Steam Review Sentiment Analysis

Author: Daniel Wertz

Goals

- Uncover trends associated with positive and negative Steam reviews
- Determine what aspects of games are highly valued by customers
- Analyze which words are more influential for specific genres
- Develop a ML model to predict the sentiment of game reviews

About Steam

- Launched by Valve Corporation in 2003
- Largest distribution platform for PC gaming
- Steam's US library contains over 50,000 games[1]

1. Dean, B. (2023, March 28). *Steam usage and catalog stats for 2023*. Backlinko. https://backlinko.com/steam-users

Datasets

Steam Reviews: [1]

- App ID
- Name
- Review Text
- Review Sentiment
- Review Votes

~6.4 million entries

Steam Store Games: [2]

- App ID
- Genres
- Release Date
- 15 other columns

~27 thousand entries

1. https://www.kaggle.com/datasets/andrewmvd/steam-reviews
2. https://www.kaggle.com/datasets/nikdavis/steam-store-games

Datasets

Merged Dataset:

- Review Text
- Review Sentiment
- Genre
- Release Date

~5.85 million entries

	review_text	review_score	genres	release_date
0	Ruined my life.	1	Action	2000-11-01
1	This will be more of a ''my experience with th...	1	Action	2000-11-01
2	This game saved my virginity.	1	Action	2000-11-01
3	• Do you like original games? • Do you like ga...	1	Action	2000-11-01
4	Easy to learn, hard to master.	1	Action	2000-11-01
...	...	...	...	...
5858833	I really ove this game but it needs somethings...	-1	Adventure;Casual;Free to Play;Massively Multip...	2011-08-31
5858834	Used to play Puzzel Pirates 'way back when', b...	-1	Adventure;Casual;Free to Play;Massively Multip...	2011-08-31
5858835	This game was aright, though a bit annoying. W...	-1	Adventure;Casual;Free to Play;Massively Multip...	2011-08-31
5858836	I had a nice review to recommend this game, bu...	-1	Adventure;Casual;Free to Play;Massively Multip...	2011-08-31
5858837	The puzzles in this game are fun, but you have...	-1	Adventure;Casual;Free to Play;Massively Multip...	2011-08-31

[5858838 rows x 4 columns]

Preprocessing

~3 million unique “words” before preprocessing

- Removing stop words, punctuation, and special characters.
- Stemming and lemmatization

~400 thousand unique words

- Removing low frequency words (count < 5)

~88 thousand unique words

Preprocessing Challenges

- Low frequency cutoff is currently arbitrary
- Removal of game names
- Removal of nonsense words and misspellings
- Long processing time

Individual Word Analysis

$$\mathrm{PMI}(w_1, w_2) = \log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)}$$

Pointwise Mutual Information:

- Quantifies the degree of association between words by comparing their observed frequency with their expected frequency
- Suitable for analyzing imbalanced data.
- The ratio comparing negative and positive PMIs provides a measure for identifying words more strongly associated with negative or positive reviews.
- Manually picked out words from the highest ratios and separated them into categories

Individual Word Analysis: Quality

Relative Association In Positive

Word	PMI Ratio
amazing	1.23
awesome	1.23
best	1.20
masterpiece	1.17
fantastic	1.17
brilliant	1.16
great	1.15
superb	1.14
perfect	1.13
outstanding	1.13

Relative Association In Negative

Word	PMI Ratio
unplayable	1.38
worst	1.33
garbage	1.33
terrible	1.31
poorly	1.30
broken	1.27
unfinished	1.26
crash	1.26
trash	1.25
crap	1.23

Individual Word Analysis: Cost/Value

Relative Association In Negative

Word	PMI Ratio
refund	1.42
waste	1.34
microtransaction	1.27
money	1.22
pay	1.18

Individual Word Analysis: Characteristics

Relative Association In Positive

Word	PMI Ratio
soundtrack	1.14
teamwork	1.10
challenge	1.10
brutal	1.09
atmosphere	1.08
humor	1.07
cute	1.07
story	1.06
art	1.04

Machine Learning Overview

Logistic Regression:

- Fits a logistic function to the input features using an optimization algorithm
- Works well with high-dimensional data
- Simple and efficient

Decision Tree:

- Creates decision rules by splitting the data on features based on a threshold
- Works well with high-dimensional data
- Easy to interpret

Logistic Regression

```
# Logistic Regression Classifier
logistic_regression = LogisticRegression(random_state=42, class_weight={-1: 0.75, 1: 0.25})
logistic_regression.fit(train_data, train_labels)
```

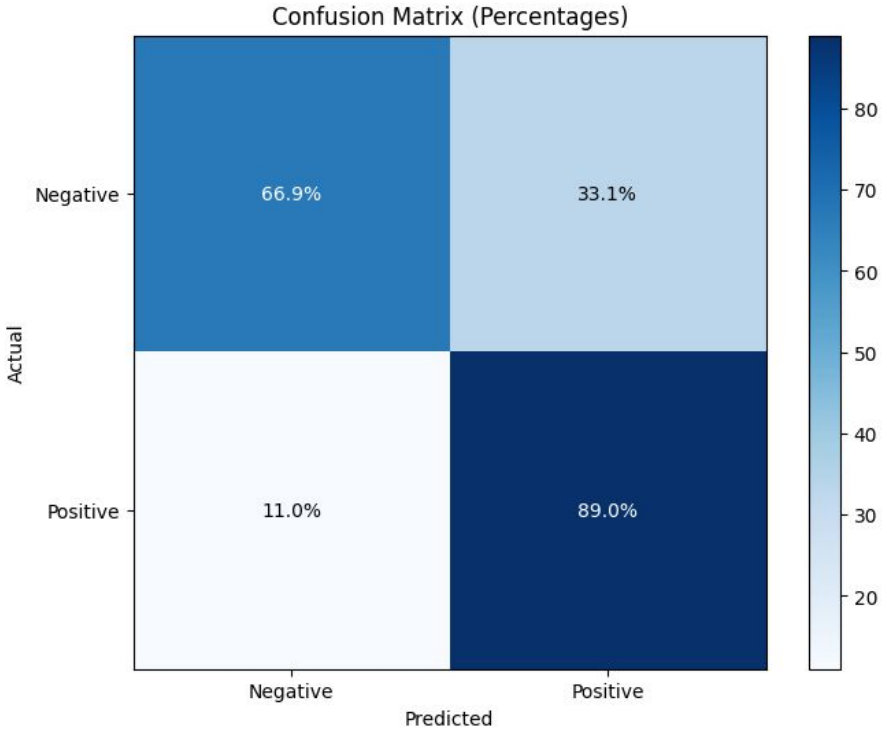
Metrics:

Accuracy: 0.85

Precision: 0.93

Recall: 0.88

F1-Score: 0.91



Decision Tree

```
# Decision Tree Classifier
decision_tree = DecisionTreeClassifier(random_state=42, max_depth=30, class_weight={-1: 0.80, 1: 0.20})
decision_tree.fit(train_data, train_labels)
```

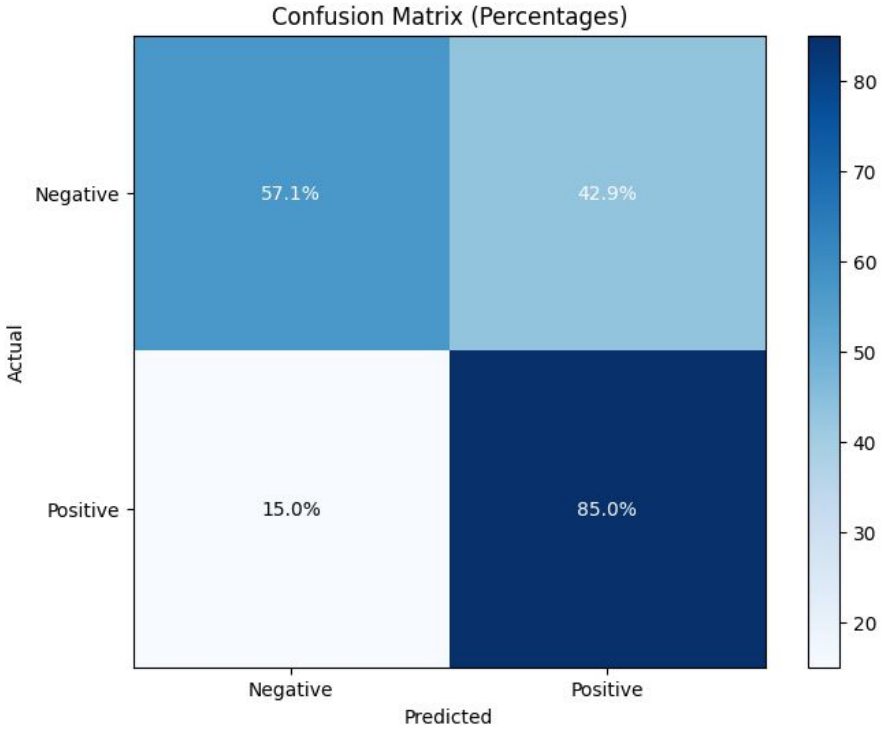
Metrics:

Accuracy: 0.81

Precision: 0.91

Recall: 0.85

F1-Score: 0.88



Future Work

- Reduce number of features
- Look for common word sequences
- Compare word frequency across genres and release date

Questions?