



Video Game Review Analysis

NLP Classification with Machine Learning and Neural Networks

Andrew Muller



Business Case

This goal of this project is to create a model that can predict sentiment of internet talk about video games. Specifically, it will be a neural network model trained on Steam reviews. The reviews are marked either "suggested" or "not suggested", corresponding to a results classification of "positive" or "negative". Eventually, the project will result in a website that, when supplied with a Twitter hashtag or Reddit thread, will analyze the sentiment of the related comments.



Resources

Modeling Data:

- Steam store user reviews
- store.steampowered.com/appreviews

Production Data:

- Twitter hashtags
 - twint python library
- Reddit comments
 - PRAW python library

Libraries used:

- BeautifulSoup
- Feather
- Flask
- Gensim
- matplotlib
- NLTK
- pandas
- PRAW
- Scikit-learn
- Talos
- TensorFlow
- twint
- WordCloud

Data Collection

- User reviews collected from Steam
- Available for any Steam user to write.
- Labeled "suggested" or "not suggested"
- Can be voted as "helpful" or "unhelpful"
- Training dataset: most helpful reviews from the 750 most popular games
- Total of 73,096 reviews, as some of the games have less than 100 reviews.





Data Processing

Preprocessing:

1. Remove markdown tags and punctuation
2. Tokenize
 - NLTK RegexpTokenizer
 - Match only English letters and numbers
3. Lemmatize
 - NLTK WordNetLemmatizer
4. Stopword removal
 - NLTK English stopwords

Feature Engineering:

- TF-IDF
- TF-IDF with Bigrams
- Document Embeddings
 - Gensim Doc2Vec transformer



Machine Learning Models

	Model	Processing	Test Accuracy
0	Logistic Regression	TF-IDF with Bigrams	0.914090
1	Multinomial Naive Bayes	TF-IDF with Bigrams	0.871614
2	Random Forest	TF-IDF with Bigrams	0.866963

- TF-IDF with bigrams outperformed other processing methods
- Logistic regression outperformed other model types
- Narrow hyperparameter gridsearch performed on logistic regression and random forest models
- Best test accuracy of 91%



Neural Networks

- Basic CNN with only dense and dropout layers
- Talos gridsearch to tune hyperparameters
- Best test accuracy of 91%
- Accuracy matches earlier machine learning model, but may better generalize to non-Steam reviews

```
model = Sequential()

# hidden layers
model.add(Dense(500, input_dim=8000, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(250, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(125, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(250, activation='relu'))

# output layer
model.add(Dense(1, activation='sigmoid'))
```

Flask Demo



Conclusions and Future Work

- Internet chatter about video games is generally positive.
- The models overpredict on the majority class, positive.
- To improve the usefulness of the project, I also need to incorporate some method of topic modelling.
- Gathering data from other sources could lead to a model that is better able to generalize.
- My product for this project can currently only live locally. Incorporating AWS is needed to get it on the internet

Thank You

Any Questions?



<https://github.com/MullerAC/video-game-review-analysis>