SWE 522

FINAL PROJECT

METACRITIC EDITORIAL GAME SCORE COMPARISON WITH CROWD RE SENTIMENT ANALYSIS

GROUP 8:
Emre Çalık - 2019719198
Berk Büyükyıldız - 2020719030

Contents

Motivation

As a group, we decided to test hypothesis of end user reviews are parallel to expert opinions in CrowdRE field. Today we are dealing with huge amount of user feedbacks. These feedbacks are important to develop and improve performance of mobile applications & games.

Another method to get feedback is asking for expert opinion. There are testing experts whom could also test and rate your product. Undoubtedly, expertise is very important. However, processing end user feedback could be much better time to time in case you need to add new feature.

Our main motivation to start this project is, observing the gap between user reviews and expert opinion. Thanks to CrowdRE we could gather so much data about popular products. Our initial assumption is whether there are extremist opinions about product, average sentiment score of user reviews is parallel to expert rating.

For testing this hypothesis, we could find a good case to implement cloud cognitive service solutions and use state of the art python libraries to analyze data. Gaining literacy in data science field is another motivation for us.

Problem

In these days hiring expert for reviewing is very expensive. For small size start-ups it is hard to hire experts to review their product to give feedback. Thanks to CrowdRE, we could get benefit of user feedbacks for our products. There are hundreds of opportunities to get feedback about your product.

However, it is always concern that, whether user feedbacks as valuable as expert feedbacks or not ? To assess current situation of CrowdRE feedbacks we decided to implement this project for comparison these two different kind of feedbacks.

As we put ourselves in the shoes of a game creating agency, we would like to know whether we could trust user feedbacks as much as expert critics. In order to assess this situation, we would like to MetaCritic website. MetaCritic is also like steam, let you view, review and share games. There are several reasons why we have chosen MetaCritic:

- MetaCritic is a huge inventory of games, you could find even less popular games on website.
- There are review option for users. MetaCritic has big and keen user base. They have so many users could give opinions almost at expert level
- Despite users MetaCritic has experts to rate the games. MetaCritic editors share their overall rating about games for each game. It is very correct place to find both end-user reviews and expert editor ratings.
- Their page routing structure is suitable for web scraping

So to implement our CrowdRE solution, we would like to implement a NLP system that gets MetaCritic user reviews by scraping and compares it editor scores.

Dataset Description

As we have used data set to get MetaCritic editorial scores, we used web scraping methods to get user reviews. So data set only serves our purpose of getting editorial ratings for each game we will scrap its user reviews.

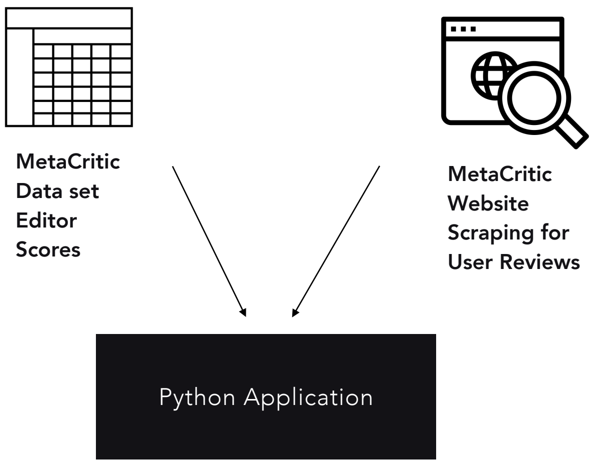

Figure 1: Input Streams and Data Sources of Application

Dataset has 16 columns and 16719 rows. In the dataset there are name of the game, platform which game has been released, some sale data (by continent), MetaCritic editorial score, Genre, Publisher agency etc.

Our main considered parts of data is Critic_Score. This column gives us the rating of the game which has been provided by MetaCritic game editors. After reading data from CSV file we create a separate data frame which include Name and Critic_Score attributes of data.

	Name	Platf	Year_of_Relea	Genre	Critic_Sco	Critic_Cou
0	Wii Sports	Wii	2006.0	Sports	76.0	51.0
1	Super Mario	NES	1985.0	Platfor	NaN	NaN
2	Mario Kart Wii	Wii	2008.0	Racing	82.0	73.0
3	Wii Sports	Wii	2009.0	Sports	80.0	73.0
4	Pokemon Red/	GB	1996.0	Role-	NaN	NaN

Table 1: Example Data Rows from Data Set

As it can be seen in table 1, there are some NaN values in Critic_Score column. Since we need this information we apply dropna() transformation while creating our new data frame. By applying this we drop rows which have NaN values in Critic_Score column. After this drop operation row number decreased to 8137.

	Name	Critic_Score	Platf
85	The Sims 3	86.0	PC
13	World of Warcraft	93.0	PC
19	Diablo III	88.0	PC
21	StarCraft II: Wings of Liberty	93.0	PC
28	Half-Life	96.0	PC

Table 2: After Dropping Unnecessary Columns and NaN values

Methodology

In order to scrap user reviews from MetaCritic website, we have used BeautifulSoup library to scrap web pages. Our initial tests consists two games. For each game we change the url we scrape and automatically scrape MetaCritic website according to given list of games.

In project we scrapped MetaCritic for Call of Duty: Modern Warfare 2 and GTA V games. We could use more games and system automatically process when the name of the game added to the list. However, we use Azure Cognitive Services and our Free Tier Account limits us to use services only for 2 games which will be mentioned more detailed in the next parts

When we get user reviews for each game one by one, we need to calculate sentiment score for each review. At this part, we have used Microsoft Azure Cognitive Services - Text Analytics API. Because, in case we need to do it we have to label game data in order to calculate sentiment . Since we have very limited time for that kind of workload using Azure API's provided solution for us.

Also not only for time limits, but also their capacity and accuracy would be considerably higher than our labelled data set. They gather billions of sentences and train their model with this huge dataset. By using Cognitive Services API in Jupyter Notebooks, we had great experience for our future projects.

For each review, we make call to Azure Cognitive Services Text API, Our review parsed to its sentences and sent to the API as document. After calculating sentiment score, Text Analytics API returns the response to the python in a wrapped object. To be able use this feature first Azure Text Analytics and Azure Client shall be installed via pip.

In response object, there is overall summary of review (Such as: positive, mixed, negative etc.) . Beyond summery, there is percentages in three labels : Positive, Neutral and Negative. While evaluating sentiment score of sentence Azure calculates three different labels of sentence. Each three labels' value can be between 0.0 and 1.0. And sum of this three labels must equal to 1.0. For example: "Positive:0.37,Neutral:0.63,Negative:0.0".

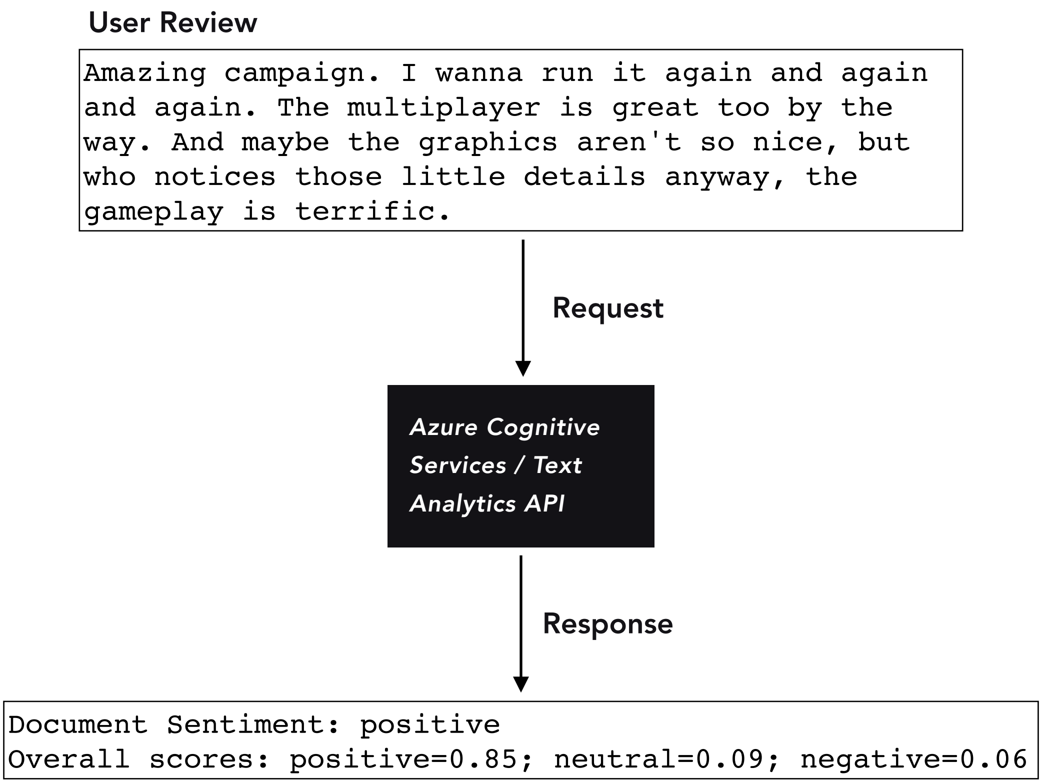

Figure 2: Sentiment Score Calculation with MS Azure

Azure API calls our limiting factor in order to increase number of games. We use Free Tier Plan and in Free Tier Plan we could make 5k API Calls per month. In one execution we call API for 110 times for single game in single execution of notebook. And we need to execute many times while development process. So not to exceed our Free Tier limits we limited our sample set with two games. But with better plan such as 2 Million API Calls per month we could analyze more than 100 games.

After getting sentiment score in terms of three different labels as seen in Figure 2, we need to develop a methodology to get an overall score between 0-100. Because MetaCritic editorial scores vary between 0-100. However our sentiment score could be only 1.0, and there are three different parameters.

For comparison we have used two different scoring algorithms:

- Fifty Percent Neutral

Overall Sentiment Score = (100 x Positive) + (50 x Neutral) + (0 x Negative)

- Thirty Percent Neutral

Overall Sentiment Score = (100 x Positive) + (30 x Neutral) + (0 x Negative)

You get full point for positive review and in first approach half of full point for Neutral reviews in the second approach for comparison, thirty percent of full point for Neutral review. In both approaches you do not get point for Negative parts. There could be some other approaches which could decrease sentiment by looking Negative but base of editorial score is 0 so sentiment shall not be below 0. If your sentiment is fully positive (Positive : 1.0, Neutral : 0.0, Negative: 0.0), you get 100 points as sentiment score. Sentiment score of comment which is in Figure 2 would be 100 x 0.85 + 50 x 0.09 + 0 x 0.06 = 89.5 in the first approach.

Also for fair comparison, we have calculated sentiment scores of games twice. One with 10 user reviews and one with 100 reviews belong to the same game. In the Conclusion part results will be discussed.

Conclusion

After calculating sentiment score for both games by applying CrowdRE principles with Azure implementation by considering 10 reviews for each game:

Name	Overall_User_Score_Fifty	Overall_User_Score_Thirty
Call of Duty: Modern Warfare 2	49.55	46.97
Grand Theft Auto V	74.70	73.66

Table 3: Sentiment Scores for Both Games with 10 reviews

It can be seen that by applying first method (Fifty percent of full points for Neutral Labels) sentiment score of Call Of Duty: Modern Warfare 2 is 49.55. In second method which user get thirty percent of full points for their Neutral parts of reviews, score decreases to 46.97. While calculating that we get sum of all sentiment scores for the game and then divide the sum to the count of reviews. This gives us average sentiment score for each game.

Same calculation methods indicate significantly higher results for Grand Theft Auto V. With first fifty percent method, average sentiment score for GTA V is 74.70 and thirty percent scoring method it decreases 73.66.

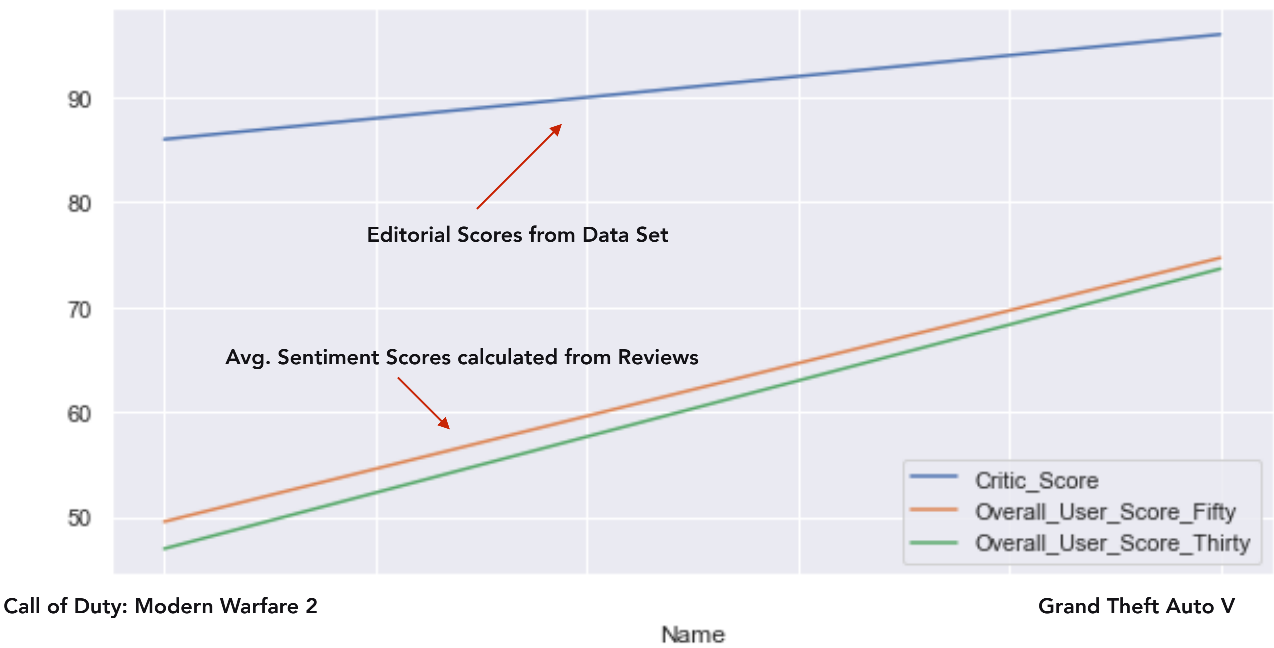

Figure 3: Editorial Score comparison with Sentiment Scores (10 reviews)

When we consider same scores for 100 reviews for each game

Name	Overall_User_Score_Fifty	Overall_User_Score_Thirty
Call of Duty: Modern Warfare 2	39.463636	38.350909
Grand Theft Auto V	74.818182	74.025455

Table 4: Sentiment Scores for Both Games with 100 reviews

With 100 reviews, sentiment score of Call of Duty: Modern Warfare 2 decreased and gap between editorial score increased. Sentiment score of Grand Theft Auto V almost did not change.

Figure 4: Editorial Score comparison with Sentiment Scores (100 reviews)

As a result, there is still gap between editorial scores and sentiment scores. However this could be related with our scoring algorithm. The hopeful part is parallelization between editorial scores and sentiment scores. Editor score of GTA V is quite higher than Call of Duty: Modern Warfare 2. When we consider sentiment scores, they are also aligned with this trend. Sentiment score of GTA V is also higher than sentiment score of Call of Duty: Modern Warfare 2. This gives us motivation for further research projects with different scoring algorithms.

Bibliography

- Anonym. (2020, April 12). How to: Sentiment analysis and Opinion Mining Microsoft. https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/how-tos/text-analytics-how-to-sentiment-analysis?tabs=version-3-1

- Geek_Dude. (2019, February 26).

Analyzing Tweet Sentiment with Azure (Python), GeekTechStuff. https://geektechstuff.com/2019/02/26/analysing-tweet-sentiment-with-azure-python/

- Rush Kirubi. (2016, December 30).

Video Game Sales with Ratings, Kaggle. https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings