Steam Store. It is a 19th century western-themed action-adventure video game developed and published by Rockstar Games. We chose the data from Dec/5/2019 - Apr/12/2020 and there are total 21044 counts of review and we used STEAM API by python package steamreviews to scrap data. The reason we choose this dataset is it has highly integrity and lots of useful columns to help us analysis the data, such as the review by the gamer and the game duration for each gamer. It is also well labeled by Steam community members. Steam community members are providing insightful information in terms of their opinions on games or other software.

a. Source: https://store.steampowered.com/app/1174180/Red_Dead_Redemption_2/

b. Code reference: https://github.com/alfredtangsw/steamvox

c. Columns description:

- Recommendationid: ID of the review
- Author
    - steamid: steam account of the reviewer
    - num_games_owned: numbers of games owned by the reviewer

    - num_reviews: numbers of reviews written by reviewer
    - playtime_forever: hours of playtime of the reviewer
    - playtime_last_two_weeks: hours of playtime of the reviewer during the late two weeks
    - last_played: the last time reviewer played the game
- Language: language of the review
- Review: the content of the review
- timestamp_created: review posted date
- timestamp_updated: review updated date
- voted_up: does the reviewer recommend the game or not
- votes_up: how many people think this review is useful
- votes_funny: how many people think this review is funny
- comment_count: how many people commented under this review
- steam_purchase: whether the reviewer purchase this game from Steam store or not
- received_for_free: does the reviewer received the game for free or not

2. List of novel questions

The application serves game developers. And the application tends to address the following business questions and generate insights for them to help them improve the product.

- What are some of the aspects that gamers care about, in general?
- What are some of the distinct aspects that different game groups care about?
- Which topic have the most positive/negative reviews?
- What are the common complaints? On what topics?
- What are the common bugs of the first month after launch?
- What are the possible reasons why veteran gamers do not recommend the game?

3. Analysis

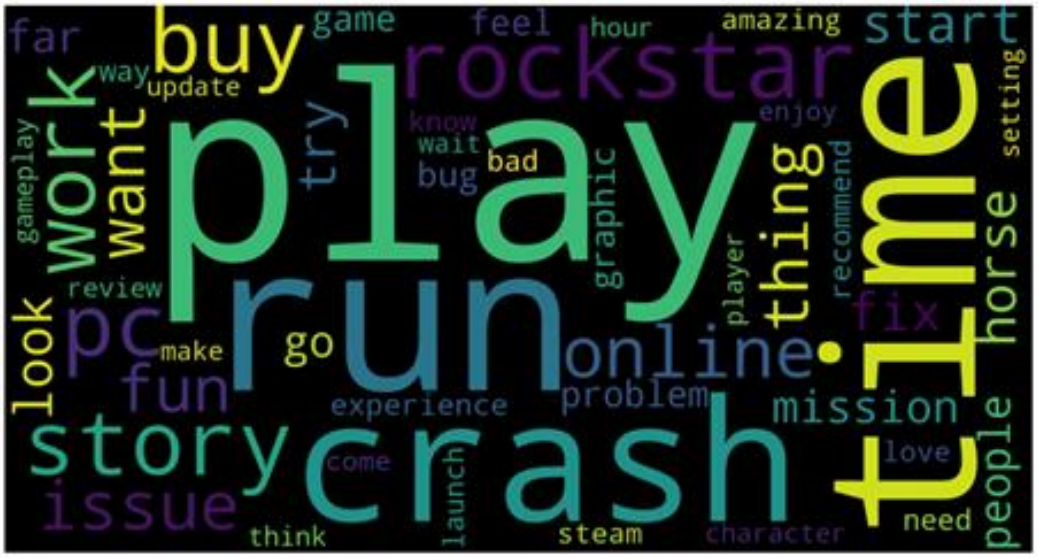
a. Data Cleaning

```
#extract usable data; save as new df
df_usable = df_raw[(df_raw['author_playtime_forever']>=3)& (df_raw_num['author_playtime_last_two_weeks']>=(10/60))] #setting cut
df_usable.head(2)
```

We cut off our dataset to grab review that has more chance is an insightful review. Attributes used include payer playtime, review length and language. We choose the reviews written by reviewers who have played more than 3 hours in the game and have played more than 10 minutes in the last two weeks. And all the reviews are greater than 4 words. The final dataset size for our analysis has 4201 usable reviews out of 21044 available.

Review Attribute	Condition
Player Playtime (when written)	1. Total >= 3 hours 2. Recent playtime >= 10 minutes
Review Length	>= 5 words
Review uniqueness	Duplicates from same user NOT allowed
Language	English only

b. Exploratory Data Analysis



First, we made wordcloud from reviews of 2 groups of players: players who have got free access to the game, and players who wrote the review in the very first month after the game launch, as reviews from these players provide valuable insights for developers to further improve the game. We also found that “crash” is one of the key words for those early players, which suggests improvements are very much needed at the early stage.

	Gaming Experience		Bug & Issues		Game Content & Design		Online Mode	
	Play time > 100 hr	Play time <= 100 hr	Play time > 100 hr	Play time < =100 hr	Play time > 100 hr	Play time < =100 hr	Play time > 100 hr	Play time < =100 hr
Positive	77.1%	66.1%	66.3%	68%	65.9%	58.6%	63.2%	67.6%
Neutral	5.8%	8.2%	8.8%	11.2%	8.2%	10.5%	11.8%	7.8%
Negative	17%	25.7%	24.9	20.8%	25.9%	30.9%	25.1%	24.6%

Second, we divided the data into loyal players and casual players, with the threshold being having played for 100 hours. In terms of gaming experience and content & design, the game is significantly better received among the loyal players over the casual players. And for bugs and issues, the reviews of the two groups don’t tend to differentiate much.

c. Topic Modeling

```
#credit to https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/ for this cell's code, which I modified to suit

topic_dict = {'0':'Gaming Experience',
              '1':'Bugs & Issues',
              '2':'Game Content & Design',
              '3':'Online Mode'
             }

#created a dictionary so I can show topic names instead of numbers, without any complicated code

def format_topics_sentences(ldamodel=lda_model, corpus=corpus, documents=documents):
    # Init output
    sent_topics_df = pd.DataFrame()

    # Get main topic in each sentence
    for i, row in enumerate(ldamodel[corpus]):
        row = sorted(row, key=lambda x: (x[1]), reverse=True)

        # Get the Dominant topic, Perc Contribution and Keywords for each document
        for j, (topic_num, prop_topic) in enumerate(row):

            if j == 0:  # => dominant topic
                wp = ldamodel.show_topic(topic_num)
                topic_keywords = ", ".join([word for word, prop in wp])
                sent_topics_df = sent_topics_df.append(pd.Series([topic_dict[str(topic_num)], round(prop_topic,4), topic_keyword

            else:
                break

    sent_topics_df.columns = ['Dominant_Topic', 'Perc_Contribution', 'Topic_Keywords']

    # Add original text to the end of the output
    orig_contents = pd.DataFrame(sent_df_ready[['original_text','token_sentences']])
    docs = pd.Series(documents)
    sent_topics_df = pd.concat([sent_topics_df,docs, orig_contents], axis=1)
    return(sent_topics_df)
```

Four topics were picked in order to achieve the highest coherence score of 0.64, the name and key words of each topic are provided below:

- Gaming Experience: Story, graphic, Rockstar, character, experience, gameplay, bug, world, open_world, single_player
- Bug & Issues: pc, crash, issue, problem, setting, steam, rockstar, performance, fix, run
- Game Content & Design: horse, way, people, mission, character, story, gun, weapon, world, animal
- Online Mode: mission, player, server, bug, rockstar, money, hacker, friend, camp, problem

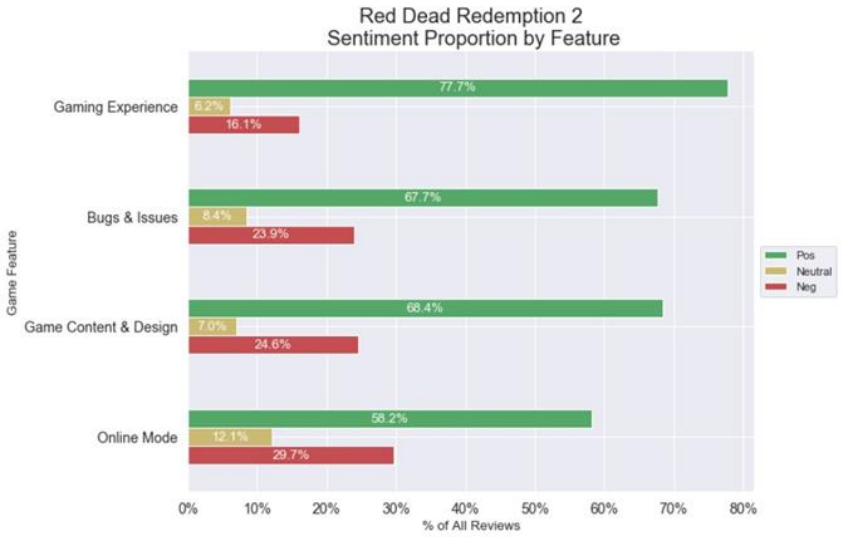
d. Sentiment Analysis

```
syntok_dom_topics['compound_sentiment'] = syntok_dom_topics['original_text'].map(lambda x: anakin.polarity_scores(x)['compound'])
syntok_dom_topics['int_scores']=sent_score_int(syntok_dom_topics['compound_sentiment'])
syntok_dom_topics.head(10)
```

	review_number	dominant_topic	topic_perc_contrib	tokens	token_sentence	num_tokens	topic_keywords	original_text	compound_sentiment	in
0	0	Online Mode	0.6938	[review, ass, minute, camp, bug, money, infini...	[online, review, only, simply, put, it, sucks,...	18	mission, player, server, bug, rockstar, money,...	online review only ... simply put , it sucks a...	-0.8968	
1	1	Gaming Experience	0.4671	[connection_issues, friend]	[it, is, wonderful, multiplayer, and, singlepl...	2	story, graphic, rockstar, character, experienc...	it is wonderful multiplayer and singleplayer g...	-0.3979	
2	2	Bugs & Issues	0.5936	[people, review, works_fine, time, pc, spec, t...	[people, claim, lot, in, reviews, that, they, ...	10	pc, crash, issue, problem, setting, steam, roc...	people claim lot in reviews that they cant run...	0.7239	
3	2	Gaming Experience	0.6778	[60_hours, story, ton, challenge]	[60, hours, through, am, just, over, 60, done,...	4	story, graphic, rockstar, character, experienc...	60 hours through am just over 60 % done the st...	0.0772	

For the sentiment score, we score and aggregate by paragraphs, because it is close to how humans read reviews, identify topics, and feel the sentiment of the reviewer. Score range is from -1 to +1, score greater than 0.1 will identify as a positive review, between -0.1 to 0.1 will identify as neutral and less than -0.1 will identify as a negative review.

Based on the sentiment analysis, the player's overall experience for the game is positive. However, the online mode might have more bug than PC mode so received more negative review within that topic. Furthermore, the game has lots of bugs and optimization issues when released on the Steam so it received lots of negative reviews after the first couple days of game launch.

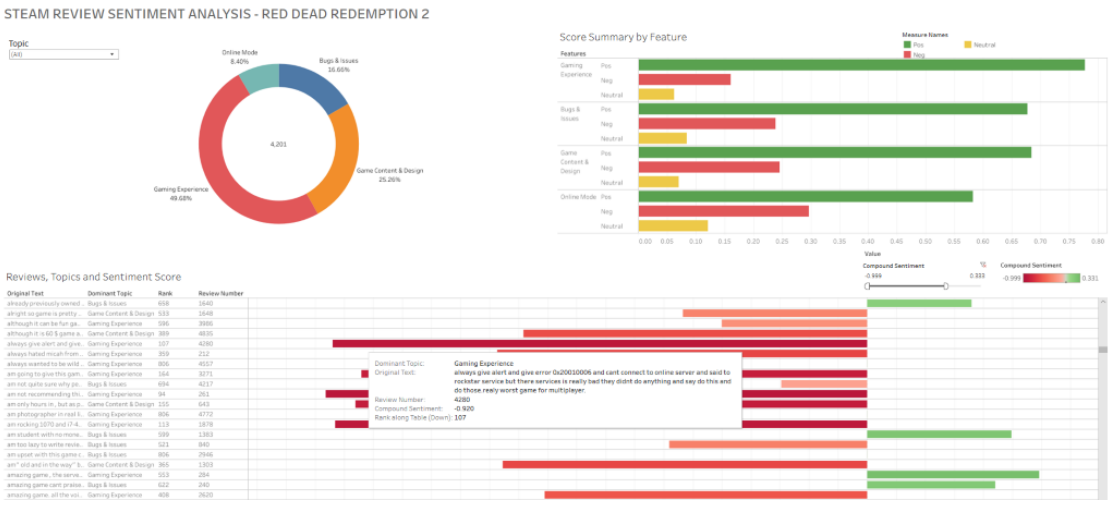


4. Application

Description:

The application will serve as a web app for game developers to explore the review data in the Steam Store. Right now, the web app is hosted on GitHub as proof of concept (Link). Developers can search for specific games based on genre or tags, the app will show text analytics results of the positive and negative review. This app will help game developers improve the game they are developing by identifying common complaints; it will also help those developers in beginning stages figure out what type of games have a higher possibility of being recommended by players. In addition, the app is expandable to include comparison between different games, in terms of score, genre and typical positive/negative reviews.

Screenshot of the web app:



5. Conclusion

At the end, we feel that text analytics provides us with powerful tool to capture the sentiment or attitude of the studied object. With this tool we are able to accomplish tasks such as to study word frequency distributions, pattern recognition, link and association analysis and visualization. However, at the same time, we also feel our current text mining techniques are not adequate enough as it fails to understand a joke or sarcasm, both of which appeared in our dataset quite frequently. Additionally, the Web App can be expanded by including different games and longer time periods.