Video Game Reviews Sentiment to Popularity

Team:
Robert Coughlin

Advisor:
Amarnath Gupta

Problem Statement

Game reviews signal quality of the game title and are generated by various communities, influencers, and **traditional media journalists** [7].

Reviews may have the potential to affect or **predict** the sales of a game based upon the hypothesis that, if a review indicates a game is good, then it will attract more owners to purchase it, or may otherwise indicate the qualities of a game that sells well which may be independent of its overall quality.

For this project, the question then is **if at the minimum, can the text of a review be used to accurately predict the ability of the corresponding game to meet a certain sales threshold?**

For the project, this threshold will be if the game has **more than 500,000 owners on Steam**, which will be the positive case for the **bestseller class.**

Data Extraction

SteamSpy
3rd party data aggregator.
Ownership data approximations.
~ 10,000 games sales data.

steam_critic.csv
Search results w/ inverse trigram distance.
Used to filter out games w/ poor search match or no match.

OpenCritic.
Review aggregator w/ search function.
Snippets of review text.

steam.csv
List of all steam games and their application IDs in steam.
Name
ID (entity matching)

Steam
Game Distribution Platform.
Game ID data and titles.
100,000+ Games.

steam_sales.csv
Partially cleaned sales data with max owner range and game ID.
Sales range

reviews.csv
Review data.
Review summaries

Data Extraction and Partial Cleanup
- Data read into local PC.
- Some games dropped.
- Cached to local PC as CSV Files.

Cleanup 1
- Features selected.
- Sentiment calculated [6].
- Some reviews dropped.

PostgreSQL Database Storage
- Stores intermediate data from cleanup 1.
- Sales and Review data are joined together by game ID on query.

~ 41 MB Used on AWS.

Data Science Pipeline

The projects seeks to verify if review text has predictive power for determining if the corresponding game meets the above threshold. Then, to test various techniques for improving the performance of the model and how to best scale it with the real-world balance of data.

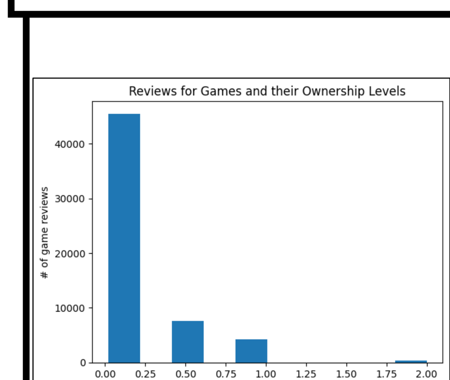
Each of the white textboxes describe a modification to the pipeline executed separately to try and improve the predictive performance of the final model.

Identify if alternative inputs improve the performance
1. TextBlob sentiment appended to vectors.
2. spaCy document and sentence vectors instead of Gensim vectors.
3. Gensim doc vectors averaged by game.

Identify if additional aggregate techniques improve the performance
1. Concatenate a game's review text together into one.
2. Combine class prediction probabilities for a game and vote for the one with the highest score.

Tune the classifiers using GridSearchCV [4] to optimize them to accuracy predict positive cases. This data will be used to modify hyperparameters for scaling and aggregation, as well as the final solution architecture.

X_grid.json
Stores best hyperparameters for learner X.



Initial extraction iteration of data indicates bestsellers will be a minority class. Composition has since changed but remains heavily imbalanced.

Scaling & Robustness
1. Test if the initial balancing technique (downsampling negative cases) scales to accommodate more and unbalanced data expected in the real world.
2. Test how well it can handle character substitutions and order swaps [2].

Classification Pipeline

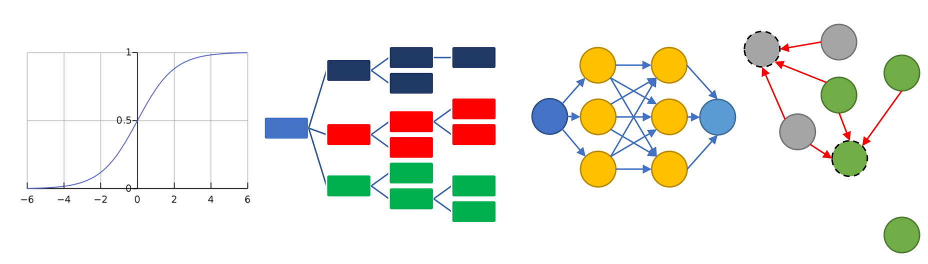
Cleanup 2
- Bestseller classes calculated.
- Review text cleaned & tagged.

Doc2Vec Transformer
- Custom wrapper for Gensim Doc2Vec
- Tagged documents are converted using Doc2Vec to vectors [1][3].
- Must be trained.

MinMaxScaler
- Scales the input vectors to [0,1].
- In most cases, from [-1,1] to [0,1].

Binary Classifier

Chose one of the following:



Logistic Regression	Random Forest	Multi-Layered Perceptron	K-Nearest Neighbors
- Simple & Fast - Linear	- Generalizable for many problems - Less likely to overfit	- Generalizable for many problems	- Simple - Non-Linear

Boolean (Is it a bestseller?)

Data Transformation

Cleanup 1 (temporary for sentiment analysis)
Cleanup 2

Original
This is a great game, with excellent mechanics, a compelling romance story, and a well optimized and bug-free execution.

Special character removal
This is a great game with excellent mechanics a compelling romance story and a well optimized and bugfree execution

Whitespace removal
This is a great game with excellent mechanics a compelling romance story and a well optimized and bugfree execution

Lowercasing
this is a great game with excellent mechanics a compelling romance story and a well optimized and bugfree execution

To word array with stopwords removed
['great', 'game', 'excellent', 'mechanics', 'compelling', 'romance', 'story', 'well', 'optimized', 'bugfree', 'execution']

Stemmatization
['great', 'game', 'excel', 'mechan', 'compel', 'romanc', 'stori', 'well', 'optim', 'bugfre', 'execut']

Doc2Vec

Tagged Document
- ['great', 'game', 'excel', 'mechan', 'compel', 'romanc', 'stori', 'well', 'optim', 'bugfre', 'execut']
- tags=[True] (training only)

Document Vector
[0, -.391, .393, .181, .381]

MinMaxScaler

Normalized Document Vector
[0.5 , 0.3045, 0.6965, 0.5905, 0.6905]

Binary Classifier

Boolean (Is it a bestseller?)
True

Final Solution Architecture

The current model's test scores from the database and the latest prediction & its certainty are outputted.

SteamSpy, Steam & OpenCritic

Database

Review & Sales Database

Server

Will the game be a bestseller (> 500,000 owners)?
Yes/No? Probability?

Model chosen manually using a config file.

Config JSON

Review Sentiment Analyzer

Hyperparameter JSON

Review Text

Hyperparameter tuning sets hyperparameters for model.

End User

Dashboard Webpage

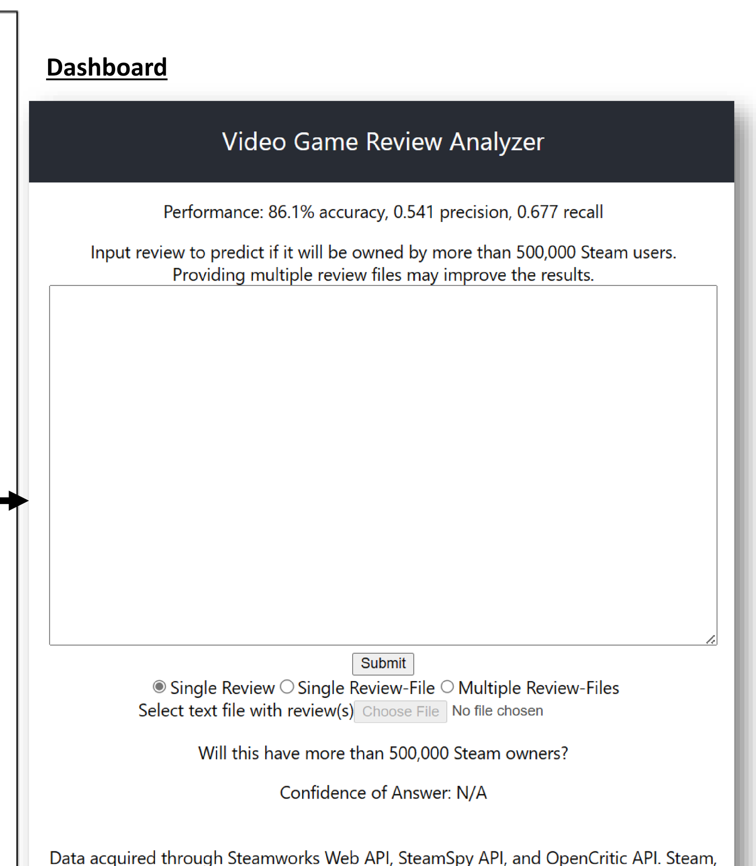
File of review(s) uploaded & submitted

New review written & Submitted

Data pipeline duplicated into a flask server [5] to receive user inputs and is fitted on server start.

Fitted model is cached to quicken future re-runs.

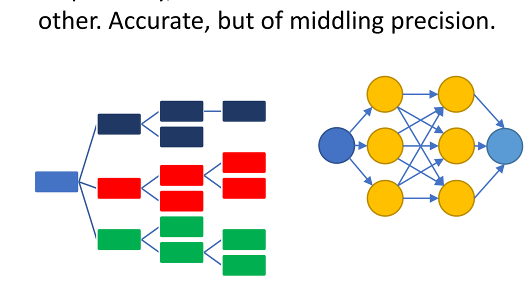
Dashboard

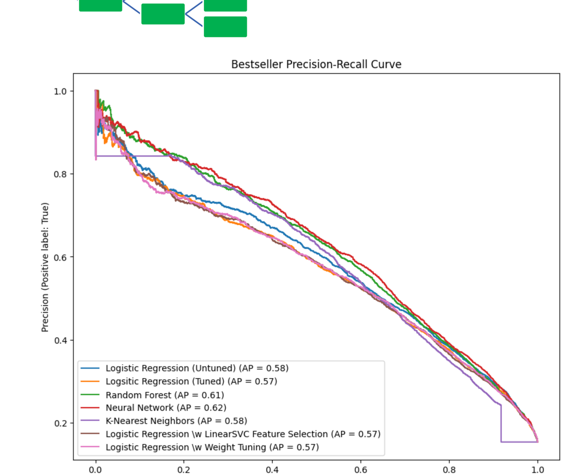


Pressing the submit button will cause the text from either the text box or the files chosen to be submitted to the server.

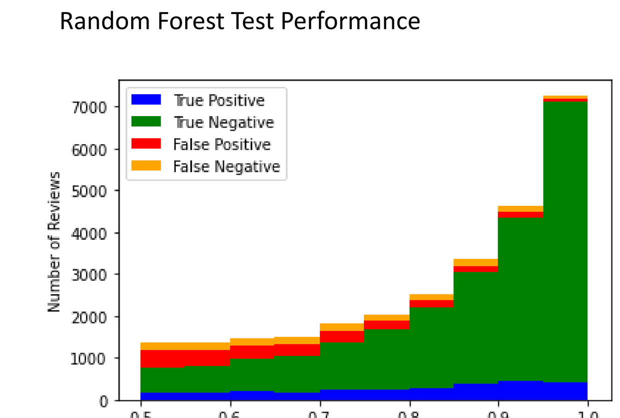
Key Findings

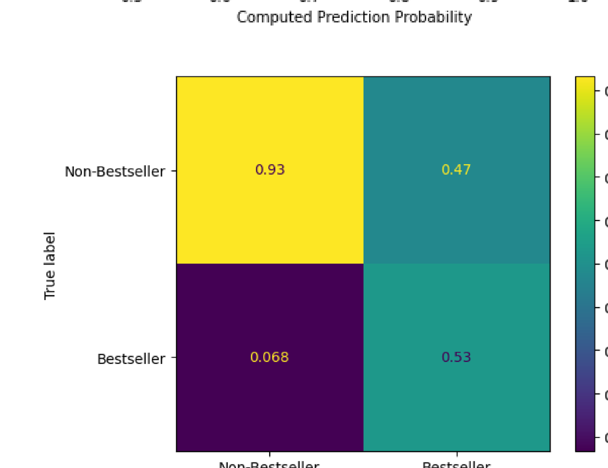
Random Forest and **Multi-Layered Perceptron** have the best F1 and accuracy respectively, with almost the best of the other. Accurate, but of middling precision.





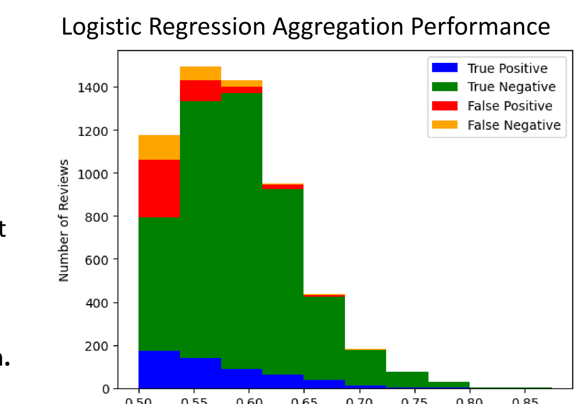
Random Forest Test Performance





Most alternative data preps and aggregating input vectors brought either **detrimental or statistically insignificant** changes in performance.

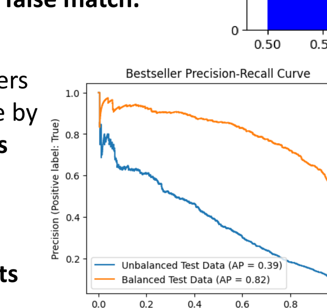
Logistic Regression Aggregation Performance



Aggregating the outputs by probability vote improves accuracy, precision, and F1 the best out of all tested aggregations.

In most cases, an **increasing certainty** of the answer **reduces the chance for a false match.**

Adding more data & scrambling characters & word order doesn't drop performance by large margins, but **downsampling leaves the models unprepared for imbalanced data**.



Other workflows modified to use **weights instead of downsampling** in reaction.

References & Acknowledgements

1. Ahn, Sang ho & Kang, Juyoung & Park, Sang-Un. "What makes the difference between popular games and unpopular games? Analysis of online game reviews from steam platform using word2vec and bass model." *ICIC Express Letters.* 11. 1729-1737. 10.24507/icicel.11.12.1729.

2. Catellier, Rémi & Vaiter, Samuel & Garreau, Damien. 2023. "On the Robustness of Text Vectorizers."

3. Li, Susan. "Multi-Class Text Classification with Doc2Vec & Logistic Regression." *Medium*, 4 Dec. 2018. towardsdatascience.com/multi-class-text-classification-with-doc2vec-logistic-regression-9da9947b43f4#:~:text=Doc2vec%20is%20an%20NLP%20tool.

4. Panjeh. "scikit learn hyperparameter optimization for MLPClassifier." *Medium.* 29 June 2020. https://panjeh.medium.com/scikit-learn-hyperparameter-optimization-for-mlpclassifier-4d670413042b

5. Reine, R. "How to build & deploy a react + flask app." *Towards Data Science.* 29 June 2022. https://towardsdatascience.com/build-deploy-a-react-flask-app-47a89a5d17d9

6. Shah, P. "Sentiment Analysis using TextBlob." *Towards Data Science.* 6 November 2020. https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524

7. Credit to Polina Haryacha for help in developing the background and problem statement, and for developing the Jupyter notebooks for Steam and SteamSpy data extraction.

All data was collected from Steam, SteamSpy, and OpenCritic through their web APIs. No personal data was collected from Steam, SteamSpy, nor OpenCritic. Steam, SteamSpy, and OpenCritic are not affiliated with this project.