Project 3

Subreddit data mining project

Subreddit #1: Overwatch.
Source: r/overwatch

Subreddit #1: League of legends.
Source: r/leagueoflegends

Approximately 4,000 submissions were analyzed for this project.

Submitted by Victor Voskovsky.

04/24/2020

Introduction:

a. **Assume you are an analyst working for a game developer:**

Your manager comes to you and and tasks you with pulling 2,000 submissions each from the Overwatch and League of Legends subreddit then build a logistic regression model with NLP to predict generate a sentiment analysis towards our competitors.

Once the model is built out, we can run further analysis into what else we could learn from this data & model.

Goal:

We want to figure out the sentiment of fans towards out competitors, and we want to be able to analyze their fan base and their sentiment towards our competitors. We can use Reddit for this assignment since it has plenty of text data we can look at.

Steps:

a. Pull in around 2,000 submissions from both subreddits

b. Create a notebook to be reused for data mining.

c. Create a notebook to be used for data cleaning.

d. Create an NLP logistic regression model

Data collection:

For my data collection I ended up using the reddit shiftpush API. This ended up being the optimal tool to mine the data we needed from Reddit.

I was able to easily pull the 4,000 submission that we set out to collect. Both subreddits have 2,000 submission, and 1k from each submission was taking from the most recent submissions and the second 1k was taken from 300 days after first 1k.

Both subreddits were pretty strong, with League of Legends leading with 4 million members and Overwatch with 2.9mil.

Some of the things I did to mine the data:

Ended up taking the most recent 1,000 + the most recent 1,000 posts after 300 days. I felt like this would give me a better distribution of data, since it spans two years instead of one.

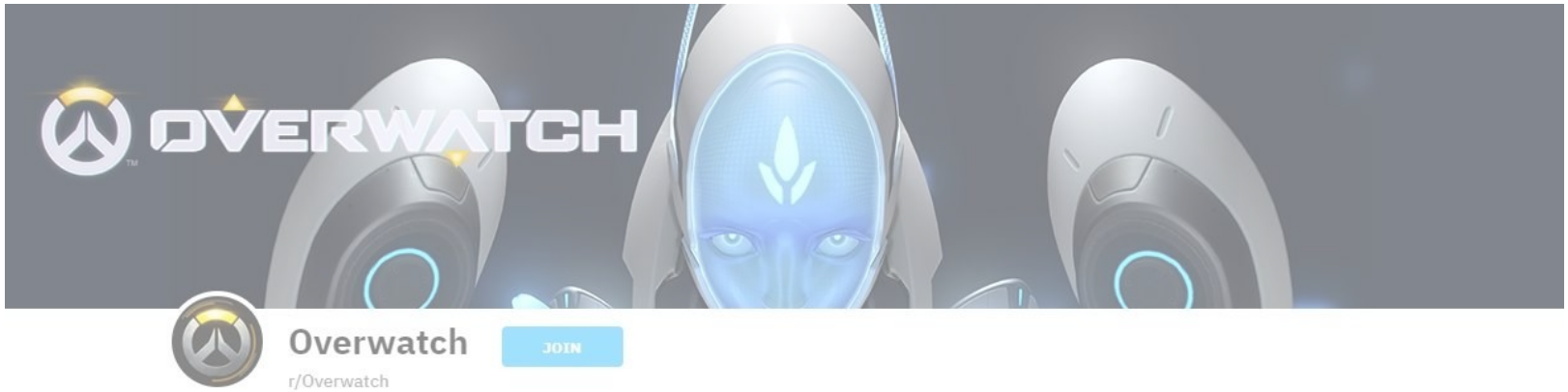
I considered using a title keyword to pull posts which only mention the game itself, I tried this but, I felt that my collinearity would be too high in this case and might make my model worse.

Data Cleaning

I ended up doing some extensive data cleaning; data generated on Reddit can be poretty rough, especially given the nature of these specific subreddits.

I extracted only the title and text from each submission, I then dropped [removed] and [deleted] submissions, emojis, punctuation and I tokenized the individual words.

I felt that the submission text and title would be enough to create a good model.



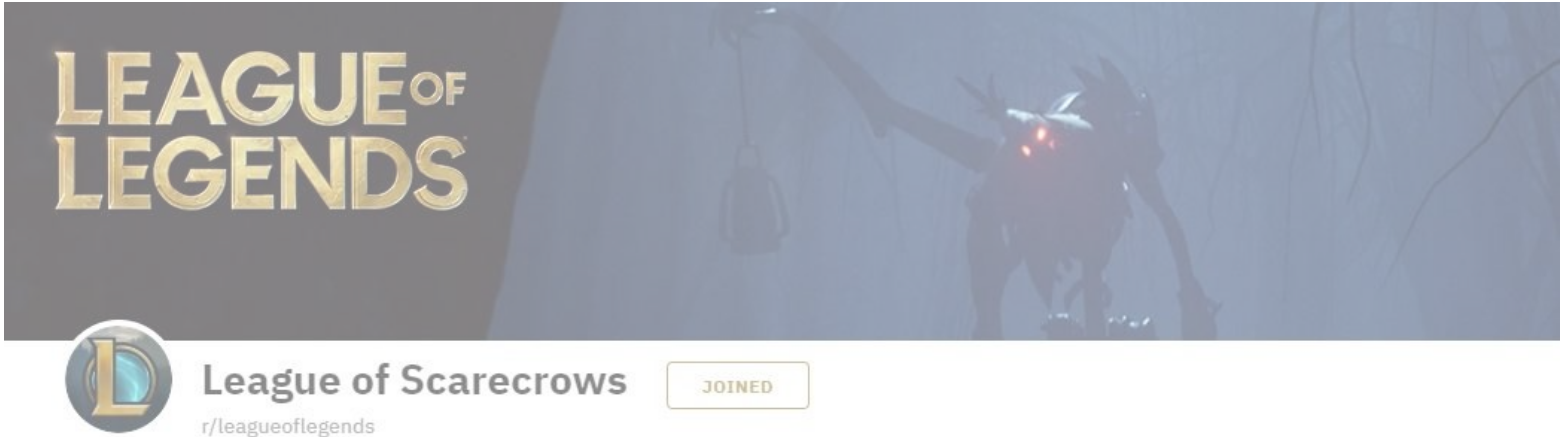
Pre-tokenizing:

subreddit	text	target
leagueoflegends	Game started and everyone had an invisible cha...	0
leagueoflegends	So I have to win 2 of 3 games to be promoted. ...	0
leagueoflegends	I broke hail of blades	0
leagueoflegends	I was thinking of what champs in LoL best repr...	0
leagueoflegends	Steve' secret message revealed?	0

subreddit	text	target
Overwatch	Solo queue is bad for my mental health...	1
Overwatch	I've always wondered about Sym's arm and have ...	1
Overwatch	my 14 yo. sister in law painted this, and i th...	1
Overwatch	my 14 yo. sister in law painted this picture, ...	1
Overwatch	that is allMy replays list is completely red	1

Lemmatizing and Stemming

After examining the data further, I actually ended up only lemmatizing the data. I wanted to use the full words for this model and I felt that stemming would potentially hurt some technical/game related words.



Model fitting: Logistic Regression

My hypothesis was that Logistic Regression would be the best method to use for this model.

Resules:

Model score on Training set: 0.9779

Model score on Testing set: 0.8742

Model fitting: TF-IDF

I wanted to fit a Naive Bayes model but in the interest of time i decided to go with a TFIDF, will run Naive-Bayes in the future.

Model performance with TFIDF on training set: 0.9649

Model performace with TFIDF on testing set: 0.8560

Conclusion & use cases:

ITFIDF model actually turned out to be slightly better than my logistic model, but i suspect that i can mess around with my data cleaning portion to make logistic generate better results.

For use cases, I think that this model could be useful in terms of analyzing our competitors, as opposed to trying to predict given submissions.

For example, we can calculate the most common words from each subreddit, we can run a sentiment analysis on a given competitor and in general we can calculate some important metrics to compare with our own subreddit.

Lastly, I would love to see what makes a top post in each subreddit.