Predicting the Virality of Youtube Videos

Data Science Journal / Model Results

1. Gather data via Google API. Limitation is 50 per call so 20 searches yielded 1000 rows.
2. Analyze / remove noisy features, clean the data, address NaN and outliers, create a ranking system (target column) EDA for structured data, hypothesis testing to determine correlated features
   a. Original data has 43 features. Analyzed all of the features and removed irrelevant features.
3. Modeling on structured data to include. This is expected to be a less useful model but just for exercise purposes. There should be obvious correlations between target and other numeric features:

Objective:

- Learn basics of dealing with the input and output of fitting machine learning models w/ numeric data

Model	**Accuracy**
Linear Regression	0.07
Logit	0.44
Logit - C hyperparameter tuning	0.446
KNN	0.7272
SVM	0.4242
Decision Tree Classifier	0.5539
Random Forest - default	0.6869
Random Forest - random search	0.707
Random Forest - grid search	0.6969
Gradient Boosting - default	0.6464
Gradient Boosting - random search	0.6969
Gradient Boosting - grid search	0.7272

Summary: KNN and Gradient Boosting Classifier have the best results with the numeric data.

4. Summary of EDA on unstructured data:

Count Based Features of Text Data	**Brief Description**	**Notes:**
title_char_count	# of characters in title	black belts have fewer chars
title_word_count	# of words in title	black belt videos have fewer words
title_word_density	character count / word count + 1	Disregard
title_punctuation_count	# of punctuations in title	black belt videos have punctuations
title_title_word_count	# of words that have first letter capitalized	Disregard
title_upper_case_word_count	# of words that are completely capitilized	Upside down parabola
title_stopwords_count	# of stop words in title	black belt video titles have lowest stop word avg
desc_char_count	# of characters in description	black belt videos have highest description char count avg
desc_word_count	# of words in description	black belt videos have highest description word count avg and no outliers
desc_word_density	character count / word count + 1	black belt videos have smallest range
desc_punctuation_count	# of punctuations in title	black belt videos have highest punctuation count
desc_title_word_count	# of words that have first letter capitalized	black belt videos have highest average words for first letter capitalized in desc
desc_upper_case_word_count	# of words that are completely capitilized	black belt videos have fewer completely upper case description words
desc_stopwords_count	# of stop words in description	black belt videos have MORE stop words on average inside description
tags_char_count	# of characters in tags	black belt videos have higher average for character tags count
tags_word_count	# of words in tags	black belt videos have higher average for # of words
tags_word_density	character count / word count + 1	black belt videos have slightly higher word density average with a smaller range
tags_punctuation_count	# of punctuations in tags	black belt videos have slightly higher average for punctuation count in tags
tags_title_word_count	# of words that have first letter capitalized	Disregard
tags_upper_case_word_count	# of words that are completely capitilized	black belt videos have fewer completely upper case tags
tags_stopwords_count	# of stop words in tags	Upside down parabola

Summary: We've studied the count data to see if viral videos have different word count data vs. others and certainly there are a lot of interesting features we will keep in the final model.

5. Modeling on unstructured data:

Objectives:

- Assess and deal with NLP features and applying machine learning models
- Try a number of different NLP features and apply different machine learning models

NLP Features	Model	Accuracy
Count and Density Based	Log Reg	0.3333
BoW, TFIDF	Log Reg	0.404
BoW, CountVectorizer	Log Reg	0.404
BoW, TFIDF	MNB	0.3232
BoW, TFIDF	Stochastic Gradient Descent (alpha range 1-2)	0.3232
BoW, TFIDF	KNN (nn range 1-20)	0.3232
BoW, TFIDF	XGBoost	0.3333
BoW TFIDF	Random Forest	0.3131
BoW TFIDF	Random Forest - random search	0.4637
BoW, TFIDF, stop words	Log Reg	0.3939
1-2 gram TFIDF, stop words	Log Reg	0.404
1-2, gram TFIDF, stop words	MNB	0.3535
1-2 gram, TFIDF, stemmed, stop words	XGboost	0.3737
1-2 gram, TFIDF, stemmed, stop words	Log Reg	0.3636
1-2 gram, TFIDF, stemmed, stop words	Random Forest - random search	0.404
1-2 gram, TFIDF, stop words	XGBoost	0.3737
1-2 gram, TFIDF, stemmed, stop words	Log Reg	0.404
1-2 gram, TFIDF, stemmed, stop words	XGBoost	0.3838
Train W2V Scratch, MeanEmbeddingVectorizer, unstemmed	Log Reg	0.303
Train W2V Scratch, TFIDFEmbeddingVectorizer, unstemmed	Log Reg	0.303
Train w2v, MeanEmbeddingVectorizer, unstemmed	XGBoost	0.3838
Train w2v, MeanEmbeddingVectorizer, unstemmed	XGBoost	0.4141

Summary:

- Best results with Random Forest, w/ random search parameters.
- Good results with Log Reg
- Stem did not make a significant difference however 1-2 gram with stop words seemed to provide best results so we'll carry these features on to the final model.
- For experimentation tried W2V. W2V did not work well – not enough data

6. Modeling on combined structured and unstructured data:

Objectives:

- Use advanced pipeline features such as Feature Union and combining with hyperparameter tuning
- Try to achieve 70 to 80% accuracy with combined numeric and text data.

NLP Features	**Model**	**Accuracy**
Count features, BoW, CountVectorizer	OneVsrest(Log Reg)	0.4718
Count features, BoW, CountVectorizer	Random Forest	0.5484
Count features, BoW, CountVectorizer	Log Reg	
Count features, BoW TFIDF	Log Reg	
Count features, BoW TFIDF	Stochastic Gradient Descent (alpha range 1-2)	0.2823
Count features, BoW TFIDF	KNN (nn =11)	0.5645
Count features, BoW TfidfVectorizer	Gradient Boosting (default)	0.8346
Count features, BoW CountVectorizer	XGBoost(objective ='multi:softmax', colsample_bytree = 0.3, learning_rate = 0.1, max_depth = 5, alpha = 10, n_estimators = 10)	0.7379
Count features, BoW TfidfVectorizer	XGBoost(objective ='multi:softmax', colsample_bytree = 0.3, learning_rate = 0.1, max_depth = 5, alpha = 10, n_estimators = 10)	0.7661
Count features, BoW TfidfVectorizer	XGBoost(learning_rate =0.1, n_estimators=1000, max_depth=5, min_child_weight=1, gamma=0, subsample=0.8, colsample_bytree=0.8, objective= 'binary:logistic', nthread=4, scale_pos_weight=1,seed=27)	0.8266
Count features, BoW TfidfVectorizer	XGBoost Grid Search	
Count features, BoW TFIDF	Random Forest	
BCount features, BoW TFIDF	Random Forest - random search	
Count features, 1-2 gram TFIDF	Log Reg	
Count features, 1-2 gram TFIDF	MNB	