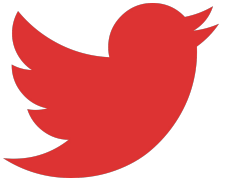

# How to Make Money and Games

Benjamin Leung

# Objective

- Find future focus for RDR2 game updates through tweets

# Tools







pandas

$$y_i t = \beta' x_{it} + \mu_i + \epsilon_{it}$$



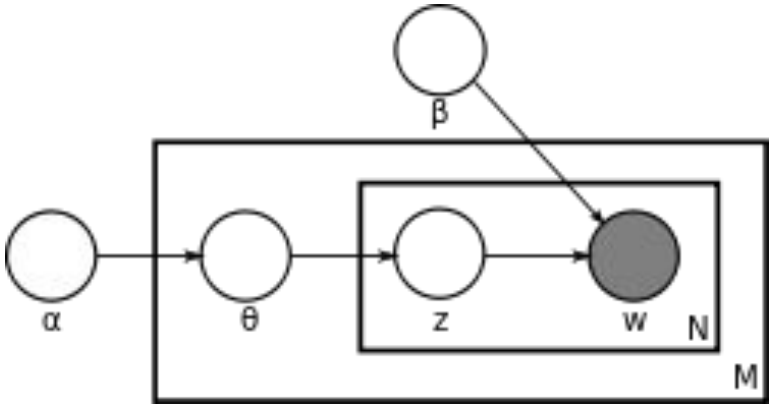



# Techniques Used

- Preprocessing
- Latent Dirichlet Allocation(LDA)
- Word2Vec
  - K-Means Clustering
  - DBSCAN

# LDA Procedure

5 Initial Topics → 5 Subtopics



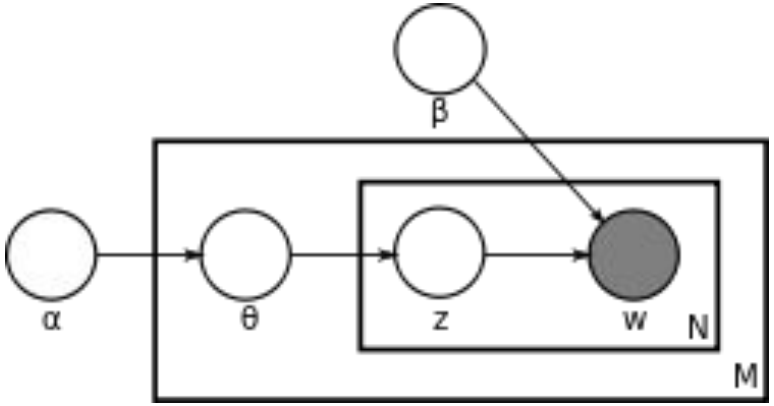


hunter
@hntrclins
Follow

damn man, i am so in love with red dead redemption 2. what a good game.

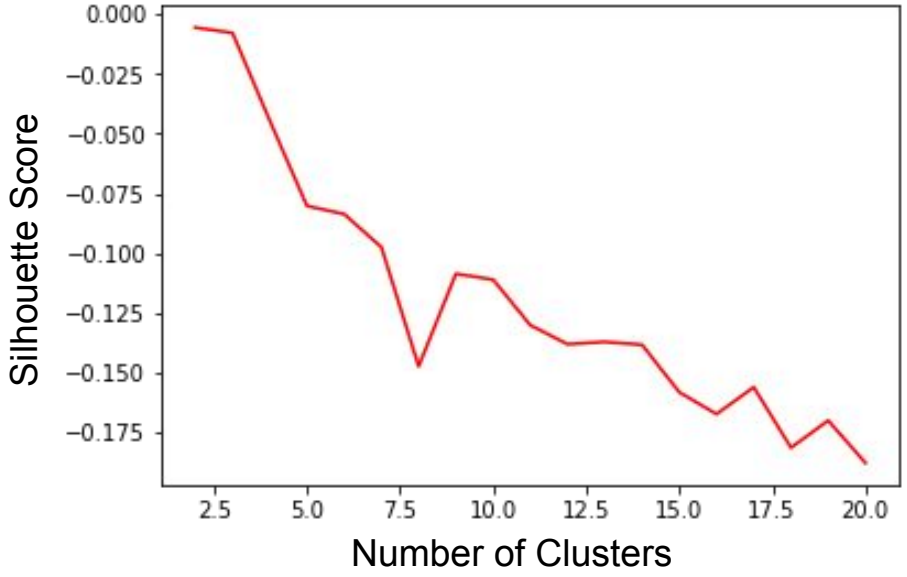
# LDA Procedure

5 Initial Topics → 5 Subtopics



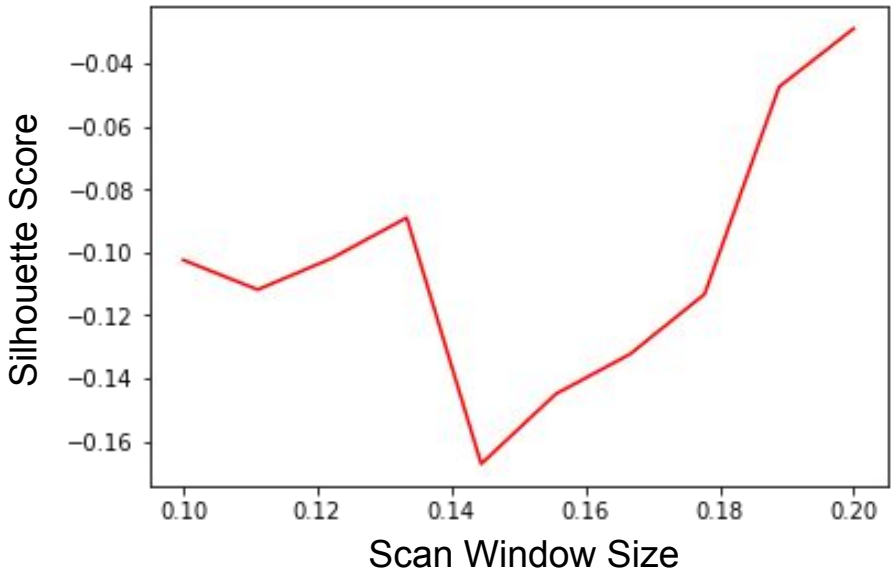

# Word2Vec -› K-Means Procedure

- SpaCy word embeddings to vectorize tweets
- K-means: Optimized with silhouette score, 8 topics



# Word2Vec -› DBSCAN Procedure

- DBSCAN: PCA and optimized with silhouette score, 7 topics, 83% noise



# Interesting Topics

| Topic | Percentage of Tweets(%) |
|---|---|
| Game Comparison | 10 |
| Streaming/Youtube | 10 |
| Buying Game | 9 |
| Easter Eggs | 9 |
| Story | 5 |
| Horse | 0.2 |

# Topic List with Overall Sentiment

| Topic | Sentiment | Standard Deviation |
|---|---|---|
| Game Comparison | 0.14 | 0.46 |
| Streaming/Youtube | 0.17 | 0.45 |
| Buying Game | 0.17 | 0.45 |
| Easter Eggs | 0.10 | 0.45 |
| Story | 0.06 | 0.41 |
| Horse | -0.12 | 0.39 |

# Conclusion

- Analysis can be applied to DLC/game updates
- Tweets reflect the hype and high quality of the game

# Next Steps

- Analyze how topic distribution/number of tweets per topic affects sales
- Assign weights to give importance to important influencers
- Use this methodology on other products



Thank you!

# Tweets Bad For Analysis

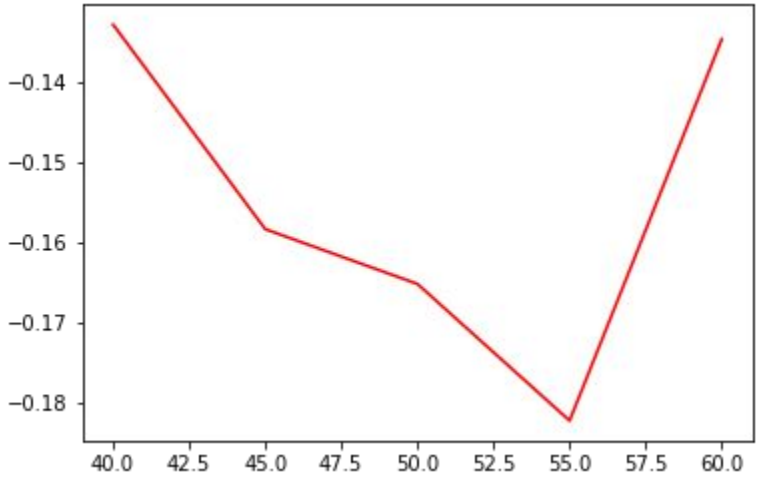
# Preprocessing

- No punctuation or emojis
- N-grams
- Stop words, including ‘red dead redemption’ and ‘game’
- Lemmatization(POS tagging/filter)

# DBScan Optimization Graphs

```
# do pca before clustering to for stronger signal and prevent overfitting
from sklearn.decomposition import PCA
pca = PCA(n_components=30)
pca.fit(vectorized_tweets_sum)
vectorized_tweets_sum_pca = pca.fit_transform(vectorized_tweets_sum)
np.sum(pca.explained_variance_ratio_)
```

```
0.6019035054124589
```



# Sentiment Histogram