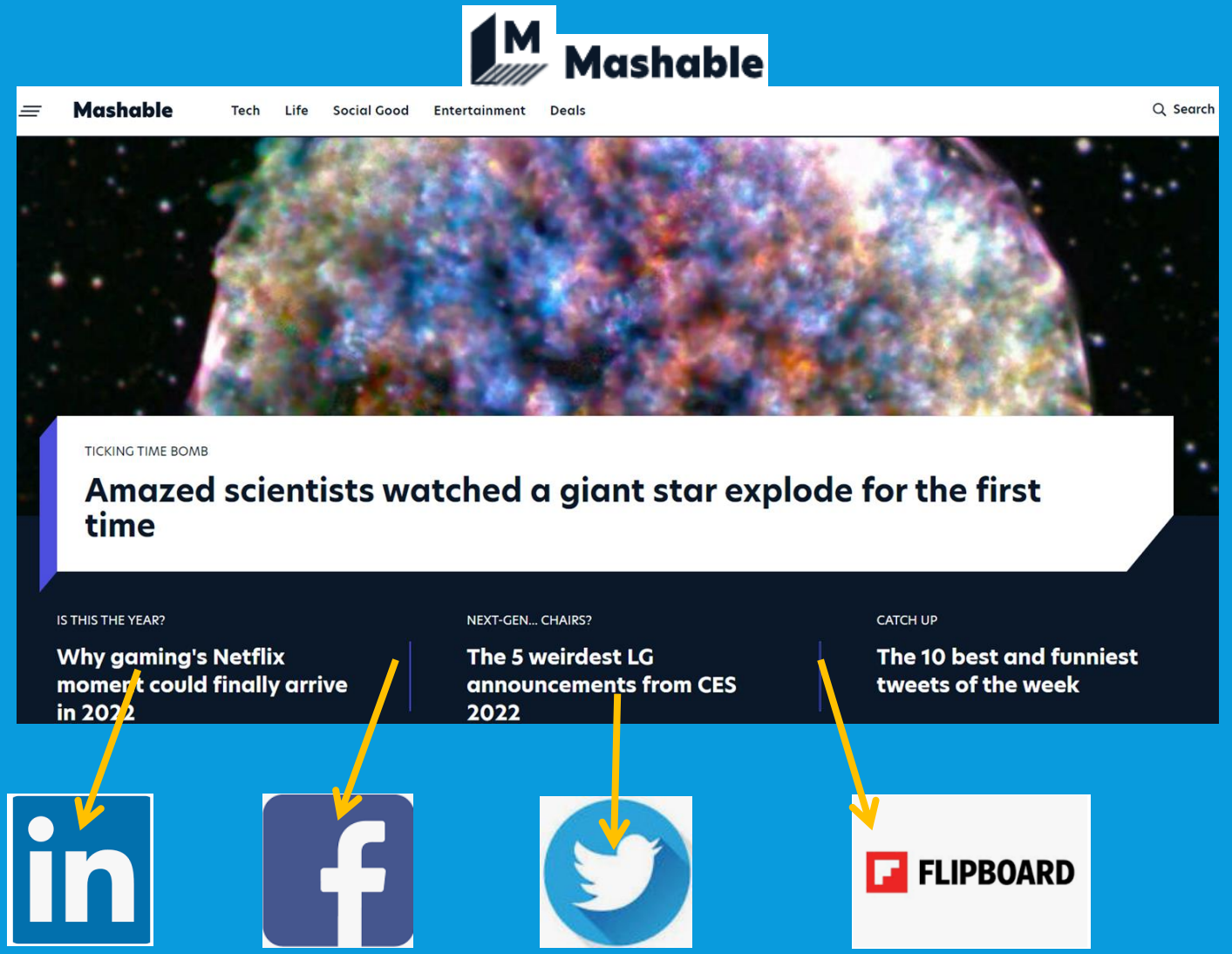
Predicting Popularity of Mashable News Articles



Joseph P. Martin
Phase 2, Module 16:
Supervised Learning Capstone Presentation
January 10, 2022

Introduction

- **Dataset: Information about articles from news site mashable.com that have been shared on social media platforms.**
- **The dataset was obtained from Kaggle, but originally comes from this research project:**
  - **Fernandes et al., 2015. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News.**
  - **The research article was consulted for information about the dataset.**
  - **Dataset created by retrieving content of all articles published on Mashable from January 7, 2013 to January 7, 2015.**
- **The outcome of interest is the number of times each article is shared. There are numerous features included, which can be categorized as follows:**

Feature	Type (#)
Words	
Number of words in the title	number (1)
Number of words in the article	number (1)
Average word length	number (1)
Rate of non-stop words	ratio (1)
Rate of unique words	ratio (1)
Rate of unique non-stop words	ratio (1)
Links	
Number of links	number (1)
Number of Mashable article links	number (1)
Minimum, average and maximum number of shares of Mashable links	number (3)
Digital Media	
Number of images	number (1)
Number of videos	number (1)
Time	
Day of the week	nominal (1)
Published on a weekend?	bool (1)

Feature	Type (#)
Keywords	
Number of keywords	number (1)
Worst keyword (min./avg./max. shares)	number (3)
Average keyword (min./avg./max. shares)	number (3)
Best keyword (min./avg./max. shares)	number (3)
Article category (Mashable data channel)	nominal (1)
Natural Language Processing	
Closeness to top 5 LDA topics	ratio (5)
Title subjectivity	ratio (1)
Article text subjectivity score and its absolute difference to 0.5	ratio (2)
Title sentiment polarity	ratio (1)
Rate of positive and negative words	ratio (2)
Pos. words rate among non-neutral words	ratio (1)
Neg. words rate among non-neutral words	ratio (1)
Polarity of positive words (min./avg./max.)	ratio (3)
Polarity of negative words (min./avg./max.)	ratio (3)
Article text polarity score and its absolute difference to 0.5	ratio (2)

- ***Research Question Addressed by Modeling: How well can we predict the number of times an article will be shared, prior to publication, based on article properties (features)?***

Introduction

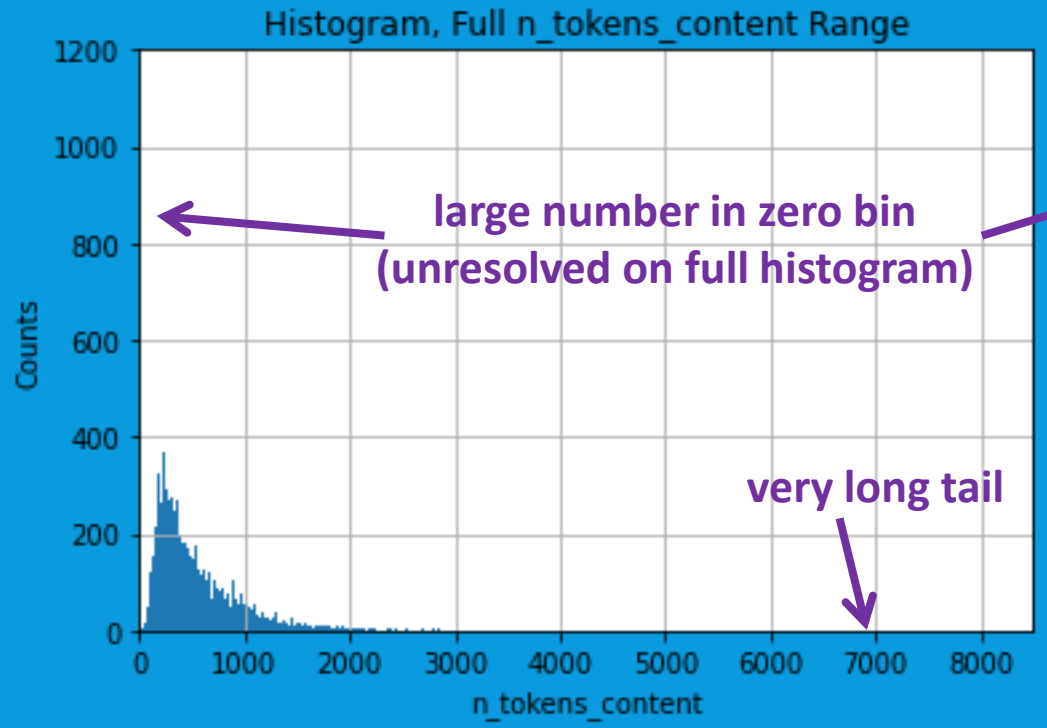
- Mashable articles carry metadata, such as keywords, data channel type, and total number of shares (Facebook, Twitter, Google+, LinkedIn, Stumble-Upon, Pinterest).

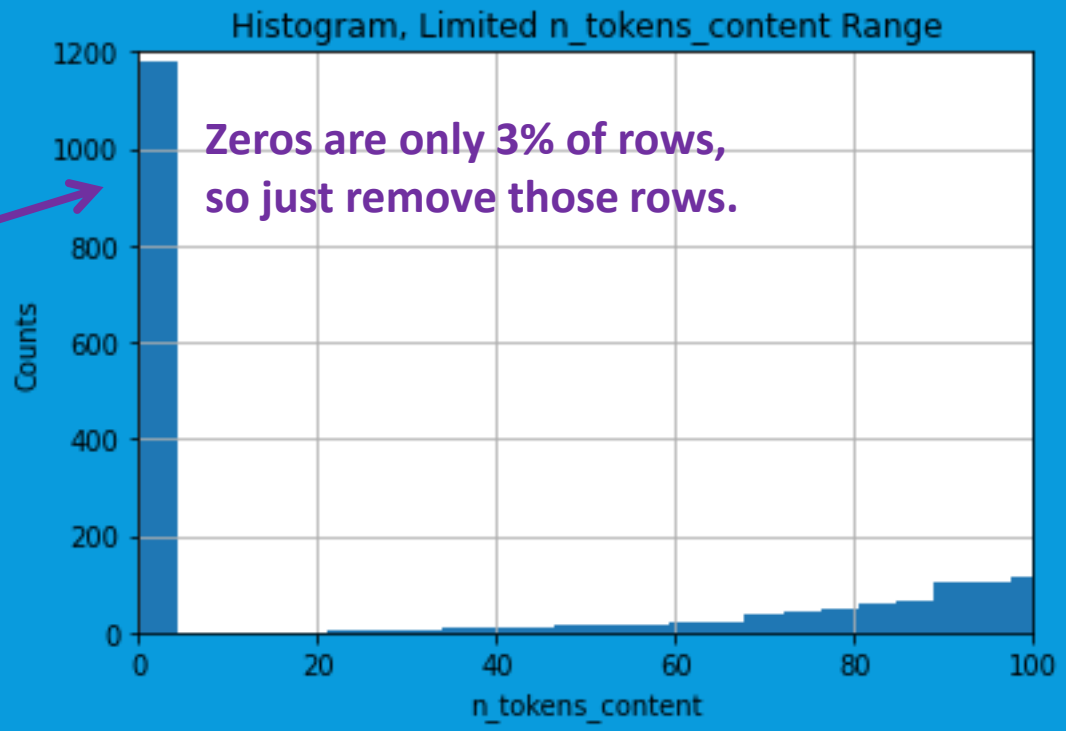
- Categorical variables were already encoded into multiple binary variables.

Data Exploration

- After reading the dataset into a pandas DataFrame, there are 39,644 rows and 61 columns.
- There are not any missing (null) data in any of the columns.
- We use the describe() method to look at the basic statistics of each column: range, quartiles, mean, and standard deviation. This allows us to assess the distribution of data including whether there are outliers. We then further explore any columns for which there is concern about the distribution / outliers. A few examples follow …

n_tokens_content: number of words in content

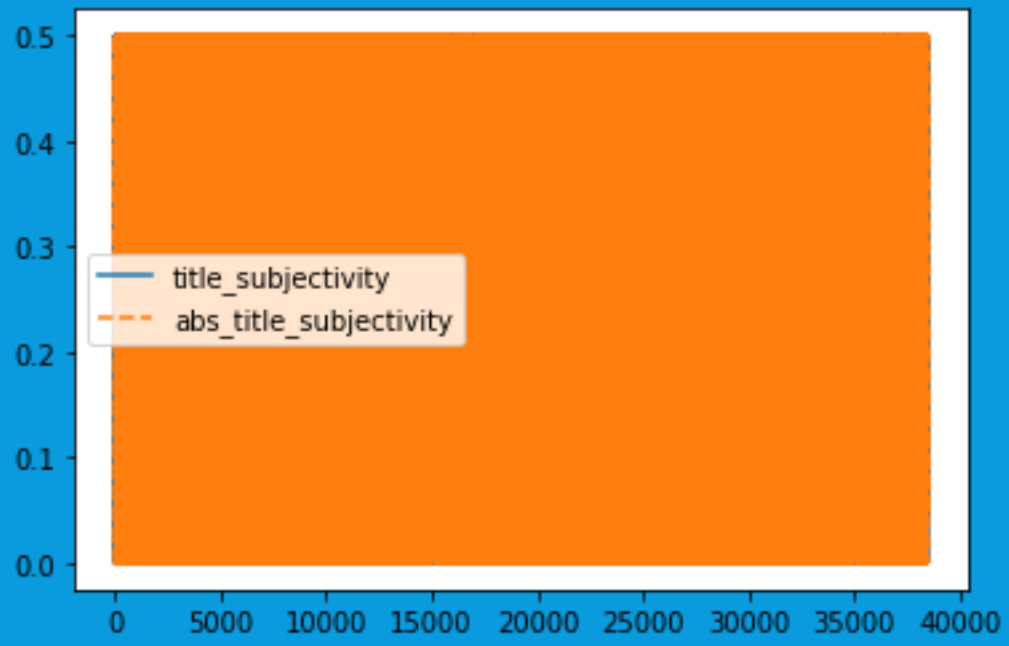


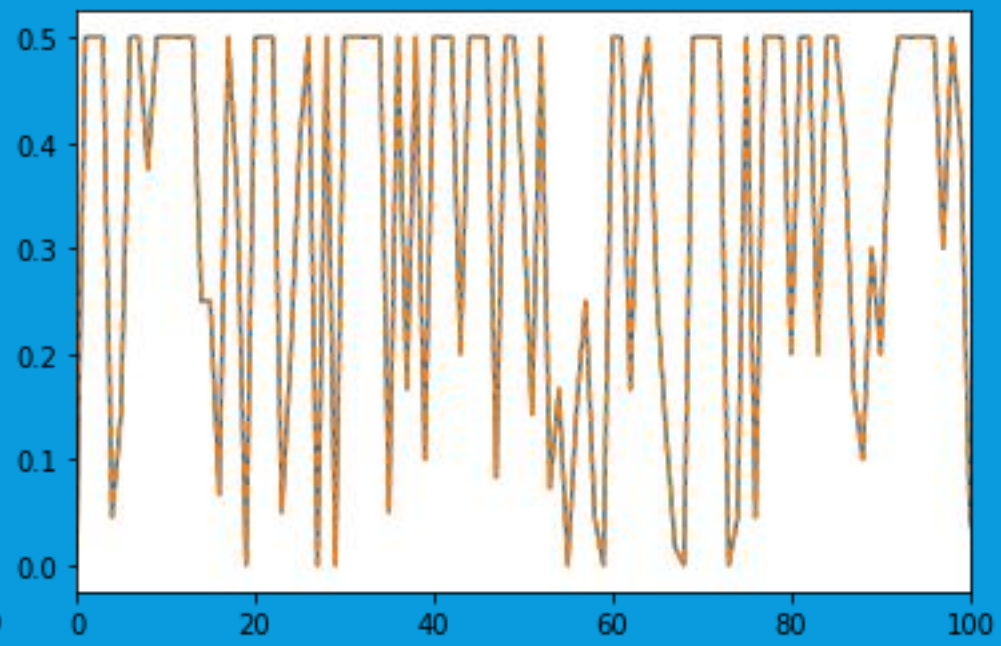


Data Exploration

- **Some of the feature columns can be determined from another feature column, such as:**

abs_title_subjectivity = |title_subjectivity – 0.5|

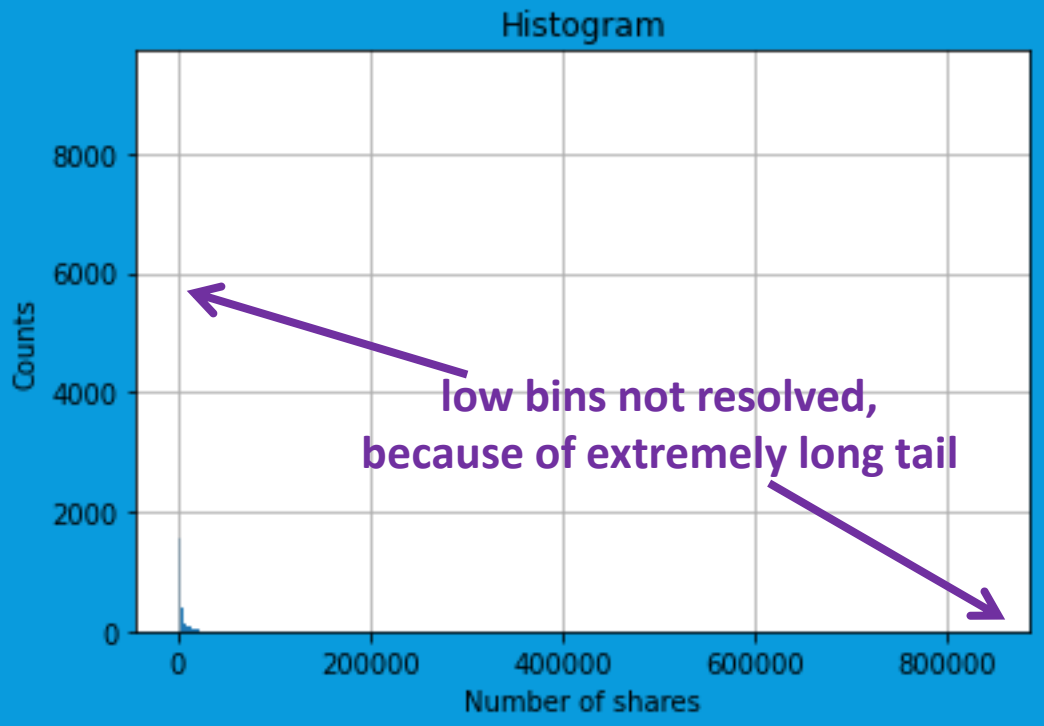


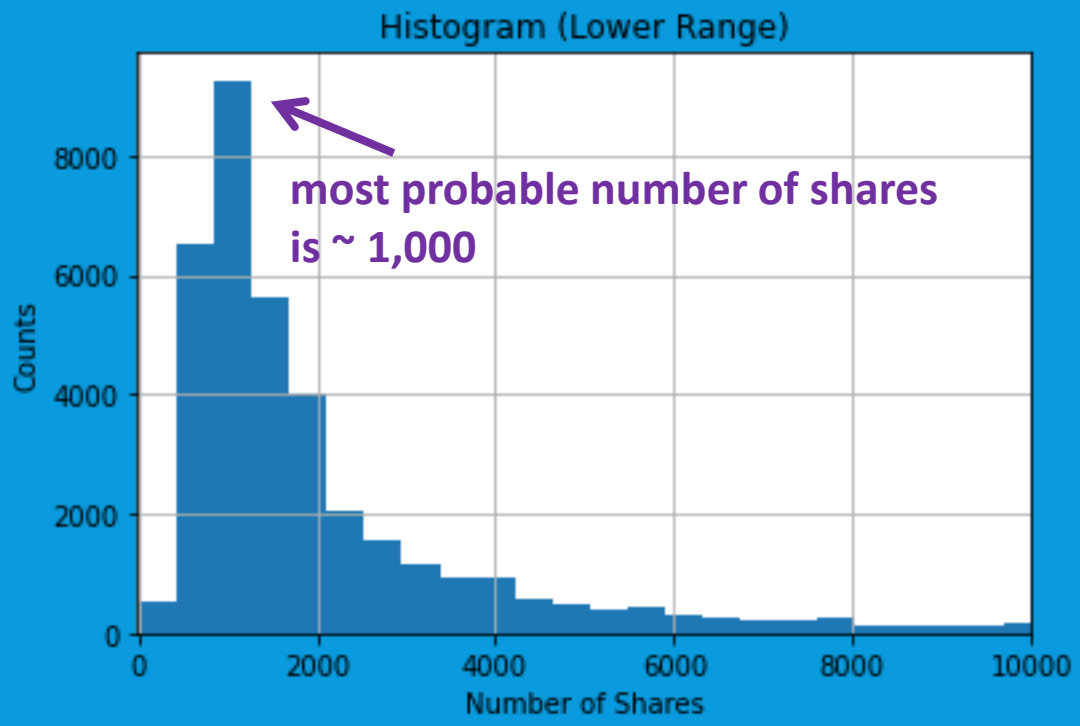


- **In these cases, one of the columns must be removed.**

Data Exploration

- **Distribution of the target outcome variable, the number of shares:**





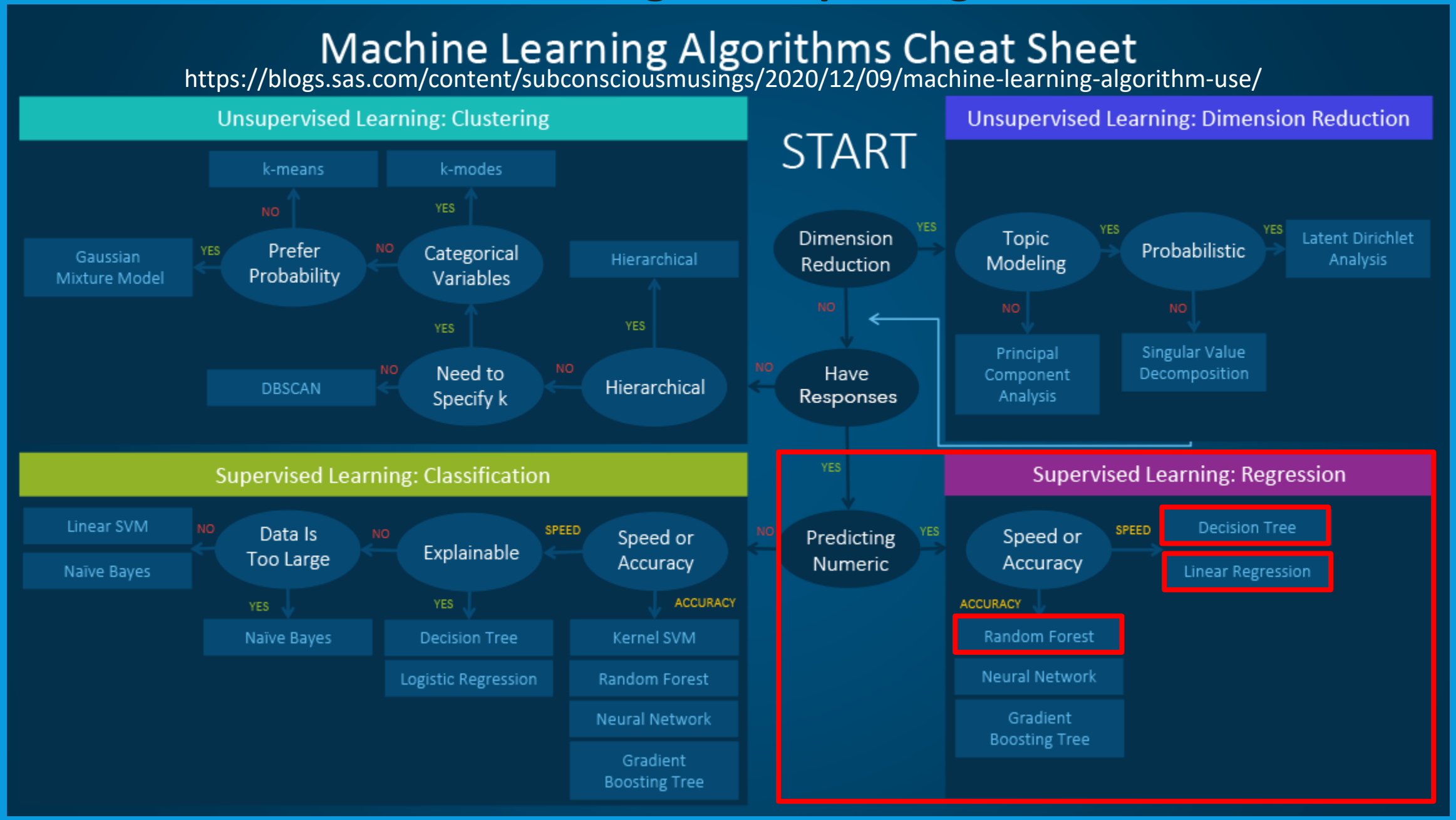
- **The number of shares is clearly not normally distributed. Most articles are at the lower end of the overall range, but then the maximum number of shares extends all the way to 843,300. There is an extremely long tail.**

- **At the end of the data exploration and cleaning process, acceptability of descriptive statistics is confirmed.**

Initial Modeling Attempt: Regression

- Because the outcome 'shares' is numerical, the models we initially attempted were regression models.

Initial Modeling Attempt: Regression



Initial Modeling Attempt: Regression

- Because the outcome 'shares' is numerical, the models we initially attempted were regression models.
- Tried regression models we learned about in this course: (1) linear regression, (2) decision tree, and (3) random forest.
- All of the models performed poorly even though they were explored exhaustively:
  - Ordinary Least Squares (OLS):
    - apply logarithmic transformations to variables with large skewness
    - if pair of features has correlation above specified threshold, exclude one of the features
    - remove features with insignificant regression coefficients until there are only significant ones remaining
    - optimal OLS model: R-squared of 0.117 for the training data and 0.107 for the test data
  - Decision Tree Regression:
    - without and with logarithmic transformations
    - excluded sparse features
    - adjust parameters in DecisionTreeRegressor: max_depth, max_features, and min_samples_leaf
    - the coefficient of determination score returned was typically negative, and went slightly above 0.0 with changes to min_samples_leaf and max_depth -> indicates the model disregards the features
  - Random Forest Regression:
    - without and with logarithmic transformations
    - excluded sparse features
    - negative scores with default parameters. how can it work if decision tree does not?

Do not understand why these perform so poorly.

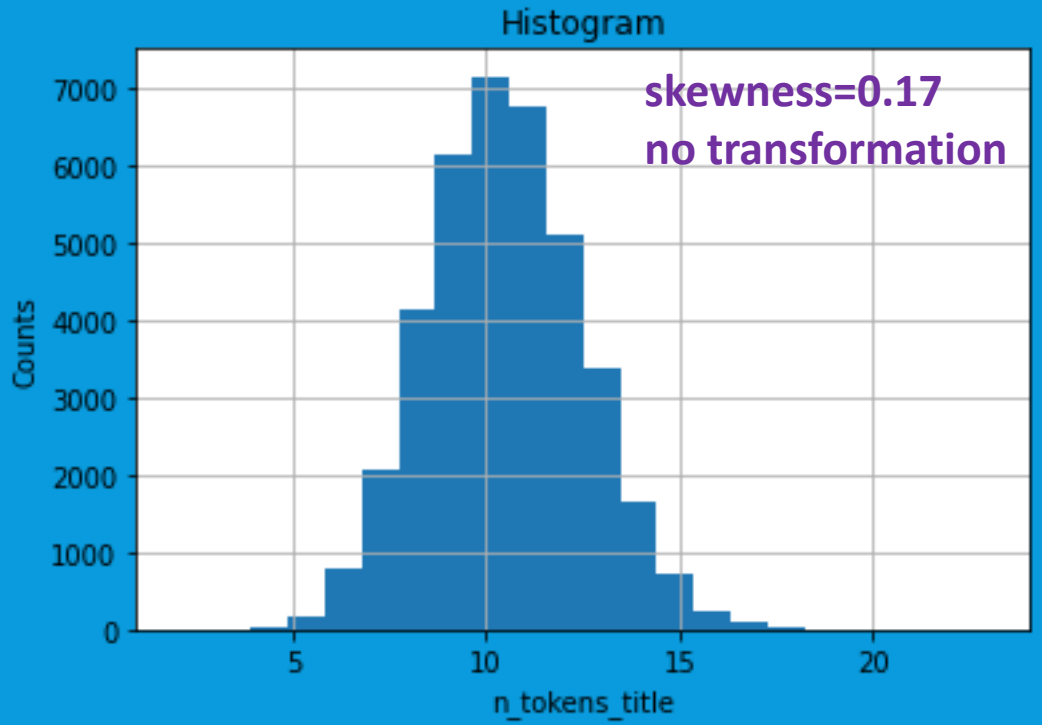
Classification Modeling

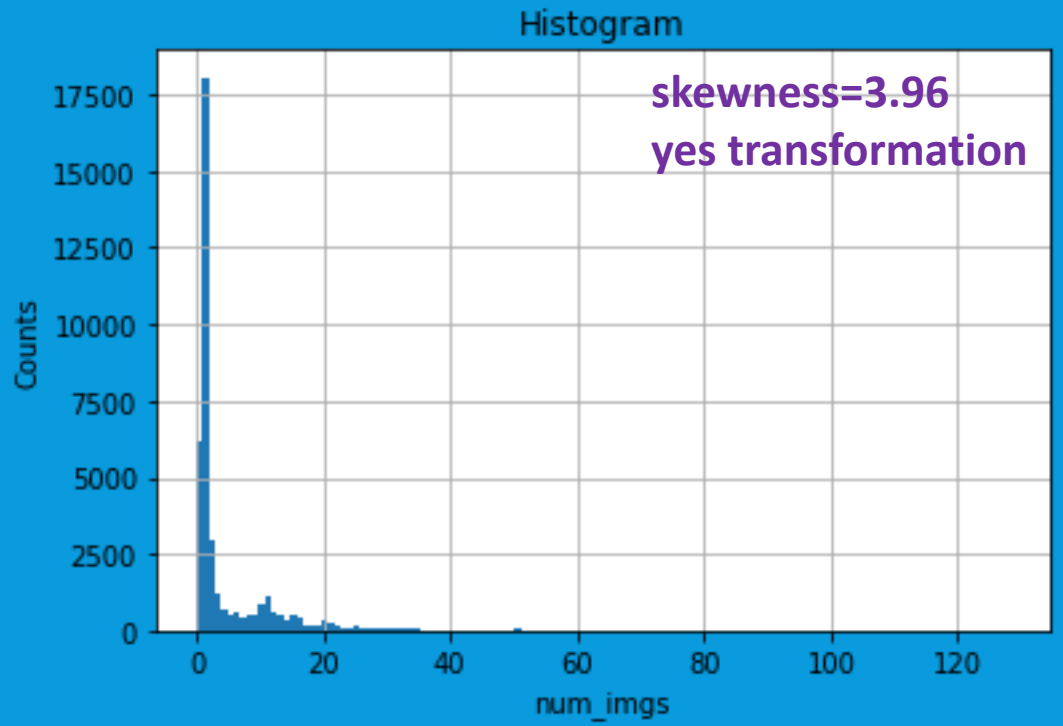
- Because regression modeling did not work, we switch to classification modeling.
- Follow what the authors of the original dataset and article did with target outcome shares.
- Change shares from numerical variable to binary classification variable:
  - The median number of shares is 1,400.
  - Shares less than 1,400 are set to 0 for unpopular.
  - Shares greater than or equal to 1,400 are set to 1 for popular.
- Remove feature n_non_stop_words, because its standard deviation is extremely small relative to its mean.
- Remove variable is_weekend because it is a linear combination of is_saturday and is_sunday.

Classification Modeling

- Apply logarithmic transformations to features that are unbound and have a highly skewed distribution (|skewness| > 3).
- In regression modeling, applying logarithmic transformations provided some performance benefit with no downside.

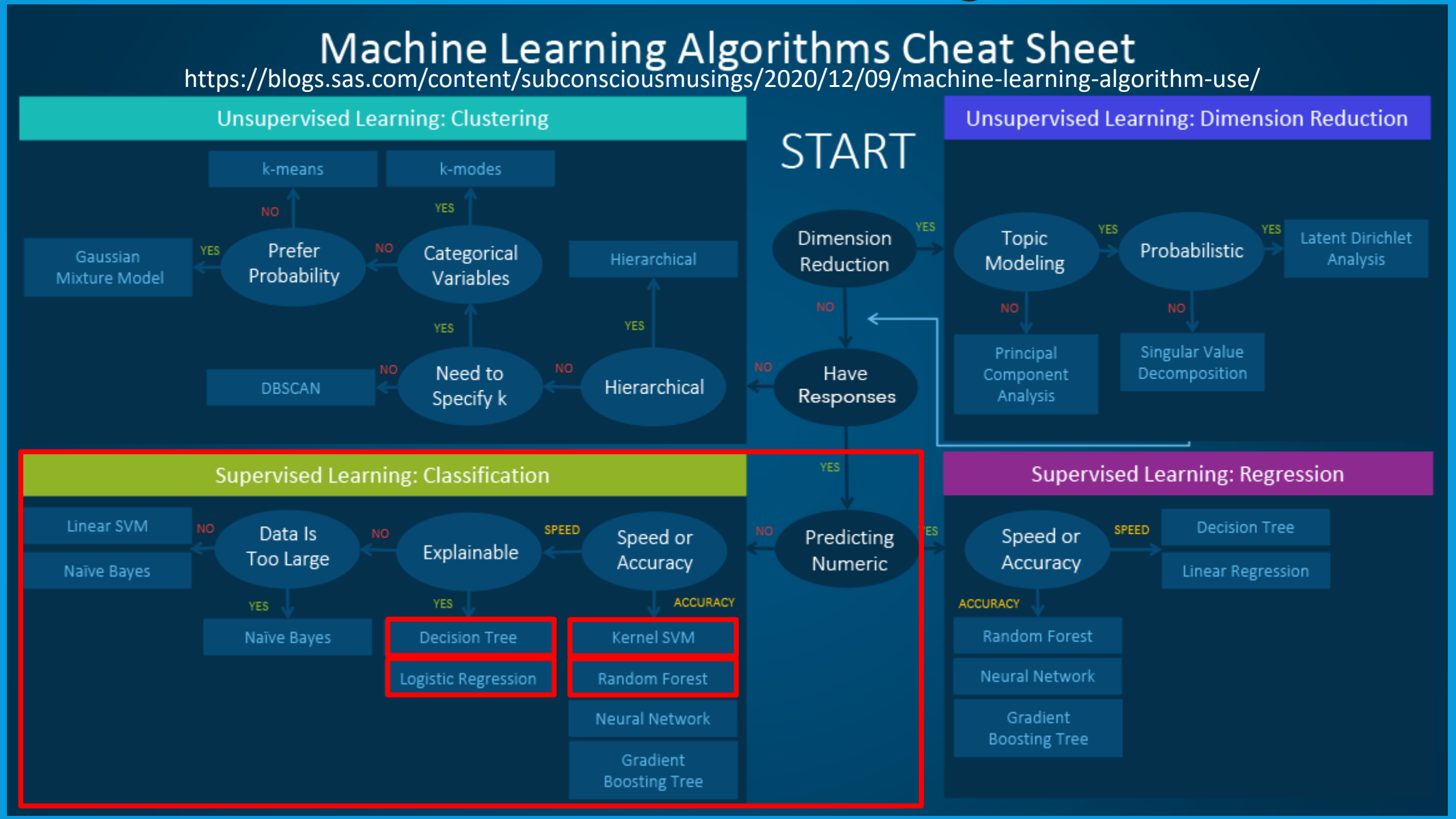
Examples





- Apply logarithmic transformations to all kw_*_* features because 5 of 9 have |skewness| > 3 and for consistency.
- logarithmic transformation: x_new = log10(x+1). The +1 is needed to handle 0s.
- *Split data into training (80%) and test (20%) sets.*

Classification Modeling



Classification Modeling: Logistic Regression

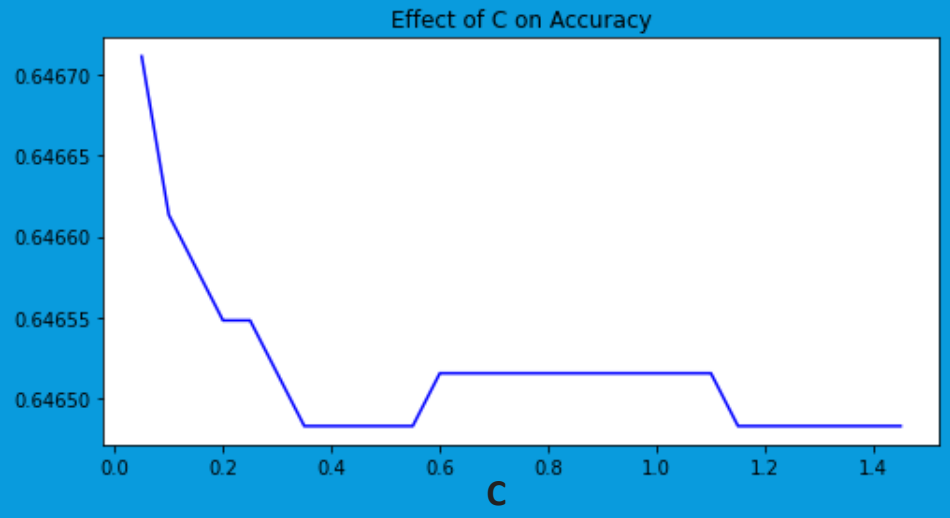
- **Determined correlations among feature pairs. Top correlations:**

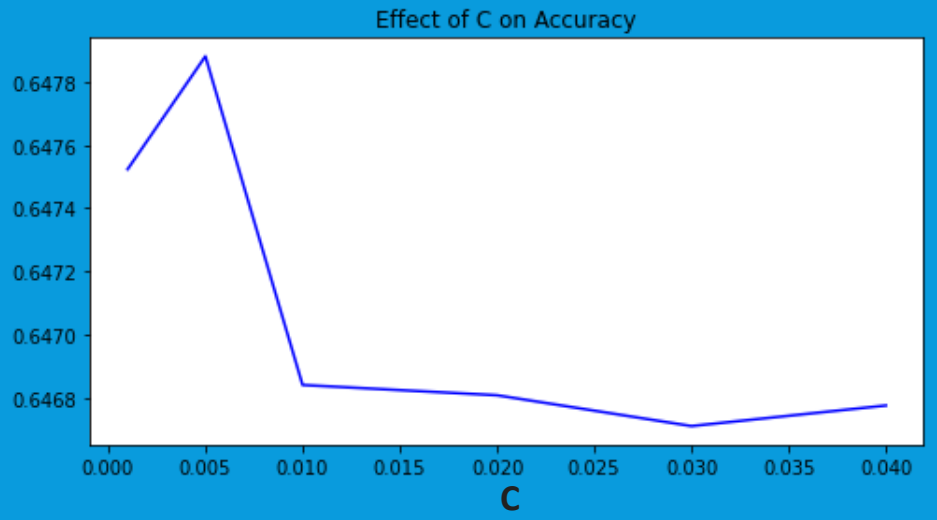
feature_1	feature_2	abs_correlation
rate_positive_words	rate_negative_words	0.997693
log_self_reference_max_shares	log_self_reference_avg_sharess	0.993977
log_kw_min_max	log_kw_min_avg	0.982953
log_self_reference_min_shares	log_self_reference_avg_sharess	0.966860
log_kw_avg_min	log_kw_max_min	0.944365
log_self_reference_max_shares	log_self_reference_min_shares	0.938316
log_kw_avg_max	log_kw_max_max	0.908581
log_kw_max_avg	log_kw_avg_avg	0.892042
n_non_stop_unique_tokens	n_unique_tokens	0.885136
log_n_tokens_content	n_unique_tokens	0.883158
data_channel_is_world	LDA_02	0.835612

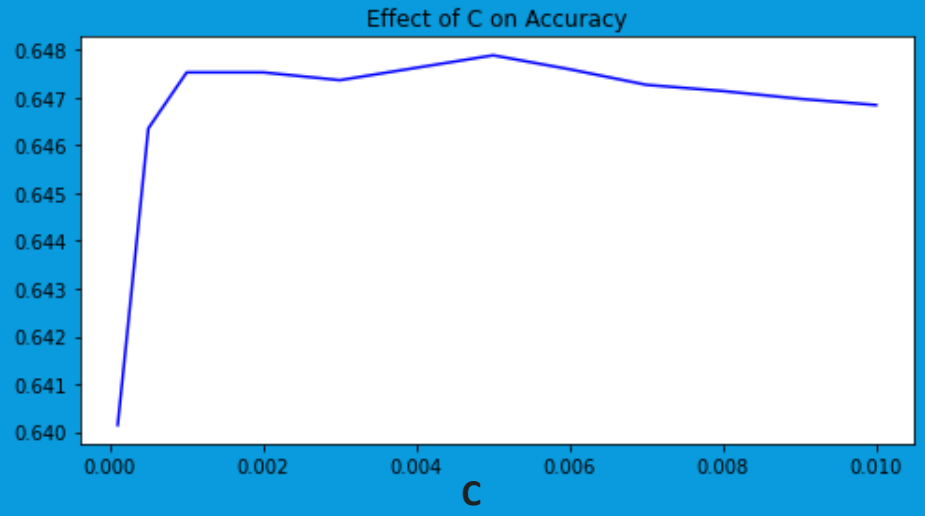
- **Features are scaled to zero mean and unit variance.**
- **Use 10-fold cross validation to investigate model performance.**
- **Test ‘solver’ and ‘C’ parameters while excluding features in feature_2 list with |correlation| > 0.8 (above list).**
- **‘solver’: algorithm used in optimization problem. Among 'lbfgs’, ‘sag’, and ‘saga’, *‘sag’* is best so used in optimal model.**

Classification Modeling: Logistic Regression

- Vary parameter ‘C’. Smaller values of ‘C’ specify stronger regularization (penalize complexity to help test performance).







- C = 0.005 results in the best accuracy so used in optimal model.

Classification Modeling: Logistic Regression

- Adjust correlation threshold for which features are used in the optimal model.
- The accuracy is actually largest when the correlation threshold is 1.00, that is, all features are included.
- Apparently, logistic regression can handle highly correlated features as long as the regularization is strong enough, which we achieved by setting C = 0.005.

Results

Model	Data	Accuracy Score
Default	Training	0.647913
Optimal	Training	0.655811
Optimal	Test	0.652756

- There is a small improvement in the training score after optimizing the model.
- The optimal logistic regression model appears to perform similarly for data within and outside the training data.

Top 5 Positive Coefficients

features	coefficients
log_kw_avg_avg	0.498426
weekday_is_saturday	0.216905
data_channel_is_socmed	0.183144
LDA_00	0.176966
weekday_is_sunday	0.158765

Top 5 Negative Coefficients

features	coefficients
log_kw_max_max	-0.205046
data_channel_is_entertainment	-0.190987
log_num_self_hrefs	-0.157781
LDA_02	-0.147306
log_kw_min_avg	-0.114621

Classification Modeling: Logistic Regression

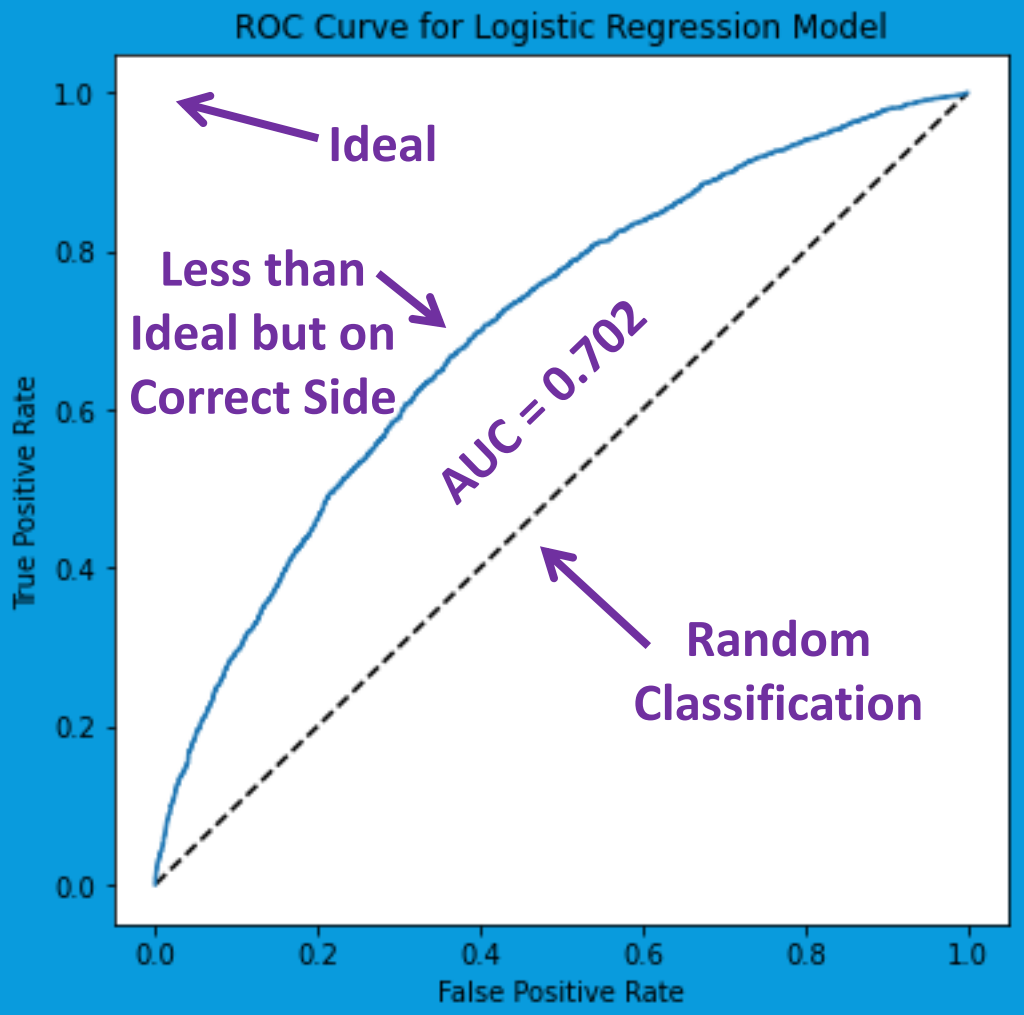
Results

	True Class Popular	True Class Not Popular
Predicted Class Popular	TP = 2830	FP = 1426
Predicted Class Not Popular	FN = 1245	TN = 2191

- accuracy = (TP+TN)/(TP+TN+FP+FN) = 0.652756
- precision = TP/(TP+FP) = 0.664944 (proportion of positive predictions that are correct)
- recall = TP/(TP+FN) = 0.694479 (proportion of instances in the positive class are correctly predicted)

Classification Modeling: Logistic Regression

Results

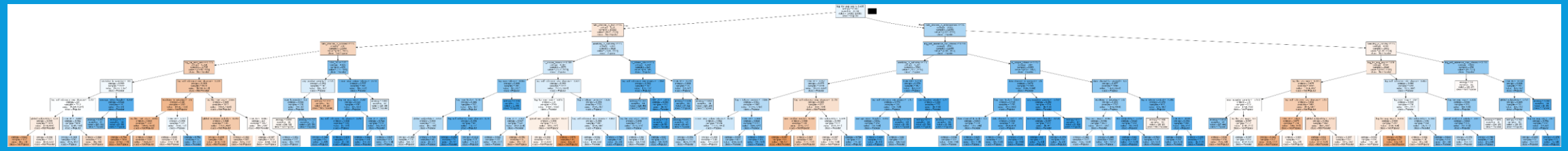


Classification Modeling: Decision Tree

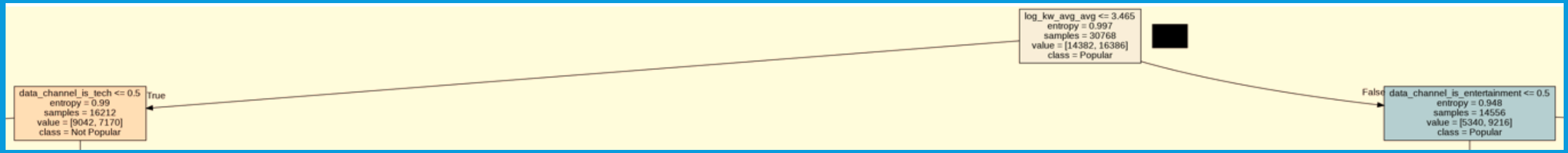
- Use 10-fold cross validation on training data.
- The trees produced with default parameters are too complex, with too much depth and too many leaves. They are likely overfitting the test set in each fold.
- Look for the optimal max_depth, the maximum depth of the trees. Accuracy score is best at max_depth = 7.
- Look for the optimal min_samples_leaf, which is the minimum number of samples required to be at a leaf node. Increasing it may have the effect of smoothing the model. Overall, the mean score does not vary much with a wide range of min_samples_leaf values. The optimal value is min_samples_leaf = 38.
- For the criterion parameter, ‘entropy’ results in a slightly larger accuracy than ‘gini’.
- The mean accuracy of the optimal decision tree model in predicting the test data is 0.643656. This is very close to and slightly larger than the average we obtained using cross validation on the training data (0.640243).

Classification Modeling: Decision Tree

- This is the optimal tree:



- This is the top of the optimal tree:



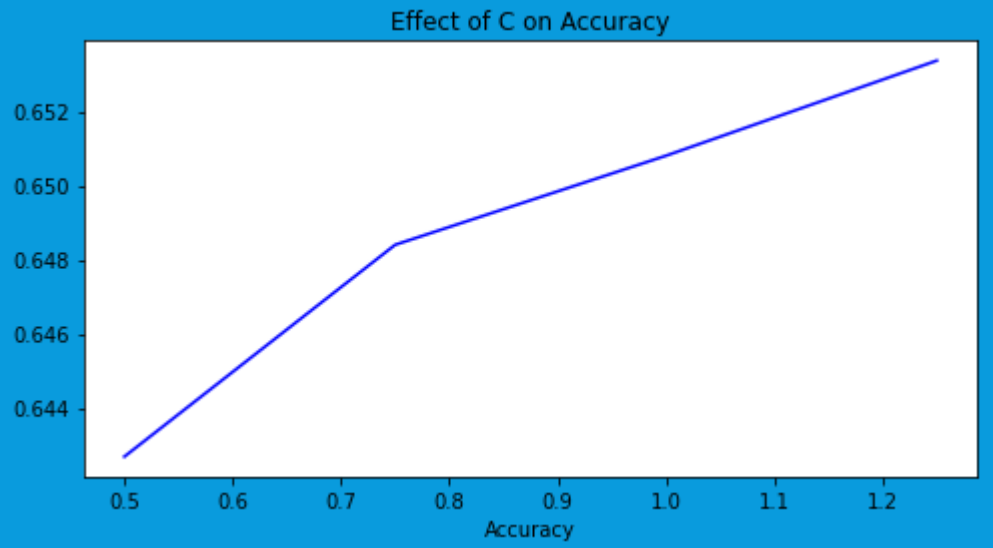
- The top node rule is log_kw_avg_avg <= 3.465.
- On the true (left) side of the top node, the next node is data_channel_is_tech <= 0.5, which in turn has data_channel_is_socmed <= 0.5 on its true side and weekday_is_saturday <= 0.5 on its false side.
- On the false (right) side of the top node, the next node is data_channel_is_entertainment <= 0.5, which in turn has log_self_references_min_shares <= 3.218 on its true side and weekday_is_sunday <= 0.5 on its false side.

Classification Modeling: Random Forest

- The random forest model is setup with the parameters obtained in decision tree: criterion = ‘entropy’, max_depth = 7, and min_samples_leaf = 38.
- Use the default value for the number of trees in the forest, n_estimators = 100.
- Use 10-fold cross validation on training data.
- A parameter test is tried in turning off bootstrap, but that just lowers the accuracy.
- The mean accuracy score for the cross validation on the training data is 0.658119.
- The accuracy score for the test data is 0.664717.

Classification Modeling: Kernel Support Vector Machine

- We try one more model, kernel support vector machine classification (SVC).
- Kernel SVC allows nonlinear methods to be used. Linear support vector machine and logistic regression both optimize linear boundaries between classes, so looking at linear support vector machine may not add very much.
- The default kernel used in the algorithm is ‘rbf’ (radial basis function), which Thinkful says should be used.
- Look at impact of varying ‘C’. The strength of the regularization is inversely proportional to ‘C’.



- This shows the largest accuracy when regularization is diminished (C > 1), so that may be overfitting. Just leave C at 1.

Classification Modeling: Kernel Support Vector Machine

- **The accuracy score for the training data is 0.655746.**
- **The accuracy score for the test data is 0.653926.**

Summary

- How well does each optimized model perform on test data?

Model Type	Mean Accuracy
Logistic Regression	0.652756
Decision Tree	0.643656
Random Forest	0.664717
Kernel Support Vector Machine	0.653926

- Random Forest is the best, but the range of accuracies among the models is quite small.
- Any of the models would provide approximately the same predictive power.
- In the article cited at the top of this project, the authors obtained accuracies of 0.67 for random forest, which was their top model, and 0.66 for support vector machine. We have done nearly just as well here.

- The most important features in the optimal logistic regression are for positive coefficients: log_kw_avg_avg, weekday_is_saturday, data_channel_is_socmed, LDA_00, and weekday_is_sunday and for negative coefficients: log_kw_max_max, data_channel_is_entertainment, log_num_self_hrefs, LDA_02, and log_kw_min_avg.
- The most important features in the optimal decision tree are log_kw_avg_avg, data_channel_is_tech, data_channel_is_socmed, weekday_is_saturday, data_channel_is_entertainment, log_self_reference_min_shares, and weekday_is _sunday. Interestingly, there is clearly some overlap in the most important features in the two models, which have very different algorithms.

Summary

- **Practical Use of the Model: There is some ability to predict which articles are likely to be popular and shared, even before publication. The features here are properties that can be determined before publication. This could allow authors and editors to adjust articles so that they will be more popular and shared more. A concern here though is that articles could deviate from serious journalism toward clickbait in order to generate more shares.**

- **Weaknesses of the Models: Because the popular / not popular split was a deliberate 50% / 50% split on either side of the median, a random prediction would produce an accuracy of 0.5. Because our accuracies range from 0.64 to 0.66, the predictive capability is somewhat but not a lot larger than random. In a way, this is acceptable, because it would be impossible to successfully predict the popularity of every article.**