Steam Reviews Sentiment and Recommendations

By : Jake Hoogstra, Elliott Iturbe, Griffin Riner

Business Problem

Steam would like to be able to tell if a review for a game is positive or negative based on the words used in the review. They also want to be able to recommend games to their users based on their review.

Summary

We created two models to accomplish this task.

The Natural Language Processing (NLP) model will determine if a review is positive or negative.

The Recommender System model will give the top 10 most or least similar games depending on the review score given from the NLP model.

Features that were focused on include:

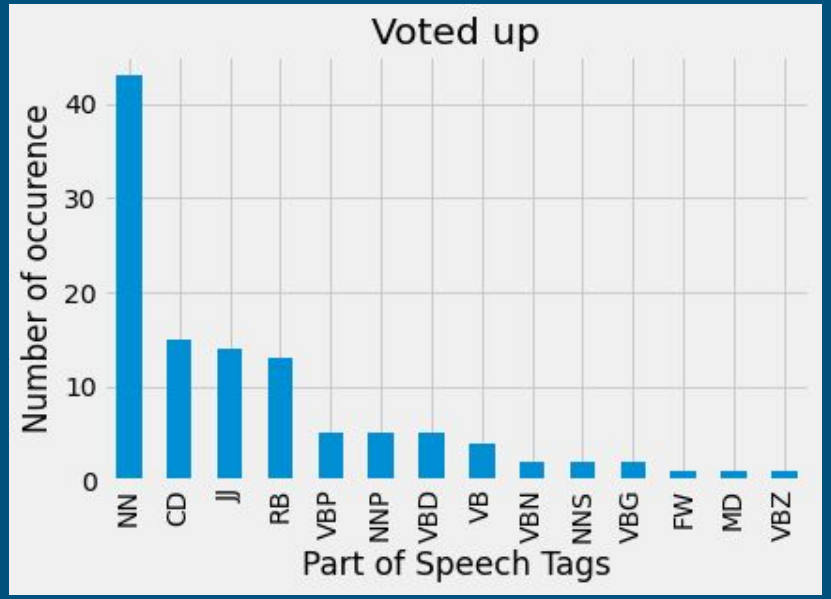
- App Tags
- Review
- Vote (upvote or downvote)
- App Title

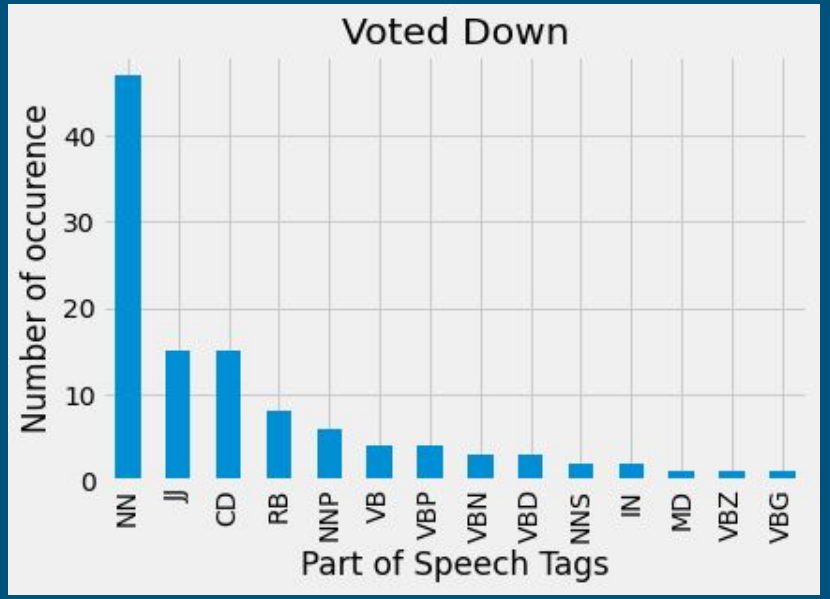
Data and Methods

- The dataset included over 23,000 reviews for 166 different first-person shooter games.
- The data also included the steam ids for the users, the tags for the games, and whether the user upvoted or downvoted the game.
- We used a MultinomialNB model for NLP and cosine similarity between the tags to calculate the recommended games.

NLP

Our NLP model takes in a review for a game and determines if the user upvoted or downvoted the game based on the sentiment of the user's review.





NLP Model/Visuals

Baseline Model

Train: 0 .997

Test: 0.808

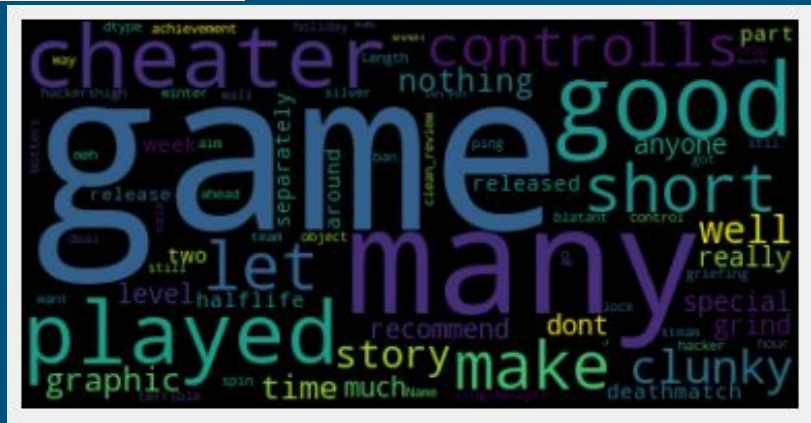
Final Model

Train: 0.942

Test: 0.845

Positive review

Negative review



Recommendation System

Our Recommendation system takes in the title of a game and the predicted review score from the NLP model and returns a list of 10 games.

- If the score was good then the 10 most similar games are recommended
- If the score was bad then the 10 most dissimilar games are recommended

The similarity scores are based on the tags given to the games.

Baseline vs Final Recommendation Model

Baseline Model

```
recommended_games('Call of Duty® 4: Modern Warfare®')
```

```
[('Portal 2', 0.8582771857135209),
 ('Portal', 0.8371327244527318),
 ('Half-Life 2', 0.7916418595395834),
 ('Left 4 Dead 2', 0.759139211311506),
 ('Counter-Strike', 0.7413840128797784),
 ('Left 4 Dead', 0.7077816939891965),
 ('Half-Life', 0.6933055022928077),
 ('Killing Floor', 0.6728345339945907),
 ('Half-Life 2: Episode Two', 0.6577451988684129),
 ('Counter-Strike: Source', 0.6502439745455961)]
```

Final Model

```
get_recommendations('Call of Duty® 4: Modern Warfare®', 1)
```

```
25     Call of Duty®: Modern Warfare® 2
96              Medal of Honor: Airborne
67                          Call of Duty®
123                     Medal of Honor™
17                      Call of Duty® 2
155           Frontlines™: Fuel of War™
146        Delta Force: Black Hawk Down
65              Call of Duty®: Black Ops
70       Call of Juarez: Bound in Blood
54       Call of Duty: United Offensive
```

Bad vs Good Review

Bad review

```
get_recommendations('Call of Duty® 4: Modern Warfare®', 0)
```

```
6                          Portal 2
4              The Ship: Murder Party
128                      Postal III
91                         The Ball
135                  Darkest of Days
164       Hamilton's Great Adventure
88                     Rogue Warrior
72                             Xotic
129     Space Trader: Merchant Marine
112     Carrier Command: Gaea Mission
```

Good Review

```
get_recommendations('Call of Duty® 4: Modern Warfare®', 1)
```

```
25      Call of Duty®: Modern Warfare® 2
96               Medal of Honor: Airborne
67                         Call of Duty®
123                     Medal of Honor™
17                      Call of Duty® 2
155          Frontlines™: Fuel of War™
146        Delta Force: Black Hawk Down
65              Call of Duty®: Black Ops
70        Call of Juarez: Bound in Blood
54        Call of Duty: United Offensive
```

Conclusions

- Using our models, Steam would be able to create recommendations based on their user’s reviews.
- This would help them tailor their recommendations to their user’s based on games they like and dislike.

Next Steps

If we were provided a system with more computational power (AWS) we would be able to use these models for all games on the Steam store and not just FPS games.

Also given more data with the user ids and more reviews per user on the same item we would like to build a recommendation system based on collaborative filtering.

Thank You

Email:

gnr400800@gmail.com

jnhoogstra@crimson.ua.edu

eaiturbe@bsc.edu

Github:

@GriffinRiner

@jnhoogstra

@eaiturbe

Index

The link here is for the Parts of Speech