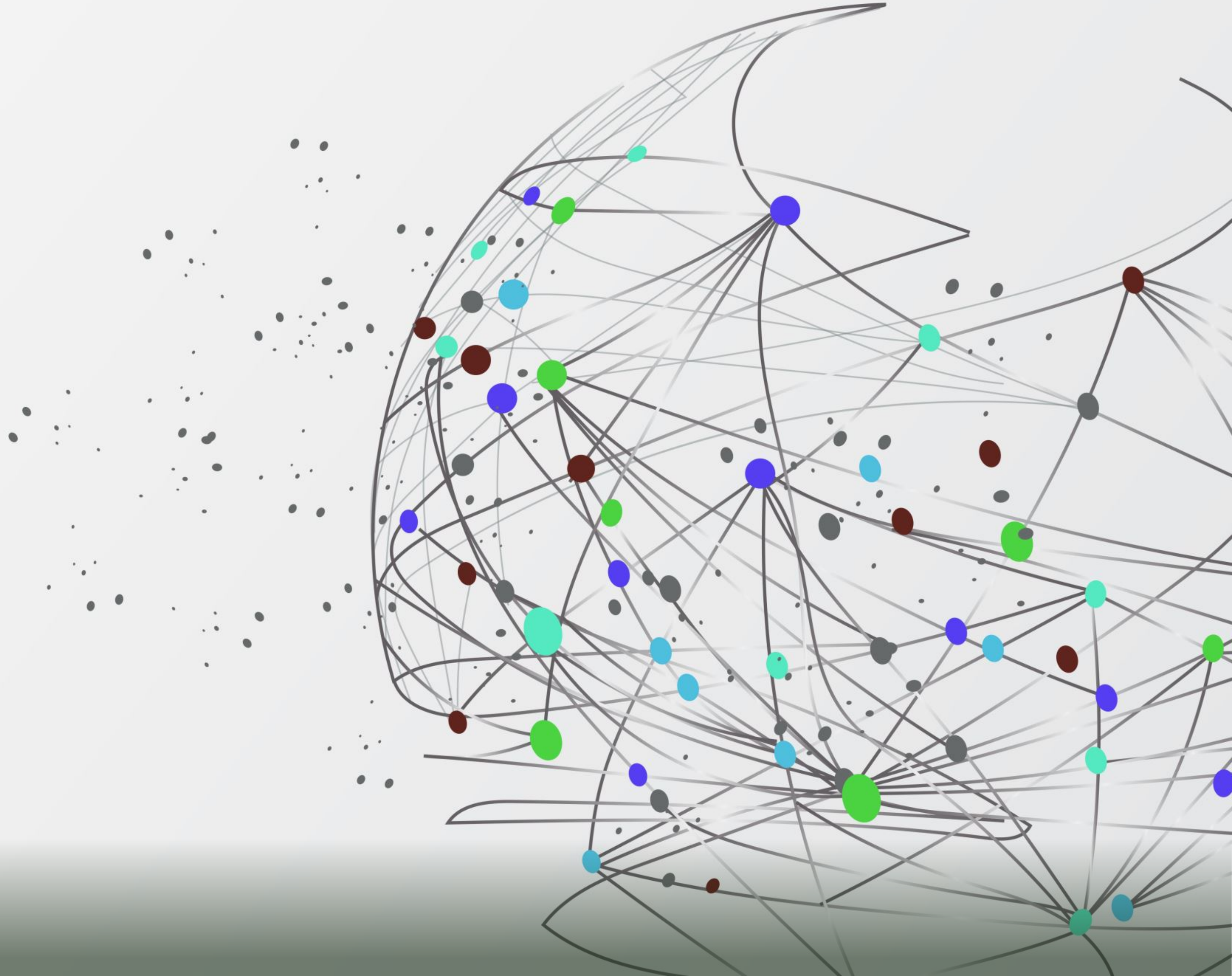
Twitter Sentiment Analysis: Nintendo E3 2018

By: Jason Zhou

With the help of: A J Sanchez

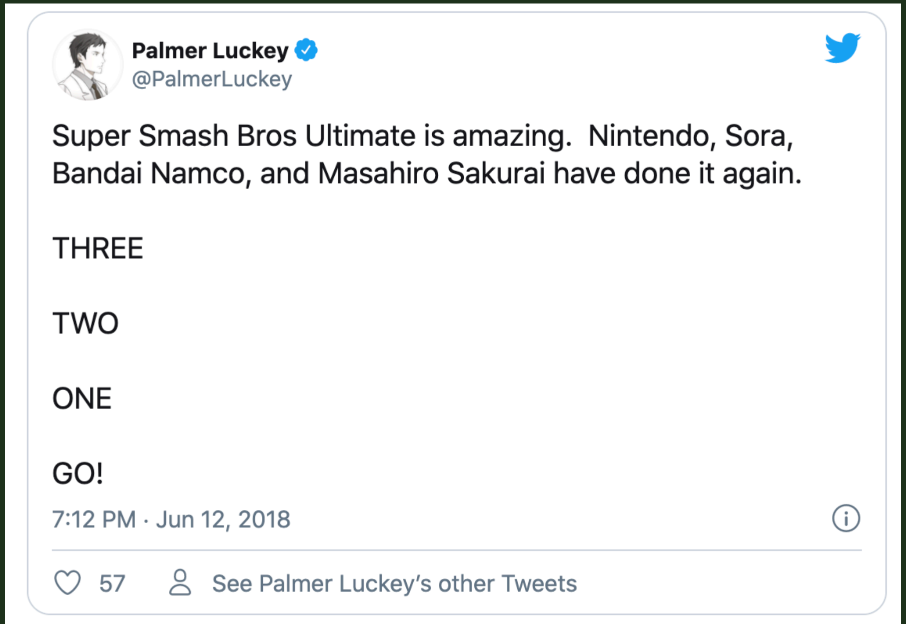


Introduction

- E3 (Electronic Entertainment Expo) is a major annual event in the video game industry
- Major companies in the industry attend the event to announce new releases, including Nintendo
- Reactions to new titles get posted on Twitter
- Major upcoming Nintendo titles: Super Smash Bros Ultimate, Fire Emblem: Three Houses, Super Mario Party

Business Problem

- Sentiment analysis of tweets responding to the announcement of the major upcoming titles: Smash Ultimate, Fire Emblem, Mario Party
- Build models that accurately label tweets as positive or negative reactions, one separate model for each game



Data Science Approach

1

Treat problem as binary classification problem, two classes “positive (0)” and “negative (1)”

2

Extract tweets corresponding to each game, based on hashtags. Separate into three distinct data sets

3

Label tweets using python package TextBlob, TextBlob takes a string and outputs polarity score between -1.0 and 1.0. Scores >= 0 categorized as positive, scores < 0 categorized as negative

4

Train classification models for each game, choose best performing model for each game

5

Set aside sample of data set to manually label using my own judgment, compare model labeling to mine, evaluate models based on classification report scores

Data Acquisition

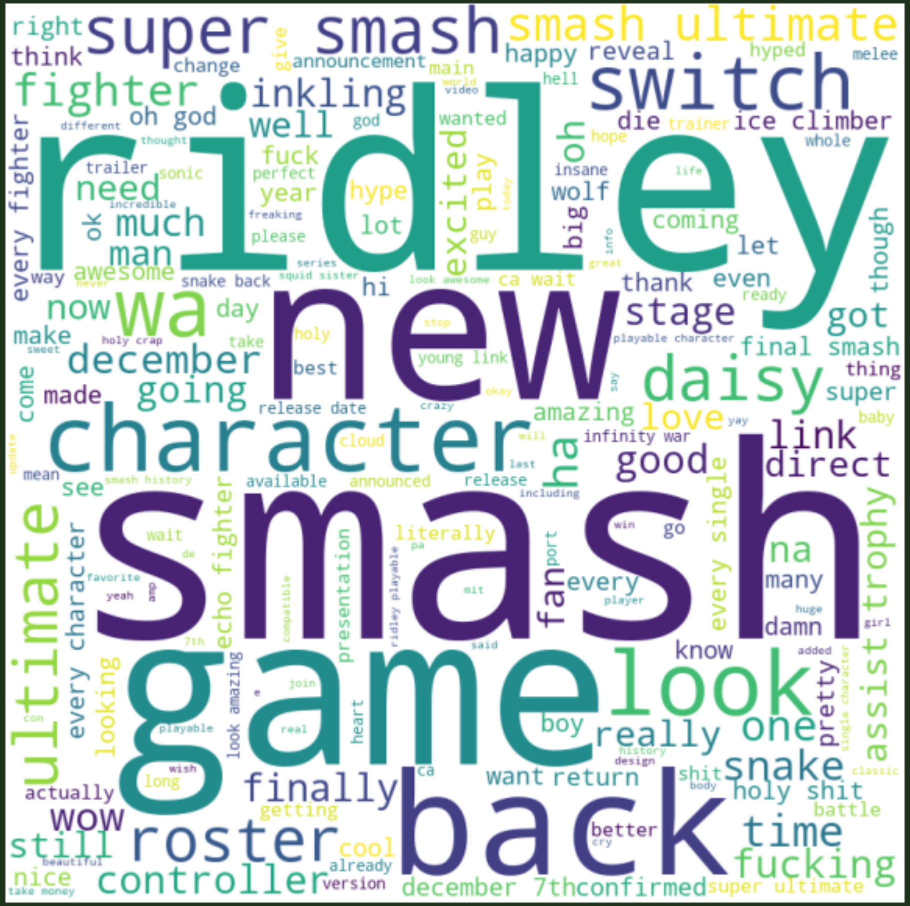
- Data set found on Kaggle, 100,000+ tweets captured during Nintendo’s presentation at E3 2018, using keywords #NintendoE3 and #NintendoDirect
- Tweet data encoded in JSON format, with predefined attributes and values
- Attributes of interest: ‘text’ and ‘entities’, which includes hashtags

Data Wrangling

- Processing text bodies of tweets
    - Removing URLs, punctuation, @s, hashtags, emojis, words not in the dictionary
    - Standardizing words/tokens, lemmatizing, lowercasing
- Categorizing tweets by game using hashtags
    - Group similar hashtags together, eg: #supersmashbros, #smashbros, #smashultimate, #ssbu
    - Gather all tweets related to a game, based on the similar hashtags
- Final columns used: ‘text’
- Detecting and removing foreign tweets by checking words in English dictionary, and by using list of foreign language keywords, eg: (el, de, le, un)

Exploratory Data Analysis

- Initial data visualization using word clouds, looking for individual words that clearly indicate positive or negative sentiment
- Feature Extraction with Vectorizers, examining correlations between token frequency vectors, identifying collinear vectors



Preprocessing

- Setting aside sample of each collection of tweets to be manually labeled by myself, using train_test_split with stratification on the labels
- Labeling the rest of the data using TextBlob, positive labeled as 0, negative labeled as 1
- Checking for data imbalance

Labels by TextBlob	Smash Bros	Fire Emblem	Mario Party
Positive	11861	1534	829
Negative	674	29	53

Baseline Modeling - Recall Scores

- Models: LogisticRegression, RandomForestClassifier, XGBClassifier, SVC

- Game1: Smash Bros Ultimate
- Game2: Fire Emblem: Three Houses
- Game3: Super Mario Party

	Game1	Game2	Game3
LogReg	0.699	0.500	0.710
RanFor	0.816	0.250	0.825
XGB	0.770	0.500	0.800
SVM	0.699	0.250	0.715

Hyperparameter Tuning

LogisticRegression:

solvers, c_values

RandomForestClassifier:

n_estimators, max_features

XGBClassifier:

n_estimators, learning_rate

SVC:

kernel, C

Extended Modeling – Recall Scores

- Before Tuning

	Game1	Game2	Game3
LogReg	0.699	0.500	0.710
RanFor	0.816	0.250	0.825
XGB	0.770	0.500	0.800
SVM	0.699	0.250	0.715

- After Tuning

	Game1	Game2	Game3
LogReg	0.787	0.308	0.817
RanFor	0.823	0.116	0.794
XGB	0.742	0.567	0.735
SVM	0.773	0.517	0.766

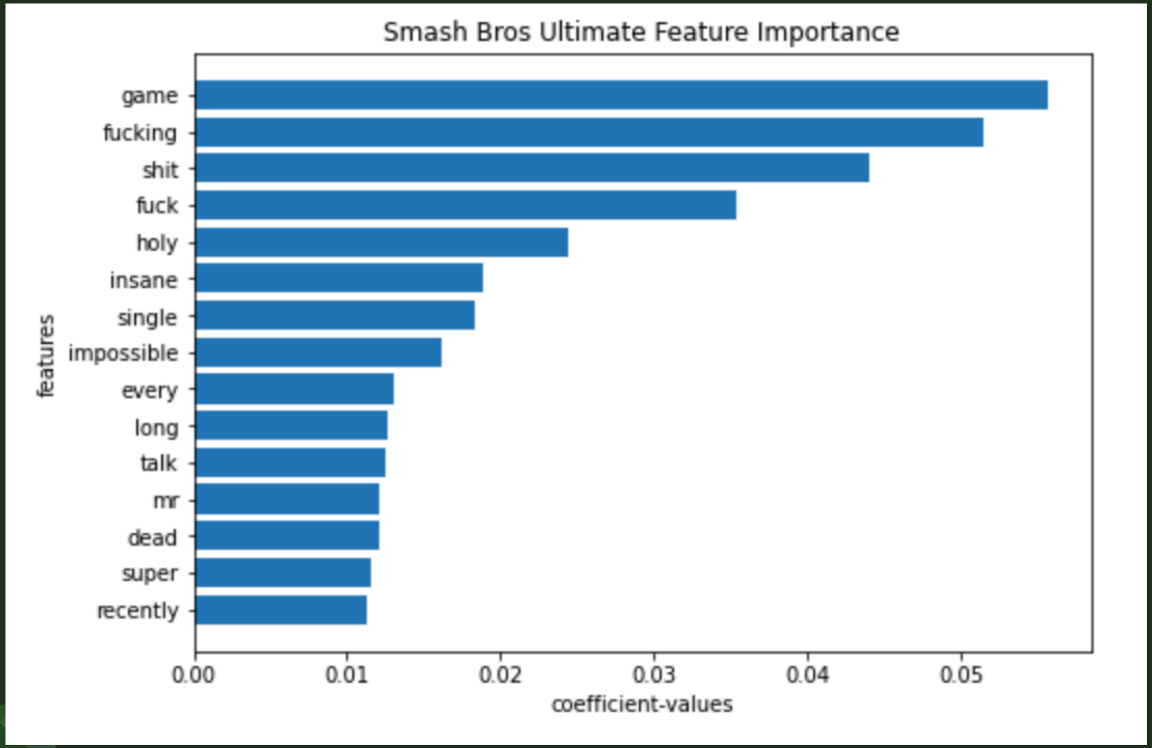
Final Models By Game

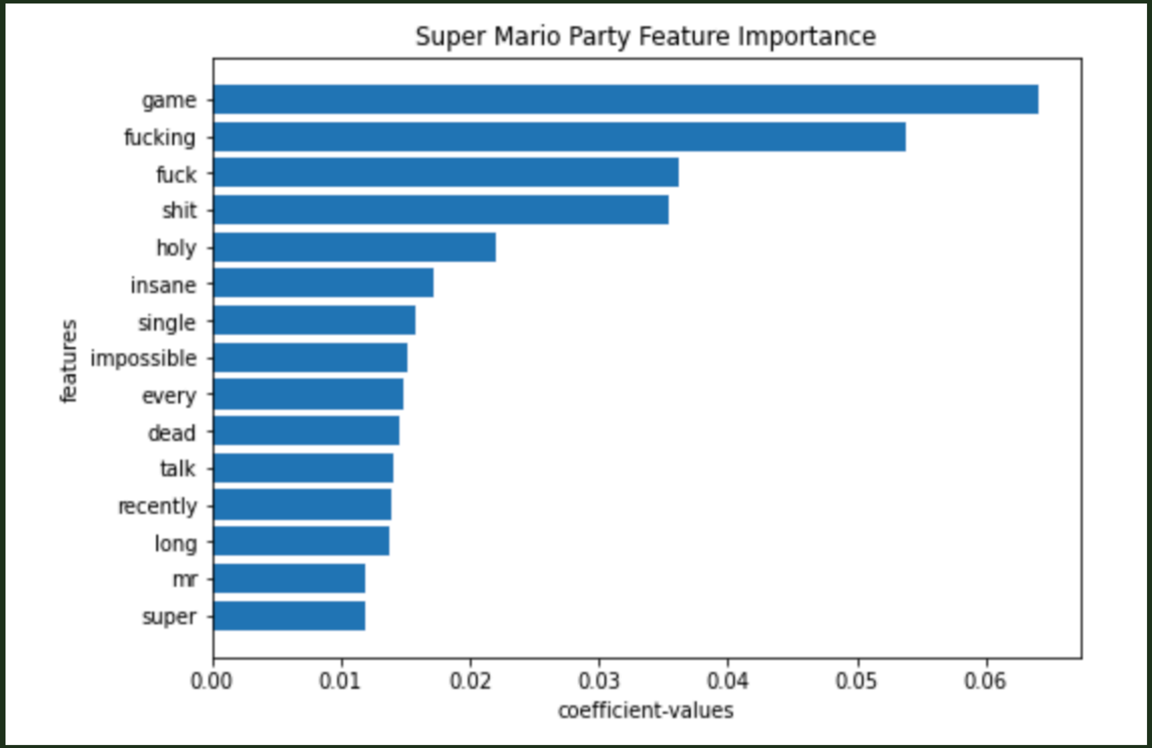
- Super Smash Bros Ultimate: Random Forest
- Fire Emblem: Three Houses: XGBClassifier
- Super Mario Party: Random Forest

Comparing Model Predictions with Manually Labeled Data

	Smash Bros	Fire Emblem	Mario Party
Accuracy	0.960	0.970	0.925
Precision	0.800	0.000	0.000
Recall	0.222	0.000	0.000
F1	0.348	0.000	0.000

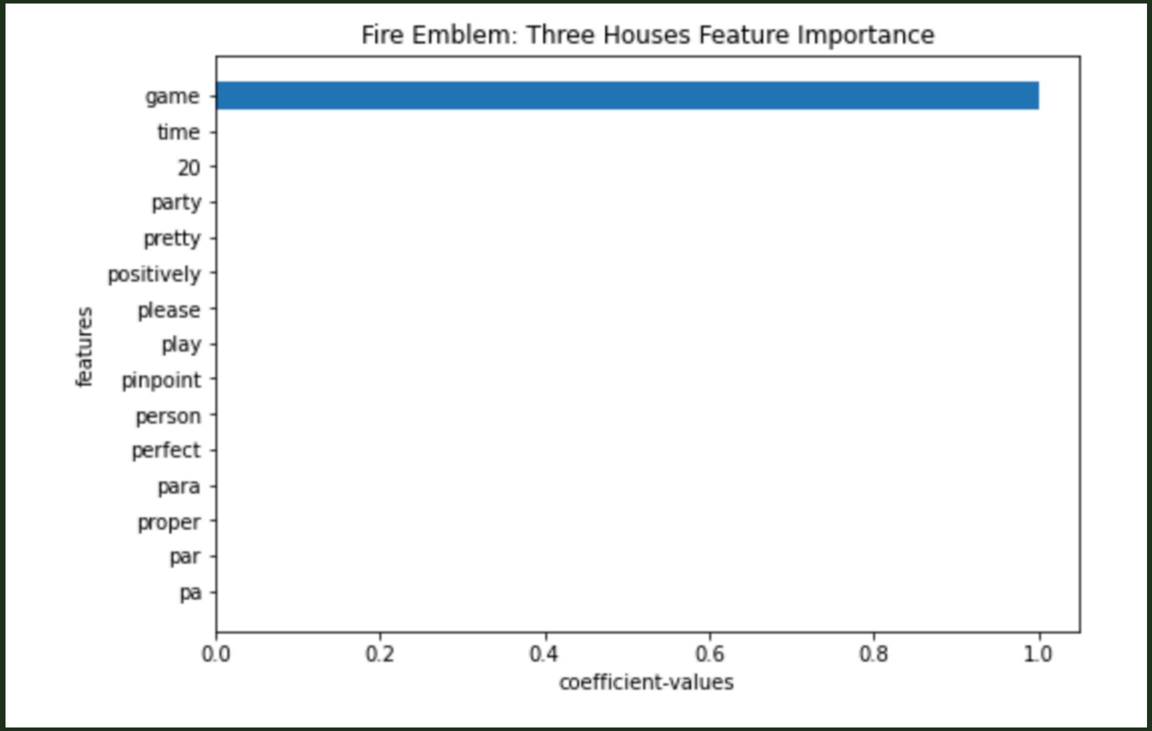
Feature Importance





Feature Importance

- Hmmm… why? How?
- Most imbalanced data set: 53:1 positive: negative
- Majority duplicate rows ,only 200 unique tweets out of 1563 total tweets
- One tweet was retweeted 1178 times, in other words the data set has 1178 identical rows



Reasons for Modeling Failure

- The language and content of tweets is more unfocused and broad compared to for example, formal reviews
- Harder for algorithms to work with documents that don't adhere to more standard language
- Bag-of-words representation of tweets is too simplistic and unnuanced

Future Improvements

- Processing and evaluating images as well
- More complex text representations than just bag-of-words representations, include phrases, sentence structure, grammar, punctuation
- Better sampling methods suited for working with imbalanced data
- Trying more models, tuning more hyperparameters