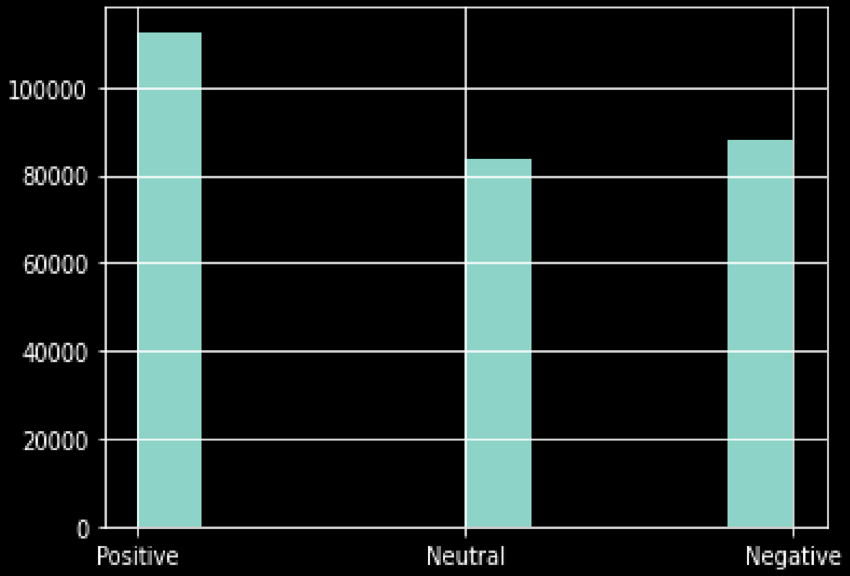
Video Game Sentiment Analysis

Intro

- Sentiment analysis can play a huge part in connecting a company to their audience
- By extracting sentiment from sources such as texts, posts, comments, etc huge amounts of valuable feedback can be attained without relying traditional numerically scored reviews

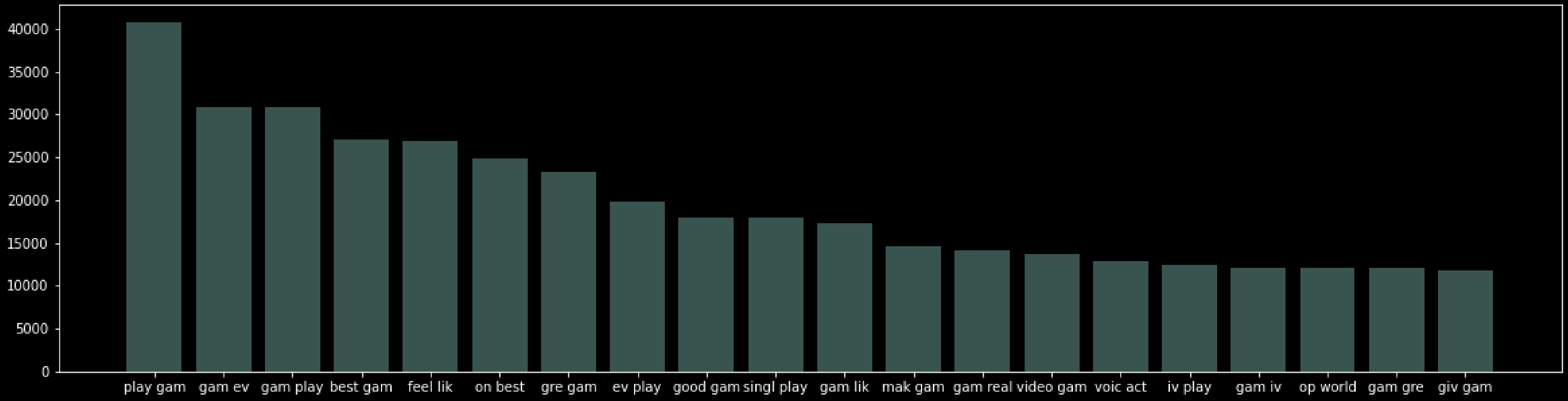
The Dataset

- The data set was a (283983, 6) data set taken from Kaggle and the columns of interest were:
  - ‘Userscore’ which contains numeric review scores
  - ‘Comment’ which contains the text for each review
- The response variable, ‘Userscore’, was converted to the classes [‘Negative’, ‘Neutral’, ‘Positive’]



The Dataset

- Exploring the frequency of the top 20 bigrams suggest that the topic of these comments commonly involve:
  - Single player and open world game types
  - Gameplay and voice acting in games



Data Cleaning

- The text was processed by:
  - removing punctuation and stopwords
  - stemming words to their root words
  - vectorizing each word into frequency columns

- The resulting dataframe contained 400000+ columns

Data Cleaning

- There were 3 methods used for dimension reduction:
  - Filtering the columns by frequency: results in a dataframe with 283983 rows and 1460 columns
  - Singular value decomposition: results in a dataframe with 283983 rows and 6 columns
  - Clustering: results in a dataframe with 283983 rows and 42 columns

Models Applied

- 4 types of predictive models were tested:
  - KNN Classification
  - Gaussian Naive Bayes’
  - Support Vector Classification
  - Neural Network
- Each model was run with up to 3 dataframes; one for each dimension reduction method
- All models were tested with a 80/20 test-train split

KNN Classifier Metrics

KNN with Frequency Filtered Dataframe				
	precision	**recall**	**f1-score**	**support**
Negative	0.614665	0.022265	0.042973	17696
Neutral	0.441989	0.019141	0.036693	16718
Positive	0.397803	0.985298	0.566776	22378
accuracy	**0.400813**	**0.400813**	**0.400813**	**0.400813**
macro_avg	0.484819	0.342235	0.215481	56792
weighted_avg	0.478383	0.400813	0.247521	56792

KNN with SVD Dataframe				
	precision	**recall**	**f1-score**	**support**
Negative	0.604553	0.681284	0.640629	17696
Neutral	0.468793	0.329764	0.387176	16718
Positive	0.616142	0.690812	0.651344	22378
accuracy	**0.581561**	**0.581561**	**0.581561**	**0.581561**
macro_avg	0.563163	0.567287	0.559716	56792
weighted_avg	0.569155	0.581561	0.570242	56792

KNN with Clustered Dataframe				
	precision	**recall**	**f1-score**	**support**
Negative	0.498493	0.738189	0.595112	17696
Neutral	0.445102	0.197871	0.273954	16718
Positive	0.593392	0.613996	0.603518	22378
accuracy	**0.530198**	**0.530198**	**0.530198**	**0.530198**
macro_avg	0.512329	0.516685	0.490862	56792
weighted_avg	0.52017	0.530198	0.503884	56792

Gaussian Naive Bayes' Metrics

GNB with SVD Dataframe				
	precision	**recall**	**f1-score**	**support**
Negative	0.462539	0.862059	0.602048	17696
Neutral	0.418567	0.247039	0.310702	16718
Positive	0.698795	0.435428	0.536534	22378
accuracy	**0.512907**	**0.512907**	**0.512907**	**0.512907**
macro_avg	0.526634	0.514842	0.483095	56792
weighted_avg	0.542688	0.512907	0.490469	56792

GNB with Clustered Dataframe				
	precision	**recall**	**f1-score**	**support**
Negative	0.444407	0.82691	0.578117	17696
Neutral	0.387075	0.243989	0.29931	16718
Positive	0.667817	0.397712	0.49853	22378
accuracy	**0.486195**	**0.486195**	**0.486195**	**0.486195**
macro_avg	0.499767	0.489537	0.458652	56792
weighted_avg	0.515562	0.486195	0.464684	56792

Support Vector Classifier Metrics

SVC with SVD Dataframe				
	precision	**recall**	**f1-score**	**support**
Negative	0.562066	0.731295	0.635609	17696
Neutral	0.483527	0.186984	0.26968	16718
Positive	0.594074	0.724819	0.652966	22378
accuracy	**0.568513**	**0.568513**	**0.568513**	**0.568513**
macro_avg	0.546555	0.547699	0.519418	56792
weighted_avg	0.551558	0.568513	0.534729	56792

SVC with Clustered Dataframe				
	precision	**recall**	**f1-score**	**support**
Negative	0.519477	0.746044	0.61248	17696
Neutral	0.479084	0.180165	0.261856	16718
Positive	0.60165	0.674591	0.636036	22378
accuracy	**0.55131**	**0.55131**	**0.55131**	**0.55131**
macro_avg	0.533404	0.5336	0.503457	56792
weighted_avg	0.539965	0.55131	0.518548	56792

Neural Network Metrics

NN with Filtered Frequency DF		
	Loss	**Accuracy**
Epoch 1	0.8	0.64
Epoch 2	0.68	0.7
Epoch 3	0.6	0.74
Epoch 4	0.52	0.78
Validation	**0.74**	**0.68**

NN with SVD DF		
	Loss	**Accuracy**
Epoch 1	1.04	0.48
Epoch 2	0.91	0.56
Epoch 3	0.91	0.57
Epoch 4	0.91	0.57
Validation	**0.9**	**0.58**

NN with Clustered DF		
	Loss	**Accuracy**
Epoch 1	0.99	0.52
Epoch 2	0.93	0.55
Epoch 3	0.56	0.56
Epoch 4	0.92	0.56
Validation	**0.92**	**0.56**

Conclusion

- The accuracy of most of the models tested seemed relatively similar ranging from around .48 to .57
- The best model was a neural network trained on a frequency filtered DF scoring an accuracy of .68
- Overall, the best model tested during this project performs significantly better than randomly guessing but is still far from reliable