


 erichansen / dsc-phase-4-project_work Public MIT license 0 stars  0 forks Star Unwatch ▾[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#) main ▾

...



erichansen word salience graphics ...

6 hours ago  16[View code](#) README.md

dsc-phase-4-project_work

Natural Language Processing: Sentiment Analysis for Steam video game reviews

Motivation

Classification/prediction of text reviews is a useful tool for many different fields and is a perfect application of Natural Language Processing (NLP) techniques within Machine Learning/Deep Learning.

In this case, we will attend to video game reviews on the Steam platform (a digital distribution service and storefront developed by Valve). Launched in 2003, users are able to purchase and download games from Steam, as well as use their cloud storage to backup saved games for use across multiple devices. As of March 29, 2022, Steam reports over 27 million concurrent users at peak logged in to their service, with over 120 million users total.

Currently, users are able to supply "would recommend"/"would not recommend" binary ratings, along with text reviews of games on Steam. This provides us with a data set of labeled data from which to begin.

Stakeholders in this process include (but aren't limited to)

- Valve (or other digital distributors, e.g. Epic Games, Apple App Store, Google Play),
- video game developers/publishers,
- video game consumers (including reviewers, critics, players, streamers, etc.),
- computer hardware manufacturers (knowledge about what games are popular could drive video card architecture, e.g.)

- cultural linguists (due to the nuanced and specialized vernacular exhibited by video gaming communities).

Business problems that such an investigation could illuminate include:

- validation of new ratings (i.e. checking consistency between user text and user binary rating). This could be helpful for several reasons.
 - Prompting users to double-check that they provided the intended binary rating, if their text review seems inconsistent.
 - Notifying Valve of potential abuse/manipulation of the rating system, in the event that users are repeatedly providing inconsistent rating/review combinations.
- automatically applying a binary rating to unlabeled reviews, which might be accessible, for example, from scrubbing the web for other game reviews.
- generating a more nuanced rating number for each review than just 0 or 1; this could be more representative when aggregating the reviews.
 - For example, a game which received three reviews, two that were just slightly negative and one very positive would receive binary scores of 0, 0, and 1, while receiving nuanced scores of .4, .4, and .9. Another game could receive two very negative reviews and one slightly positive one and receive the same binary scores, (0, 0, 1) but receive different nuanced scores, .1, .1, and .6. In aggregate, the binary system would rank these games the same, but would give a higher average nuanced rating to the former game.

Data Sources

Kaggle data source: <https://www.kaggle.com/code/luixmartins/starter-eda-steam-game-review/data> This includes 17494 text reviews from 2011-2018 (primarily 2014-2018), with game title, labeled as positive or negative. Investigation is performed on: characters & words in reviews, reasonable distribution of stopwords, most common words in positive/negative reviews, top bi-grams in positive/negative reviews.

Some typical preprocessing:

- Apply lowercase
- Remove punctuation
- Remove numbers
- Remove stopwords
- Remove white spaces
- Apply lemmatization
- and stemming

Metrics

Different metrics inform our analysis in their own way. There are some NLP projects that prioritize log-loss, AUC, f1, accuracy, and others.

Part of our choice depends on the following question - should this analysis investigate or explore the level of nuance of a review? Or should the end goal be simply to predict positivity/negativity well?

- Log-loss is a highly useful metric for evaluating probability. In this case, our models will be estimating a probability of being a positive review along the way (or, arguably, the level of positive sentiment in the review), so this will be a contender. This measure will continue to give feedback throughout tuning.
- AUC (and ROC curves) measure the ability for a model to separate labels correctly into two distinct categories. That is included in our goal, so this is also a contender metric that we will look at. ROC curves can serve as a useful visualization for comparison of different models. They are also useful when there is a mostly-even distribution in labels (which is what we have). If they were not even, a precision-recall curve would be more useful.
- f1 is a blend of recall and precision, and is worth consideration.
- Accuracy will have less "gray area," and won't really be able to consider the nuance (e.g. a lukewarm review might give .6 positivity), but if our priority at the end of the line is to correctly predict positive/negative (and ignore the degree of each), then this could be considered. This measure will likely give less feedback during tuning, as once a prediction meets the right label, it is considered good enough. Accuracy and f1 have better business interpretability.

So, in the end, we will optimize our models on log-loss, but report f1 and accuracy for better business interpretability.

Feature Engineering

In the course of the analysis, a Term Frequency-Inverse Document Frequency (TF-IDF), a CountVectorizer, and a GloVe representation are created for our raw review data.

Forecasting Methodology

Classification

The initial approach is that of a binary classification (is a given review positive or negative), since the initial data is labeled that way. Another common approach is to classify as positive/neutral/negative, but that would be an investigation for the future. An additional future goal could be to move ahead with a "positivity/negativity" continuous ranking (since many of the models create a non-binary or probabilistic rating already).

Models

Model performance was ultimately ranked on accuracy, since this was the most business-interpretable metric of those listed above.

Logistic regression, Naive Bayes, support vector machines, singular value decomposition, XGBoost, and several Deep Learning models (including Dense network, RNN, LSTM and GRU) were constructed and applied to above feature sets. The best performing model (Logistic regression on TF-IDF set) had 84.3% accuracy on the test set. Industry benchmarks are in the mid 90%, so this is not terrible for an exploration project (and one with LOTS of vernacular/slang).

Further Improvements

One of the desired future outcomes would to be to develop a "positivity/negativity" continuous rating to text reviews, with which to test consistency on non-binary ratings. Also, improved performance on the deep learning models could be pursued, perhaps with additionally tuned or complex models or more data.

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

- Jupyter Notebook 100.0%