Gaming the System: Machine Learning Methods to predict user video game enjoyment

Greg Bodik, Lucca Guimaraes, Matthew Quinn
(gnb23, lbg45, mtq6)

Subject Background

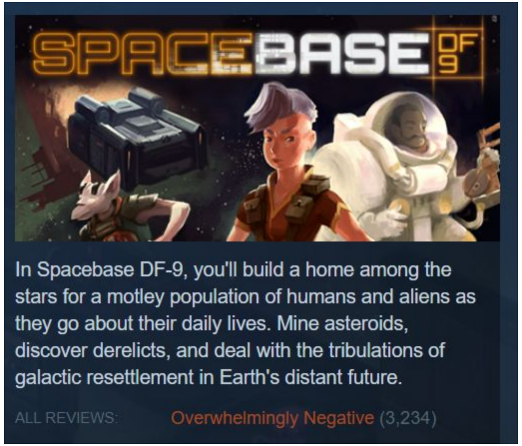
- Steam is the largest digital games platform for PCs. As of February 2022, it’s storefront contains 10,696 games [**source**: steampowered.com]

- Each game has a store page was various pieces of information about the game, such as genre or price.

- Importantly, Steam also features a system of user reviews which could provide compelling data to game developers and marketers

Steam's System of User Reviews

Overwhelmingly Positive	95-100% positive reviews
Very Positive	80-94% positive reviews
Positive	80-99% positive reviews (few in number)
Mostly Positive	70-79% positive reviews
Mixed	40-69% positive reviews
Mostly Negative	20-39% positive reviews
Negative	0-19% positive reviews (few in number)
Very Negative	10-19% positive reviews
Overwhelmingly Negative	0-9% positive reviews



Project Motivations and Goals

Data Scraping

- Obtain a useful dataset to learn abou
prefe
stear

Machine Learning

- Train several ML models of varying
accuracy and link model parameters to differences in performance

Deployment and Analysis

- Develop basic gui
odel
and
te its
- Interpret findings and form larger narrative

Goal: Find aspects of steam data that can reliably predict users' feeling about video games. Use those metrics to train several machine learning models, which in theory could serve as tools to help develop more compelling games.

The Data: Sources and Webscraping

- Data came from two sources:
  - Scraping information from store.steampowered.com (steam's official website)
  - data.world's Steam Game Dataset
- In our webscraping phase, we used the webscraping tool 'scrapy' to collect information from each game's storefront
- Further research led us to the *Steam Game Dataset* which had other features obtained from steamdb.com, a website which does not support scraping

The Data: Part II

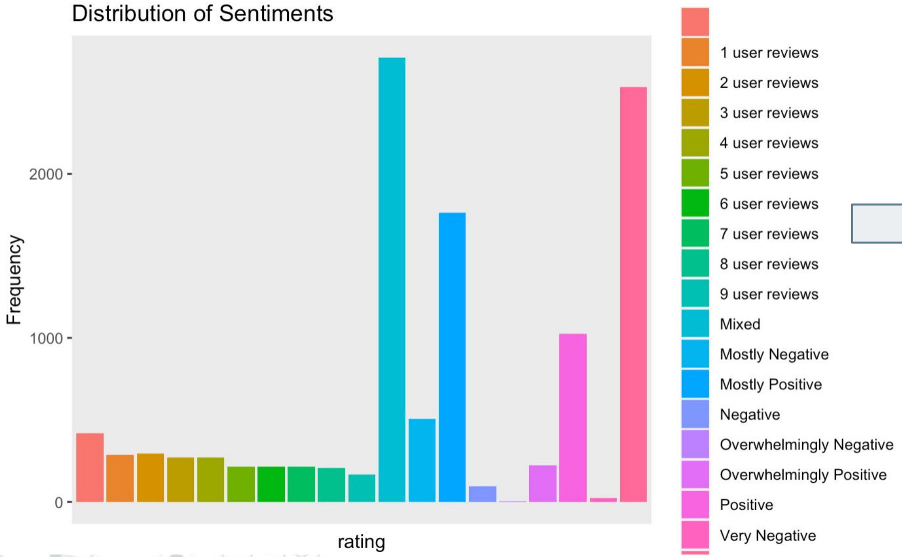
- Each data source had its own advantages/disadvantages
  - Our own webscraped data **had valuable information on user sentiment** about games (obtained from reviews they left) but was poorly organized due to the structure of tags on steam's site
  - Data from data.world was **well-organized** but didn't have sentiment information
- **Solution**: we merged the two datasets by common titles
- After merging them together and filtering we were left with **6,939 unique games across 88 features.**

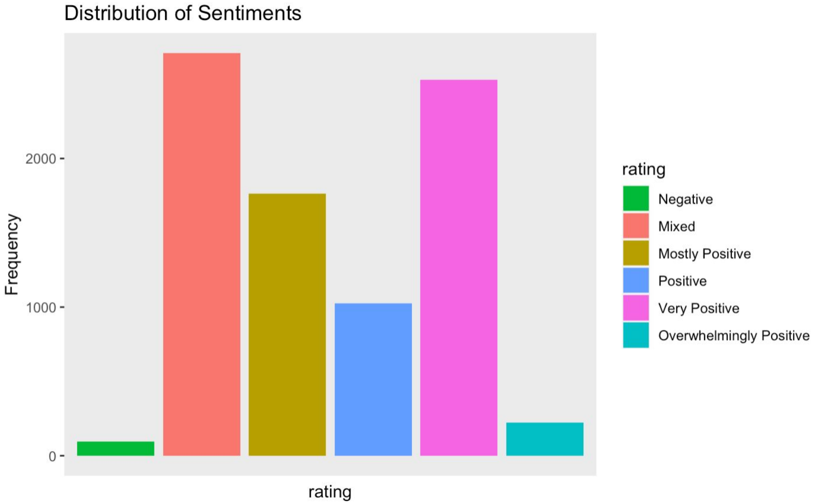
AboutText_polarity	AboutText_subjectivity	ShortDescrip_polarity	ShortDescrip_subjectivity	DetailedDescrip_polarity	DetailedDescrip_subjectivity	RelDate_converted
-0.100000	0.300000	0.468750	0.750000	-0.100000	0.300000	736026
0.245726	0.545299	0.099074	0.473148	0.245726	0.545299	736053
0.114394	0.686869	-0.037500	0.591667	0.114394	0.686869	736173

3 games in our dataset and *some* of the available features

Data Cleaning/Exploration

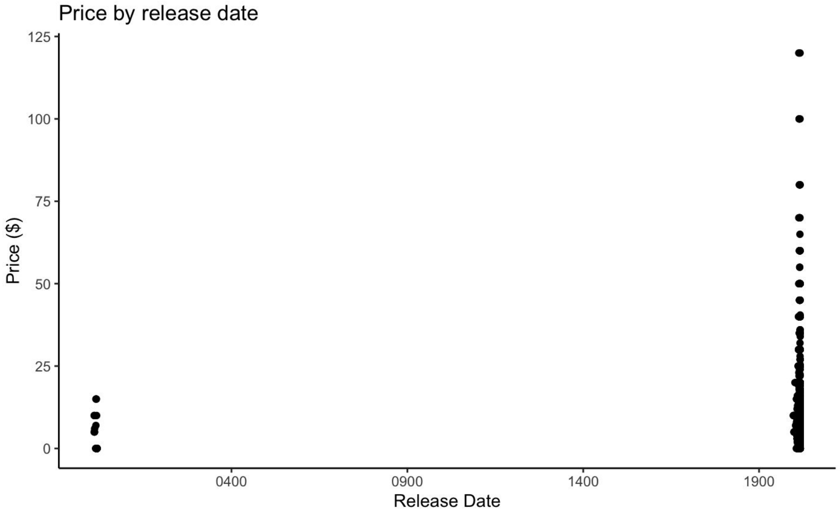
Grouping related user sentiments

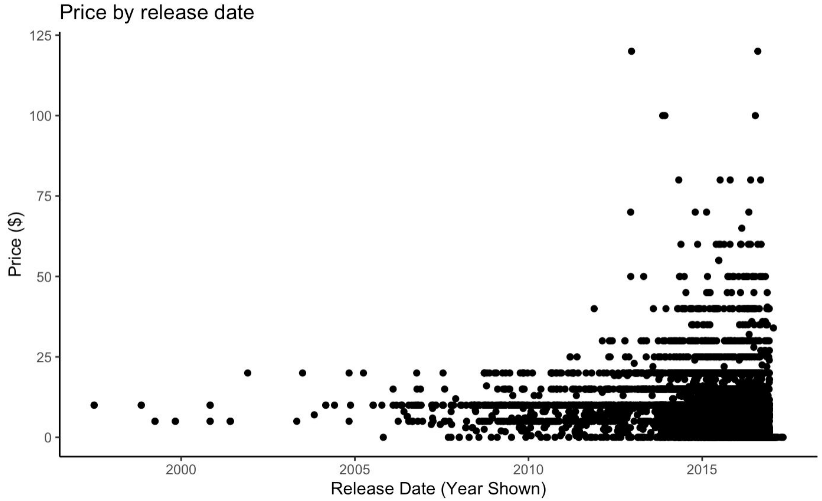




Data Cleaning/Exploration

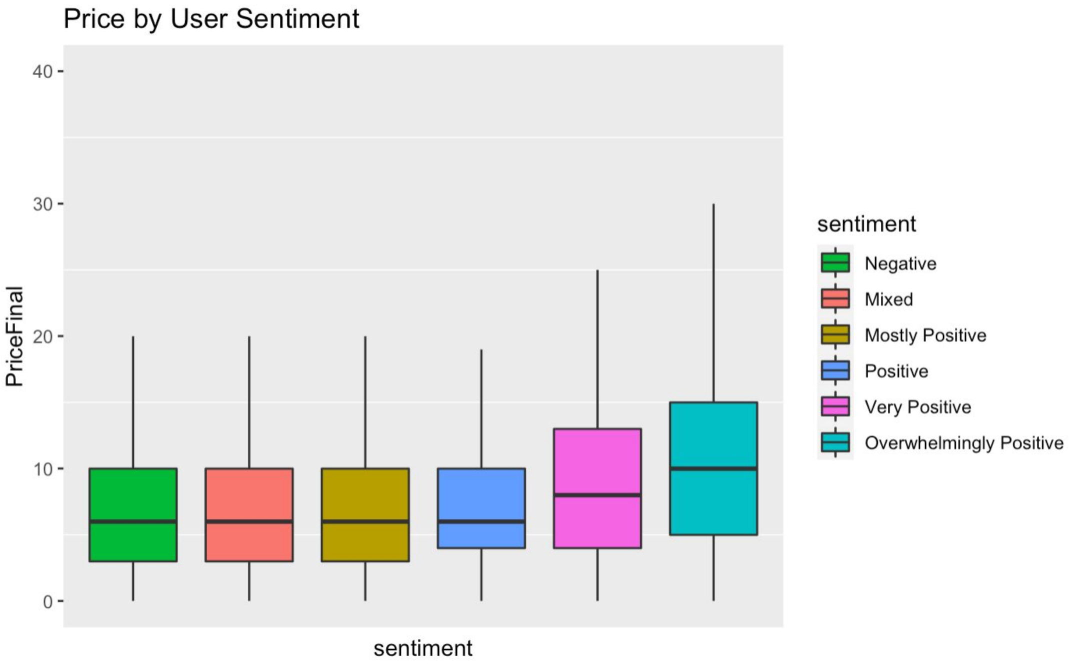
Making use of dates





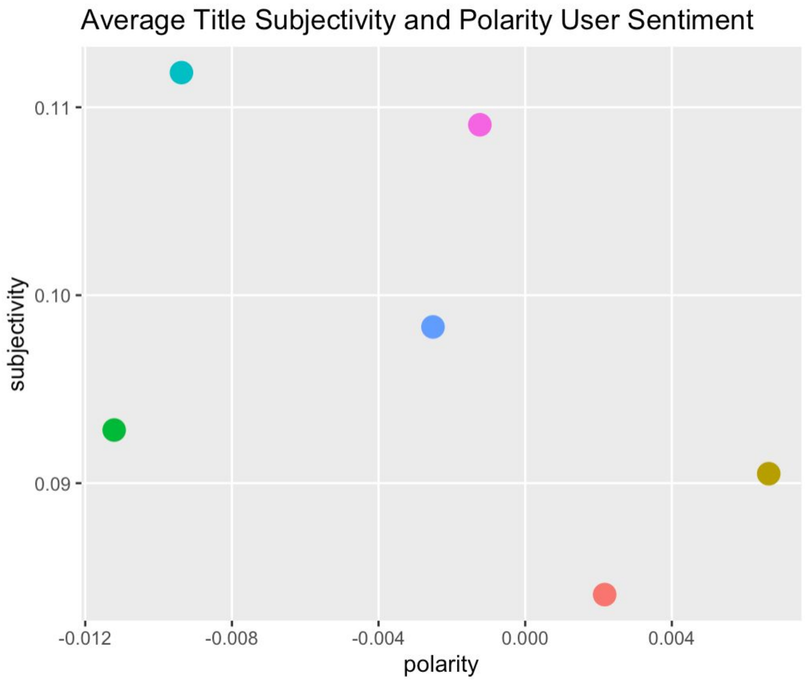
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
"1997-06-30"	"2014-06-12"	"2015-07-17"	"2014-12-19"	"2016-04-21"	"2017-05-01"

Data Cleaning/Exploration



Feature Engineering: Text Processing

- NLP-based sentiment analysis was used to determine the polarity and subjectivity of certain features.
- The Textblob package uses the NLTK toolkit to create a bag-of-words model from text, and derive averaged pooled sentiment scores from its individual words
- **Polarity**: [-1, 1] representing [negative tone, positive tone]
- **Subjectivity:** [0, 1] representing [not subjective, very subjective]



Feature Selection

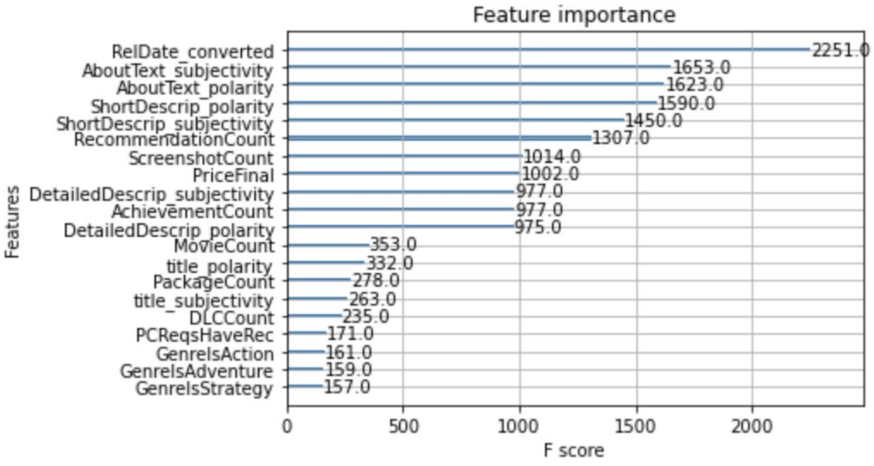
	Recursive Feature Elimination (RFE)	XGBoost Feature Importance	Random Forest Feature Importance
Top 10 features ranked	DLCCount Package Count Controller Support Platform: Linux? Platform: Mac? PCReqsHaveRec Category:Multiplayer? Genre:IsCasual? Genre:IsStrategy? Genre:IsSimulation?	Release Date About Text Subjectivity Short Description Polarity Recommendation Count Screenshot Count Price Detailed Desc. Subjectivity Achievement Count Detailed Desc. Polarity Movie Count	Recommendation Count Release Date Achievement Count Genre:IsSimulation? Genre:IsMultiplayer? Price DetailedDesrip subjectivity About text subjectivity MacReqsHaveMin? Platform:Mac?

Feature Selection *(continued)*

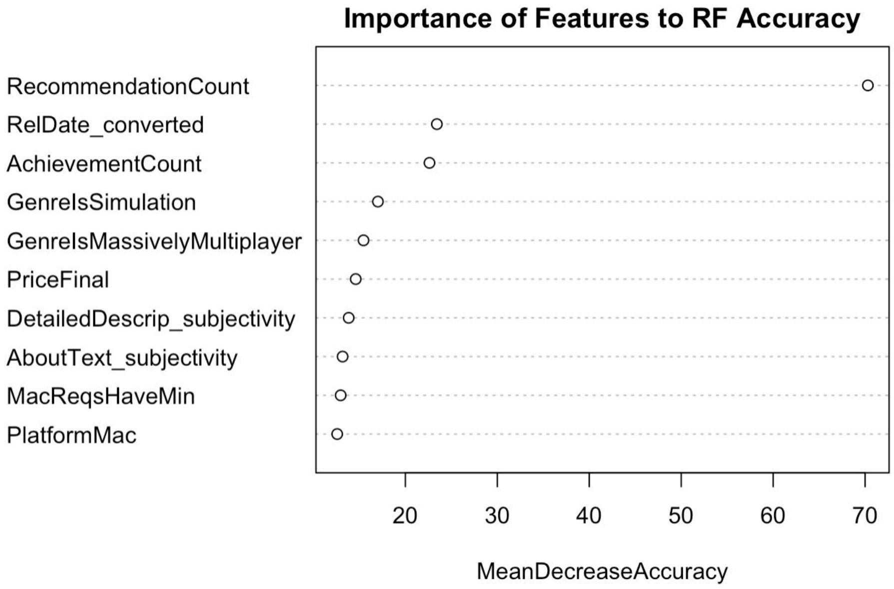
	Recursive Feature Elimination (RFE)	XGBoost Feature Importance	Random Forest Feature Importance
Top 10 features ranked	DLCCount Package Count Controller Support Platform: Linux? Platform: Mac? PCReqsHaveRec **Category:Multiplayer?** Genre:IsCasual? Genre:IsStrategy? **Genre:IsSimulation?**	**Release Date** **About Text Subjectivity** Short Description Polarity Recommendation Count **Screenshot Count** **Price** Detailed Desc. Subjectivity **Achievement Count** Detailed Desc. Polarity Movie Count	Recommendation Count **Release Date** **Achievement Count** **Genre:IsSimulation?** **Genre:IsMultiplayer?** **Price** DetailedDesrip subjectivity **About text subjectivity** MacReqsHaveMin? Platform:Mac?

Amalgam of top 7 features across the evaluation methods was chosen to train our models

Graphical summary of feature selection

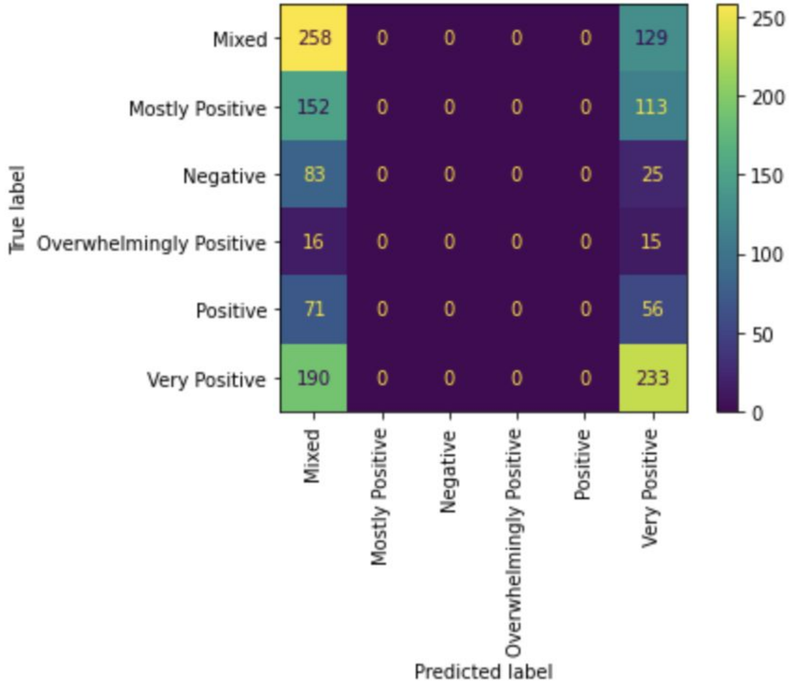


XGBoost feature importance



Random Forest Feature Importance

Multinomial Logistic Regression

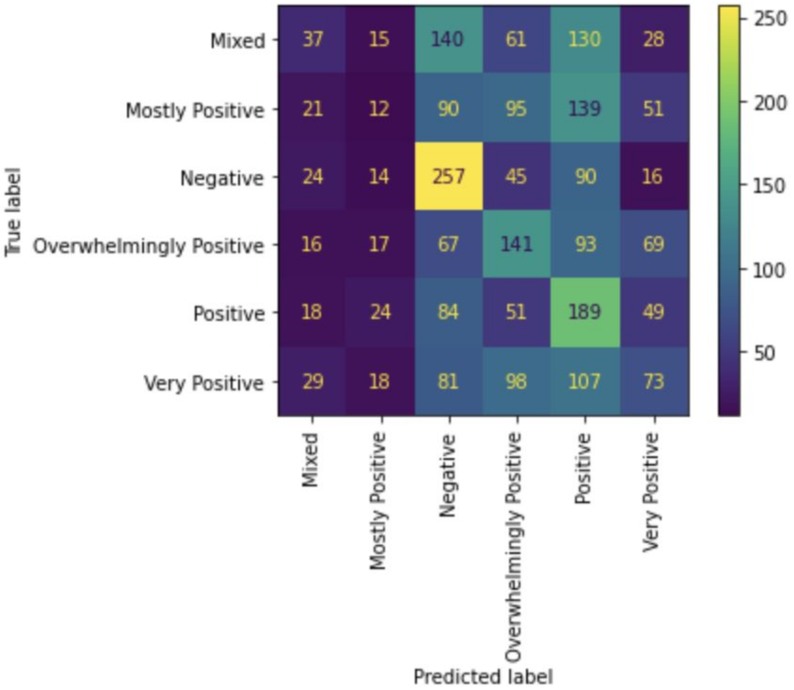


Randomize search with 10 fold CV and 2 repetitions

- C parameter
- Penalty
- Solver

Training tried to reweight classes inversely proportional to their frequency in the data

Multinomial Logistic Regression [After class rebalancing]

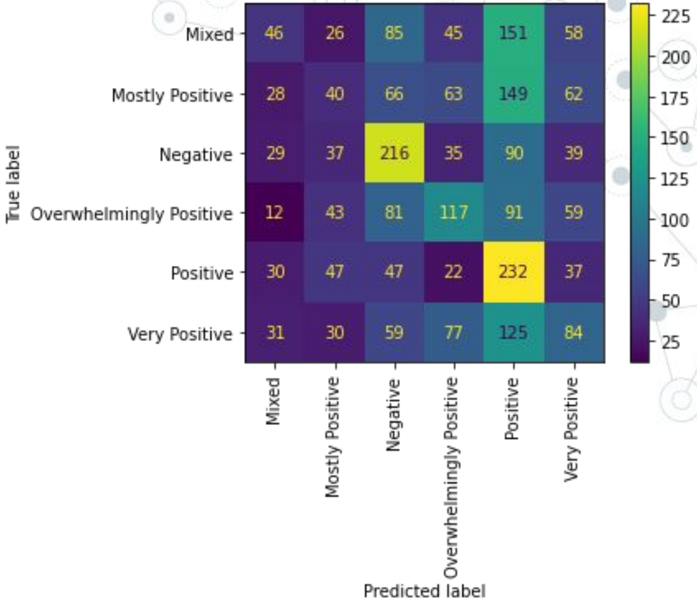


- Rebalancing classes proves somewhat useful in increasing accuracy
- Still not a large enough model for the given problem

	precision	recall
Mixed	0.26	0.09
Mostly Positive	0.12	0.03
Negative	0.36	0.58
Overwhelmingly Positive	0.29	0.35
Positive	0.25	0.46
Very Positive	0.26	0.18
accuracy		
macro avg	0.25	0.28
weighted avg	0.26	0.28

Linear discriminant analysis (LDA)

- Use likelihood function to create a linear decision boundary between classes
- QDA is more generalizable but requires too many parameters to be estimated for such a problem
- LDA is a foundational classification tool

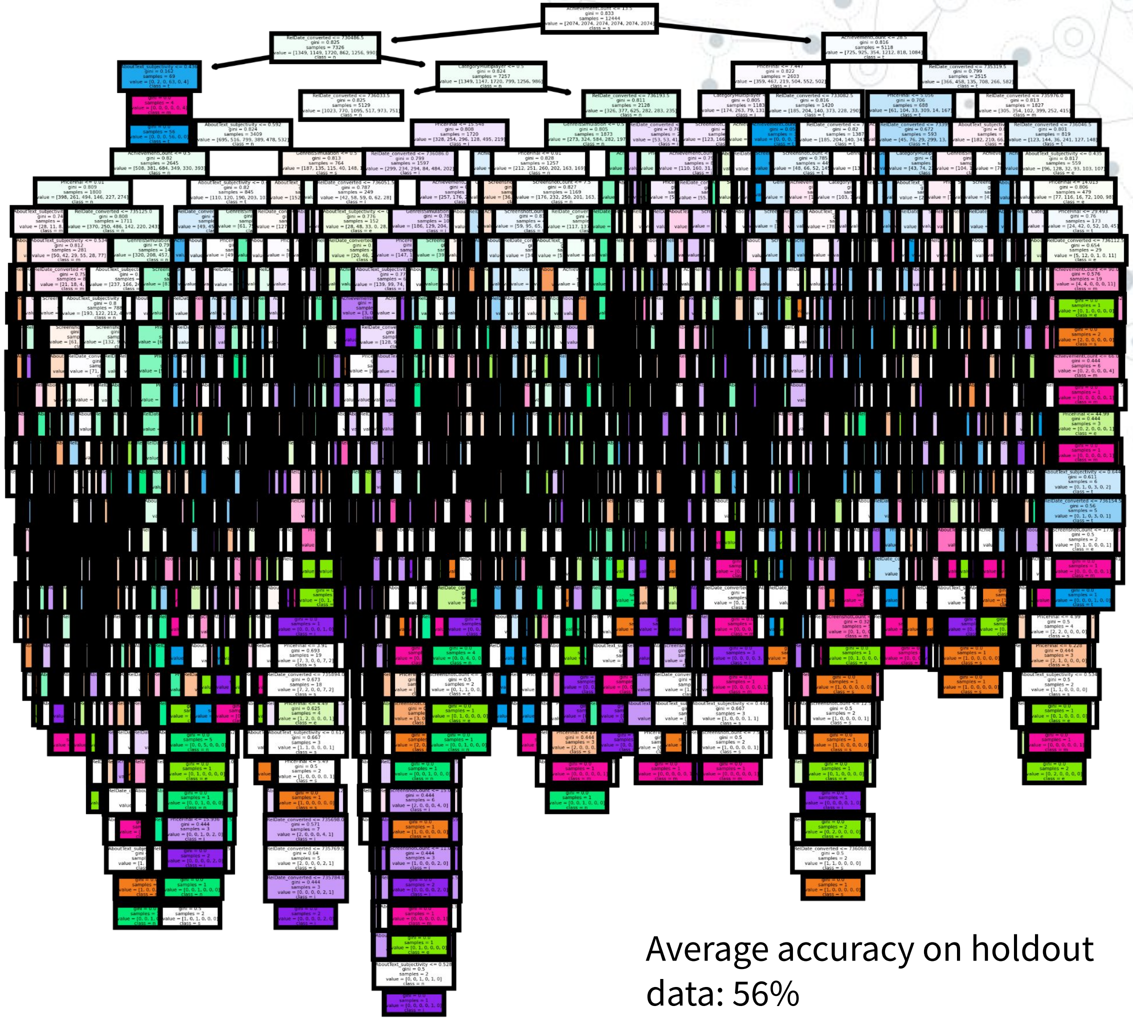


	precision	recall
Mixed	0.26	0.11
Mostly Positive	0.18	0.10
Negative	0.39	0.48
Overwhelmingly Positive	0.33	0.29
Positive	0.28	0.56
Very Positive	0.25	0.21
accuracy		
macro avg	0.28	0.29
weighted avg	0.28	0.30

Random forest

- Randomized search was used over :
  - Number of trees
  - Tree depth
  - Features in best split
  - #Samples to split internal nodes
  - #Leaves to split internal nodes
  - if bootstrapping should be used

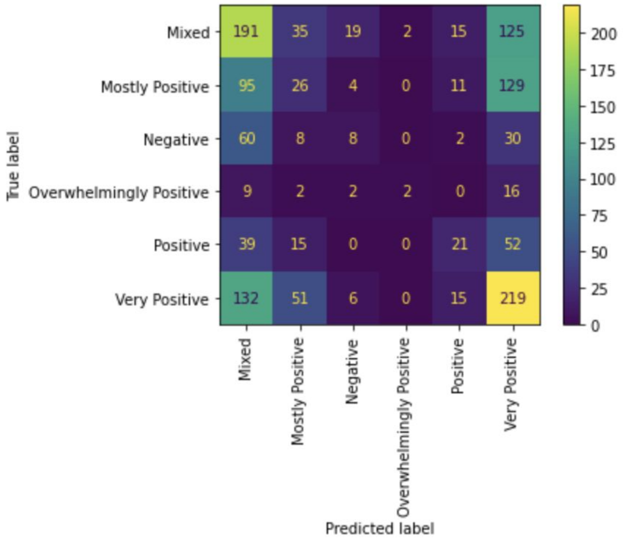
	precision	recall
Mixed	0.36	0.30
Mostly Positive	0.45	0.40
Negative	0.68	0.71
Overwhelmingly Positive	0.83	0.89
Positive	0.62	0.74
Very Positive	0.41	0.39
accuracy		
macro avg	0.56	0.57
weighted avg	0.56	0.57



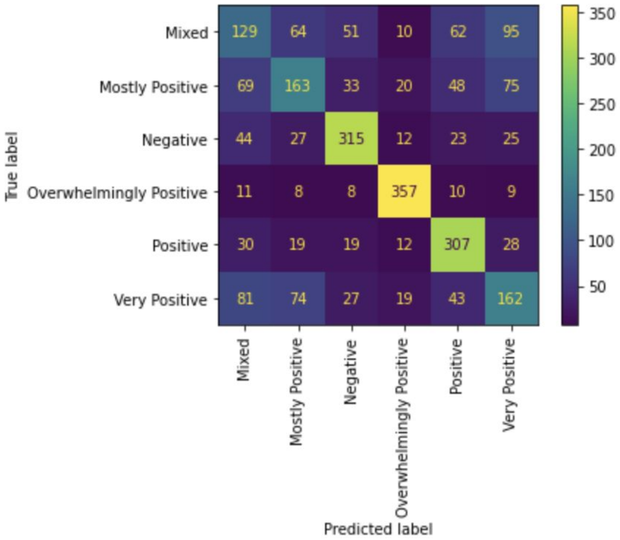
Average accuracy on holdout data: 56%

Random Forests: The benefit of upsampling

We used the Synthetic Minority Oversampling Technique (SMOTE) to oversample our underrepresented classes and artificially strike class balance in our dataset



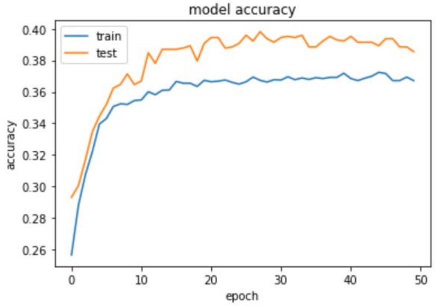
SMOTE

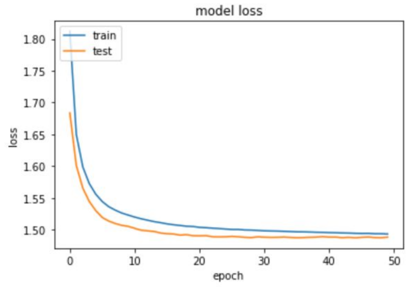


Neural Network (Design and Results)

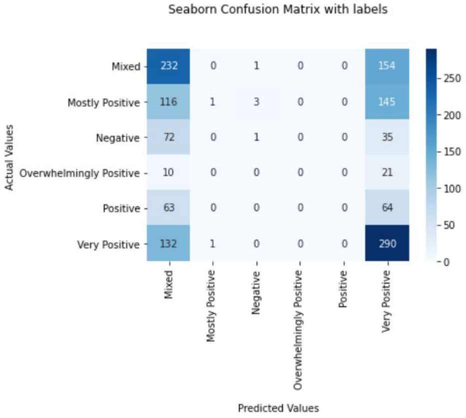
Parameter	Value
# of Epochs	50
Batch Size	20
# of Neurons	8
Dropout Rate	0.0

- We coded our own neural network architecture using the *keras* API for tensorflow
- [NN coding, training, testing took up much of our project time]
- Like other models, we optimized its hyperparamters using an exhaustive gridsearch
- The prediction accuracy was 37.18%
- Even a small neural network was unable to generalize on these data.



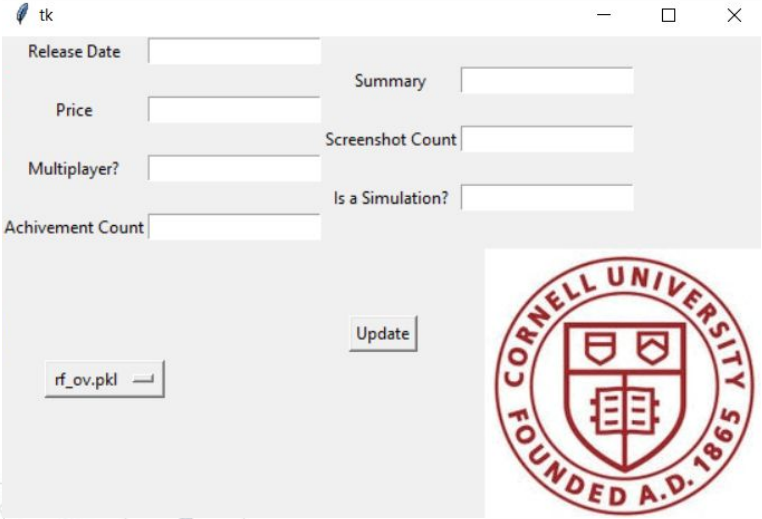


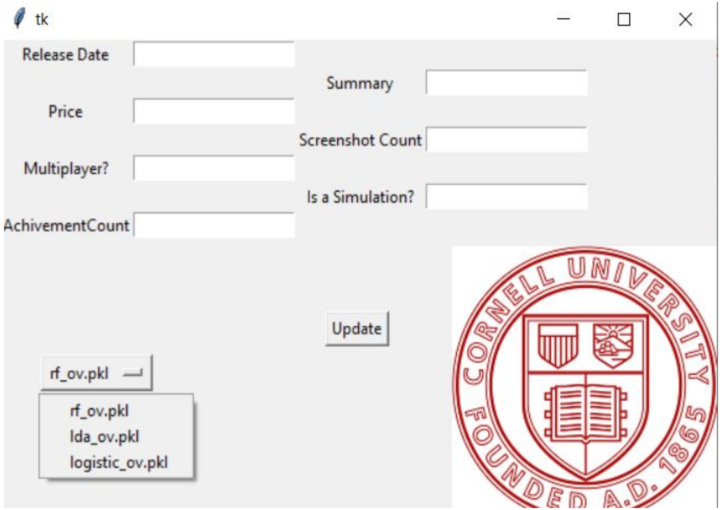

```
42/42 [==============================] - 0s 583us/step
```



GUI [and Live Demo]

-We used Tkinter to create the interactive GUI featuring a drop down menu featuring the different models.





Conclusions and Takeaways

- Although difficult, **objective metrics** can be used to predict subjective aspects of human life, in this case, user sentiment about video game
- Feature selection methods, especially when combined can effectively **reduce data dimensionality**, while preserving explainability
- **Oversampling** methods *can* effectively augment data and help to **re-balance classes**
- Larger models are not always the answer
- When the input space is small, slightly **smaller models** may be better able to **generalize relationships within the data** (random forest)
- Systematically training models using AI intuition (and sound data pipeline) results in better outcomes than relying pretrained architectures and default values