Online News Popularity Prediction

18 Apr ‘18

Pavani Komati (.1) and Vedasri Uppala (.1)

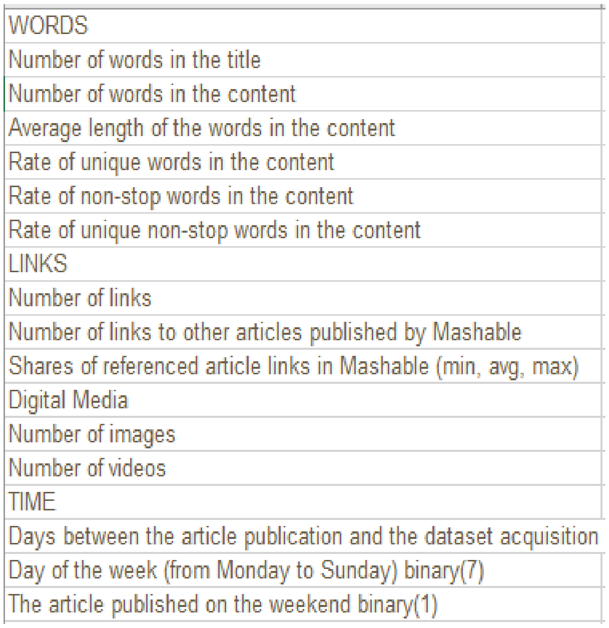
Overview and Business Goal

This project aims to predict the popularity of news articles published in Mashable over a period of two years. The Mashable website has been scraped and various features have been collected for each news article published. This data set is available at UCI repository. The data set has information of 39644 news articles and 61 features are collected for each of them. The ‘shares’ attribute which denotes the number of shares a news article has received on social media is the target variable. Predicting the number of shares a news article would receive can be an important business goal to the organizations. A new article to be published can be rephrased or formatted to the form of articles which received high popularity so that the current article gets maximal number of shares.

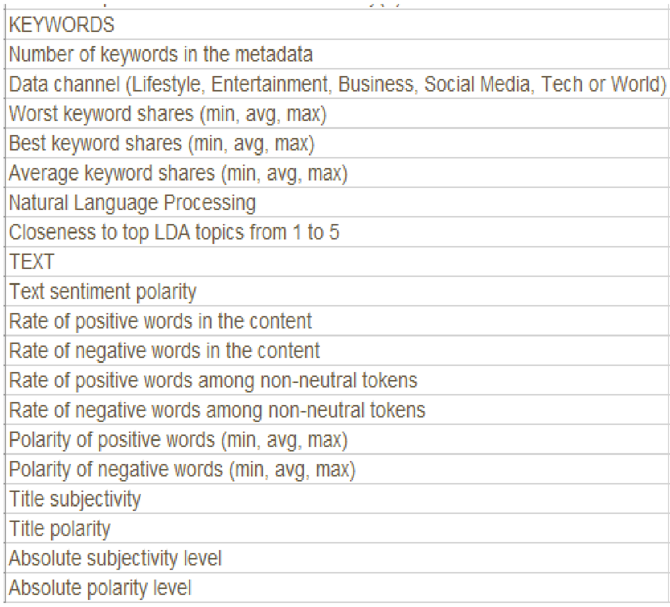
Data Set Description

The data set used for the project is available at UCI repository (here). It has 39644 instances of news articles and 61 features for each of them. The features contain information about URLs, temporal data statistics and sentiment analysis.

The following tables summarizes the attribute information.



WORDS
Number of words in the title
Number of words in the content
Average length of the words in the content
Rate of unique words in the content
Rate of non-stop words in the content
Rate of unique non-stop words in the content
LINKS
Number of links
Number of links to other articles published by Mashable
Shares of referenced article links in Mashable (min, avg, max)
Digital Media
Number of images
Number of videos
TIME
Days between the article publication and the dataset acquisition
Day of the week (from Monday to Sunday) binary(7)
The article published on the weekend binary(1)



KEYWORDS
Number of keywords in the metadata
Data channel (Lifestyle, Entertainment, Business, Social Media, Tech or World)
Worst keyword shares (min, avg, max)
Best keyword shares (min, avg, max)
Average keyword shares (min, avg, max)
Natural Language Processing
Closeness to top LDA topics from 1 to 5
TEXT
Text sentiment polarity
Rate of positive words in the content
Rate of negative words in the content
Rate of positive words among non-neutral tokens
Rate of negative words among non-neutral tokens
Polarity of positive words (min, avg, max)
Polarity of negative words (min, avg, max)
Title subjectivity
Title polarity
Absolute subjectivity level
Absolute polarity level

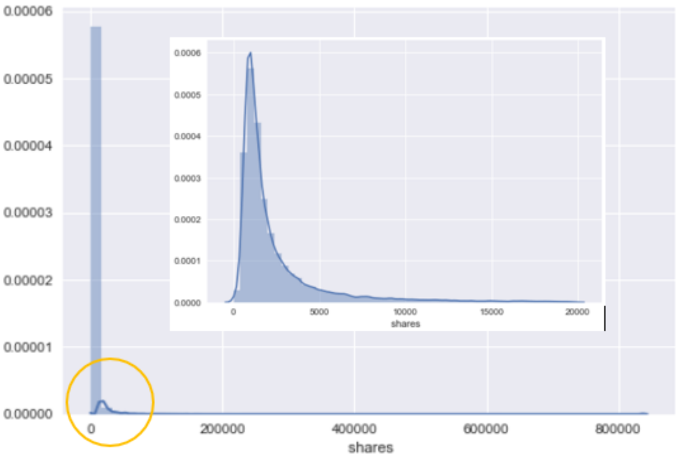
The entire data has been randomly divided into training and testing set. 75% of the data has been used for training the remaining data has been used for finding generalization error of the models.

Goals

1. Since the target variable is numeric, we have performed regression task on it.
2. The 'shares' attribute has been discretized into two categories "Popular" and "Unpopular" to perform Classification on it.
3. We have also performed clustering to identify if the data can be clustered.

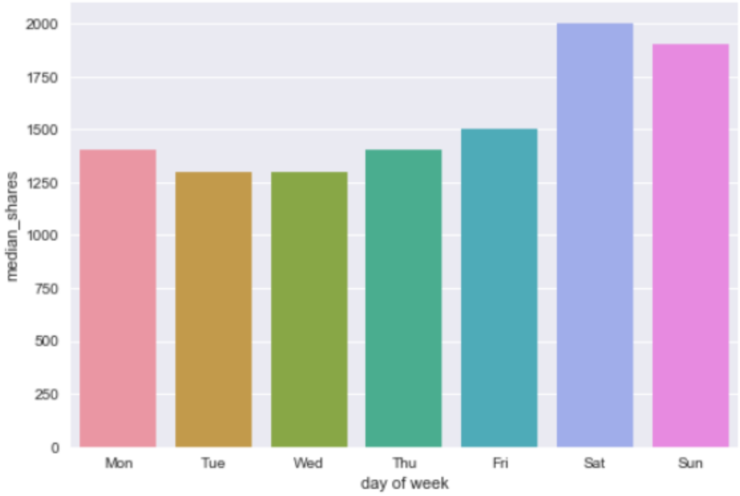
Exploratory Data Analysis

The following graph shows the distribution of target variable 'shares'.



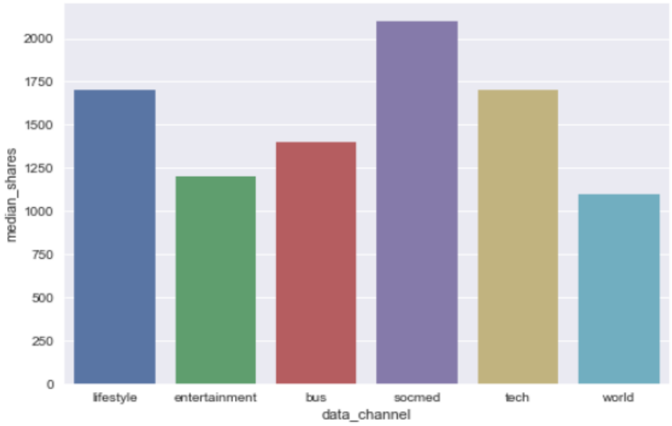
From the graph we can see that the distribution of the target variable is right skewed. So, mean of shares may not be a good measure of the center. This suggests an important action (log transformation) that can be taken during preprocessing.

The following graph gives information about the number of shares received by articles published on each week day.



From the graph, the articles published in the middle of the week received lesser number of shares than the articles published during the week end. So, this important insight can be used by publishers to publish new articles during the weekend so that they become more popular.

The news articles in the dataset belong to one of the six predefined categories (Lifestyle, Entertainment, Bus, Social Media, Technology and World). For each category, the median shares received in the social media has been calculated and the results are shown below.

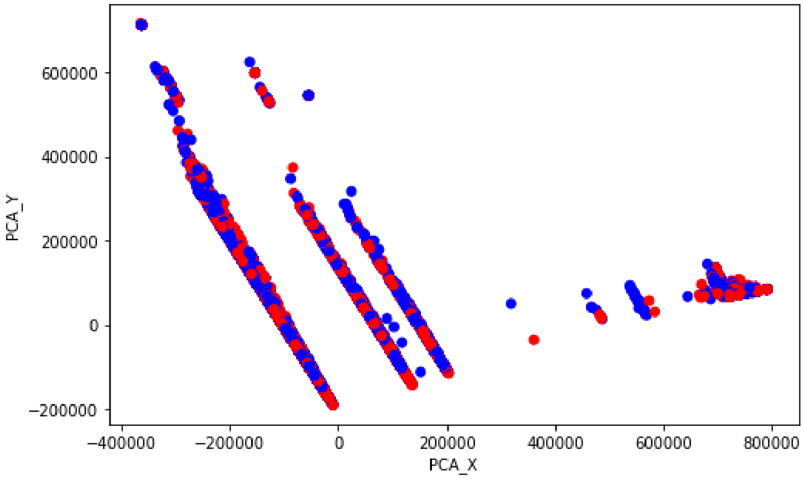


We can see that amongst the 6 categories news articles belonging to 'social media' category have received highest popularity in digital media

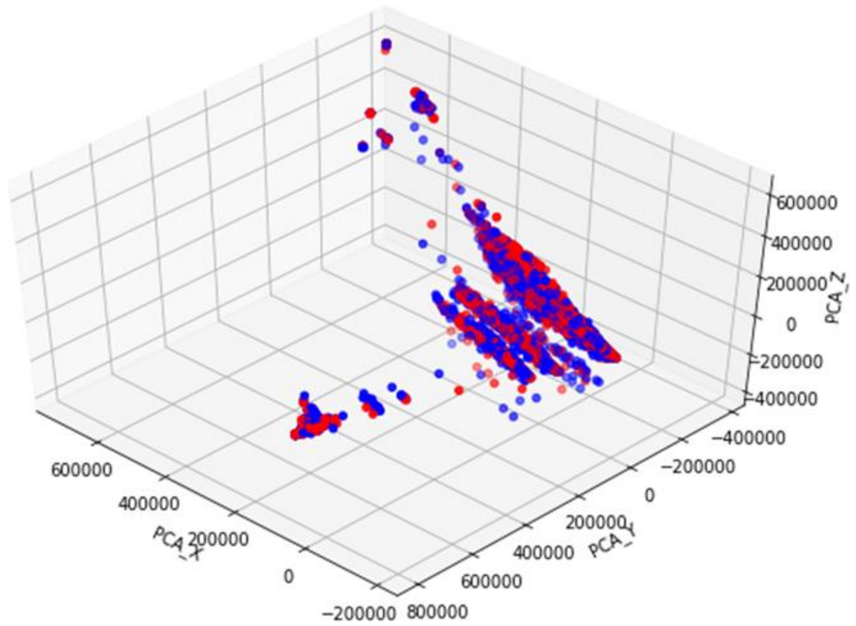
Principal Component Analysis

Principal Component analysis has been applied on the data set to analyze the covariance between the features. It has been observed that taking two principal components has explained 92.487% of the variance and three principal components together have explained 96.007% variance in the data. We have plotted the data points with the principal components and the results are shown below. To have better visualization the target attribute 'shares' has been discretized into 'popular' and 'unpopular' using median of shares (1400) as the threshold.

In the two graphs below, red dots denote the data points which are unpopular (shares <= 1400) and blue dots denote data points that are popular (shares > 1400)



This graph shows the data points projected into a 2-dimensional space using the two principal components of PCA.

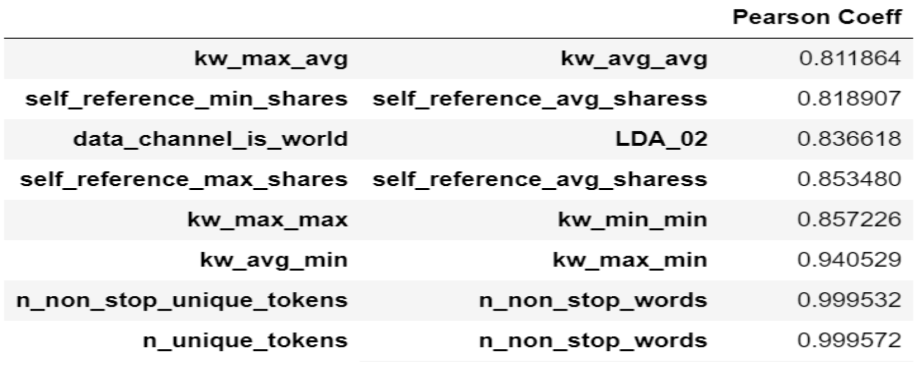


This graph shows the data points projected into a 3-dimensional space using the three principal components of PCA.

From the above two graphs it can be surprising to see that the PCA dimensions are not well suitable for discriminating popular and unpopular news articles. It is clearly evident that the new features do not demarcate the two classes. Therefore, we can conclude that the classes are not separable in the PCA-space. Since classification algorithms work well when there is a manifold that separates the classes, the classification models may perform only reasonably and not exceptionally despite meticulous feature selection.

Correlation Analysis

The data set has 58 predictive features and some of them might be correlated to each other. We have not included the entire correlation matrix because of its size (58 x 58). The table below shows some pairs of attributes which are highly correlated i.e whose Pearson Coefficient is greater than 0.85. In the following pairs of highly correlated attributes one in each pair is a potential candidate for removal and this will be handled in the preprocessing section.



		Pearson Coeff
kw_max_avg	kw_avg_avg	0.811864
self_reference_min_shares	self_reference_avg_sharess	0.818907
data_channel_is_world	LDA_02	0.836618
self_reference_max_shares	self_reference_avg_sharess	0.853480
kw_max_max	kw_min_min	0.857226
kw_avg_min	kw_max_min	0.940529
n_non_stop_unique_tokens	n_non_stop_words	0.999532
n_unique_tokens	n_non_stop_words	0.999572

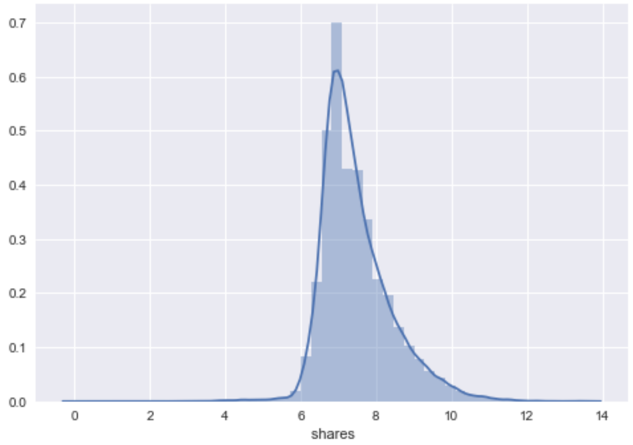
Data Preprocessing

Handling Missing Values

The UCI machine learning repository which hosts the current data set has indicated that the data set has no missing values. Taking this fact into consideration we have not performed any missing value analysis.

Data Transformation

As mentioned in the analysis section, the distribution for the target variable is right skewed. So, we have applied logarithm transformation to the target variable which will give it a more uniform distribution. This will be useful to reduce the range of values and to dampen the effect of extreme values.



Previously the distribution for 'shares' has been right skewed. The graph on the left shows the distribution for 'shares' after applying the log transformation. It is clearly evident that the distribution is more uniform now.

The data set has 44 attributes and each attribute has different range. This will be a problem for some modeling techniques. So, we have normalized the data using customized coding to fit into 0-1 range.

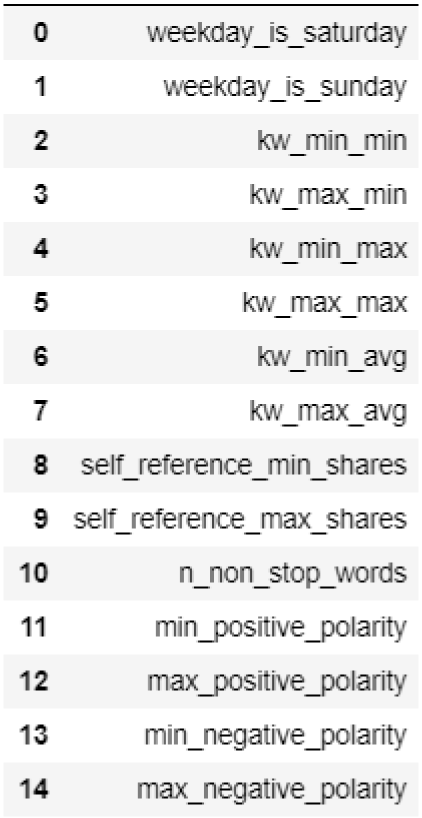
Feature Subset Selection

The data set has two attributes 'url' and 'timedelta' which denote the url of the published news article and the number of days between article publication and the data set acquisition. Clearly, we can see that these two attributes do not possess any predictive power and cannot contribute to predicting the popularity of a new news article. So, these features have been eliminated from the data which will be used for model building.

In the earlier section we have discussed that certain pairs of attributes have very high correlation. So, these attributes contain redundant information and can be dropped without loss of data useful for predicting the popularity of the news articles.

The data set has binary attributes ‘is_weekday_saturday’ and ‘is_weekday_sunday’ which denote whether the article is published on a saturday or sunday respectively. In addition to this there is another attribute ‘is_weekend’ which indicates whether the article is published on a weekend. Since this is a redundancy of information we have eliminated the attributes ‘is_weekday_saturday’ and ‘is_weekday_sunday’.

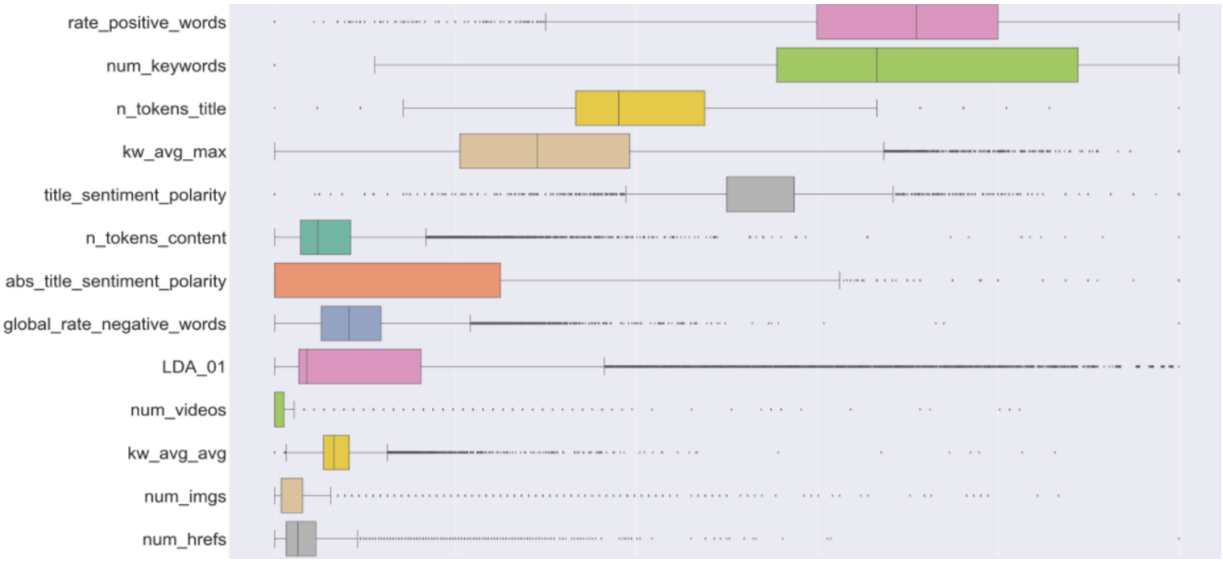
The data set has features ‘min_positive_polarity’, ‘max_positive_polarity’ and ‘avg_positive_polarity’ which denote the minimum, maximum and average polarity of positive words in the content of the news articles. Clearly, we are only interested in the average polarity of the positive words in the news article and not the minimum and maximum polarities. The correlation between such pair of attributes is also very high. So, these features have been eliminated from the data set. There are other sets of attributes having similar problems and same rules have been applied to all of them. The table consists of list of features removed using this logic.



0	weekday_is_saturday
1	weekday_is_sunday
2	kw_min_min
3	kw_max_min
4	kw_min_max
5	kw_max_max
6	kw_min_avg
7	kw_max_avg
8	self_reference_min_shares
9	self_reference_max_shares
10	n_non_stop_words
11	min_positive_polarity
12	max_positive_polarity
13	min_negative_polarity
14	max_negative_polarity

Data Cleaning

After removing the attributes mentioned earlier the data set has been checked for any potential outliers. A box plot has been created to visualize these outliers from few attributes. It can be observed that most of the data is shown as outliers in the plot.



The news articles are checked for outliers using the 1.5*IQR rule i.e. values that are beyond Q3 + 1.5*IQR and values that are below Q1 - 1.5*IQR are considered as outliers. Total of 37,533(i.e., 95% of the data) records are detected as outliers by applying this rule. Hence all the data shown as outliers in the plot cannot be considered as erroneous data and should not be eliminated.

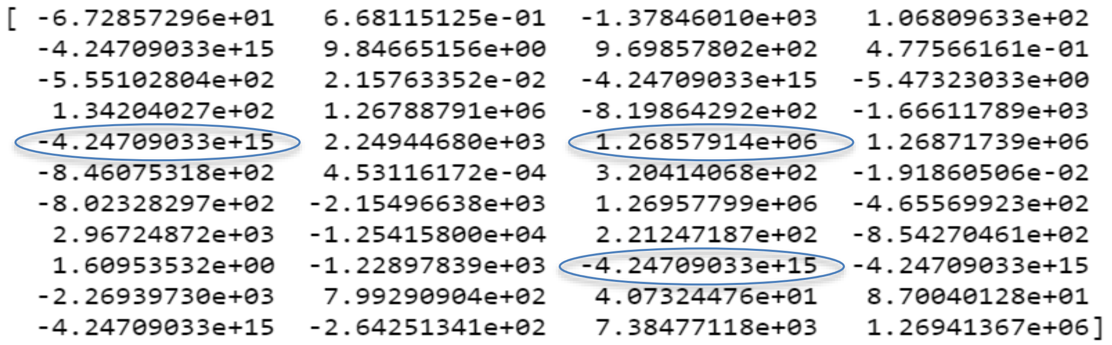
Model building

1. Regression

In the given data set, the target variable 'shares' is a numeric attribute. So, regression task can be performed to predict the number of shares given a news article. We have performed Ordinary Least Squares Regression, Logistic Regression and Lasso Regression. The findings are outlined below.

❑ Ordinary Least Squares Regression

Linear Regression has been applied to the data set with 44 attributes. Normalized data has been used to build the model and to test the results. After creating the model, the coefficients that the model has assigned to each of the 44 features are given below.

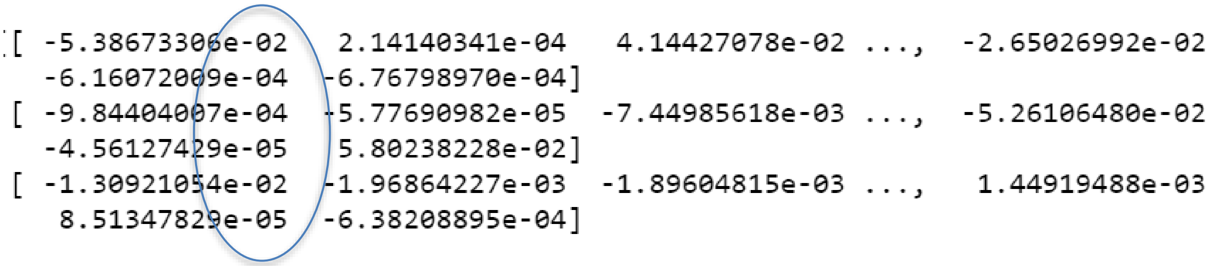


```
[ -6.72857296e+01   6.68115125e-01  -1.37846010e+03   1.06809633e+02
  -4.24709033e+15   9.84665156e+00   9.69857802e+02   4.77566161e-01
  -5.55102804e+02   2.15763352e-02  -4.24709033e+15  -5.47323033e+00
   1.34204027e+02   1.26788791e+06  -8.19864292e+02  -1.66611789e+03
  -4.24709033e+15   2.24944680e+03   1.26857914e+06   1.26871739e+06
  -8.46075318e+02   4.53116172e-04   3.20414068e+02  -1.91860506e-02
  -8.02328297e+02  -2.15496638e+03   1.26957799e+06  -4.65569923e+02
   2.96724872e+03  -1.25415800e+04   2.21247187e+02  -8.54270461e+02
   1.60953532e+00  -1.22897839e+03  -4.24709033e+15  -4.24709033e+15
  -2.26939730e+03   7.99290904e+02   4.07324476e+01   8.70040128e+01
  -4.24709033e+15  -2.64251341e+02   7.38477118e+03   1.26941367e+06]
```

From the screenshot we can see that for some attributes the coefficients are very high. This suggests that the model might have overfitted the data. The model has been applied to test data and predictions have been generated. The Coefficient of Determination (R squared) has been calculated to study the goodness of the fit.

❑ Logistic Regression

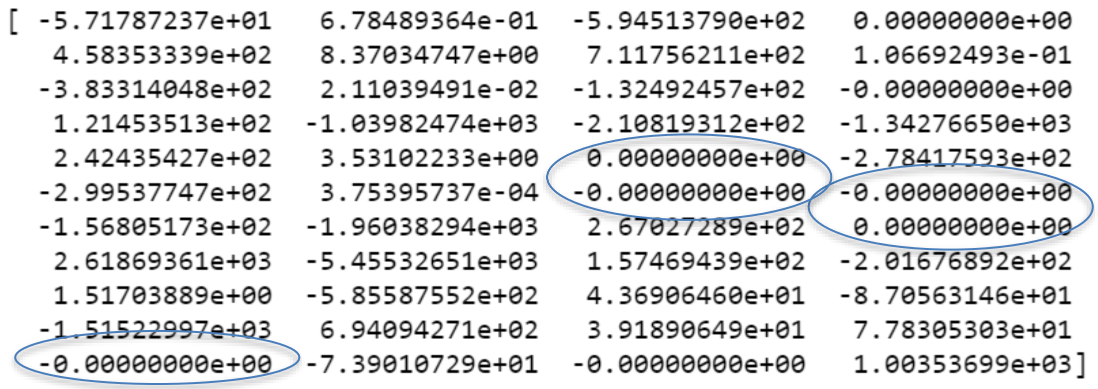
Logistic Regression also has been applied to the model and some of the coefficients for the attributes are shown below. LogisticRegression package in scikit learn has been used. Logistic Regression performs regularization to avoid over fitting. The model adds L2-penalty so that there is shrinkage of coefficients for the attributes. The same can be observed in the screenshot below. We can see that the magnitude of coefficients is diminished in contrast to OLS regression. The R squared value obtained is slightly better than that obtained for OLS Regression. But one disadvantage with Logistic Regression is that the computational time required for building the model is very high. It nearly took two hours for the model to build.



```
[ -5.38673306e-02   2.14140341e-04   4.14427078e-02 ...,  -2.65026992e-02
  -6.16072009e-04  -6.76798970e-04]
[ -9.84404007e-04  -5.77690982e-05  -7.44985618e-03 ...,  -5.26106480e-02
  -4.56127429e-05   5.80238228e-02]
[ -1.30921054e-02  -1.96864227e-03  -1.89604815e-03 ...,   1.44919488e-03
   8.51347829e-05  -6.38208895e-04]
```

❑ Lasso Regression

To avoid the overfitting discussed above, Lasso Regression is performed. Lasso (Least Absolute Shrinkage and Selection Operator) is a regression method which automatically performs regularization to avoid overfitting and performs feature subset selection to eliminate unwanted features. Feature subset selection is done by assigning coefficients as 0 to some features so that they do not affect the model. The coefficients Lasso Regression generates are going to be sparse. Lasso Regression tends to prefer solutions with fewer parameters effectively reducing the number of variables upon which the given solution is dependent. This is very important for us since we have many features. SkLearn's Lasso package has been used to build the model. The coefficients which are shown in the figure (next page) are sparse as discussed earlier. From the values, it can also be seen that some coefficients are zero and hence the model is independent of these features. We have also calculated R-squared for the model



```
[ -5.71787237e+01   6.78489364e-01  -5.94513790e+02   0.00000000e+00
   4.58353339e+02   8.37034747e+00   7.11756211e+02   1.06692493e-01
  -3.83314048e+02   2.11039491e-02  -1.32492457e+02  -0.00000000e+00
   1.21453513e+02  -1.03982474e+03  -2.10819312e+02  -1.34276650e+03
   2.42435427e+02   3.53102233e+00   0.00000000e+00  -2.78417593e+02
  -2.99537747e+02   3.75395737e-04  -0.00000000e+00  -0.00000000e+00
  -1.56805173e+02  -1.96038294e+03   2.67027289e+02   0.00000000e+00
   2.61869361e+03  -5.45532651e+03   1.57469439e+02  -2.01676892e+02
   1.51703889e+00  -5.85587552e+02   4.36906460e+01  -8.70563146e+01
  -1.51522997e+03   6.94094271e+02   3.91890649e+01   7.78305303e+01
  -0.00000000e+00  -7.39010729e+01  -0.00000000e+00   1.00353699e+03]
```

Regression Summary

As mentioned earlier, we have applied OLS, Logistic and Lasso Regressions to our data set. The results are tabulated and are shown below. R-squared value is slightly better for Logistic Regression. But overall, we have found that the R-squared value is not close to 1 for any of the models. Therefore, Regression task on the data set is not giving us satisfactory results. In the next section we have explored Classification task to identify if it is giving good results.

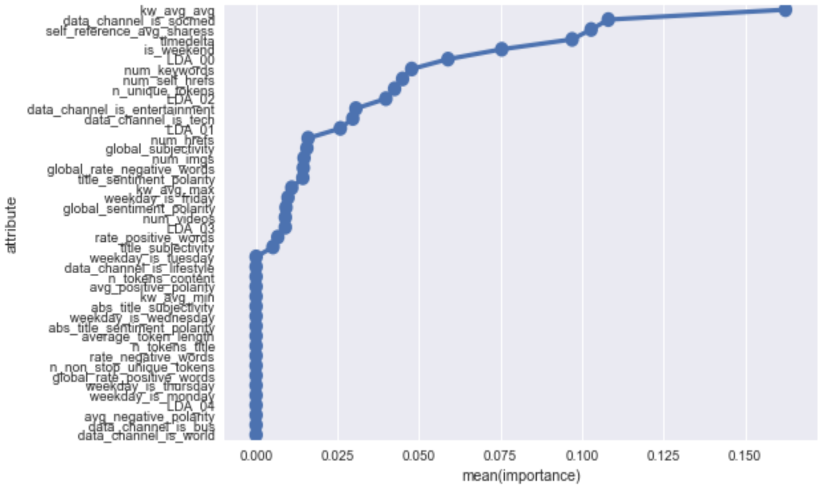
	OLS Regression (Linear)	**Logistic Regression**	**Lasso Regression**
R squared	0.0135553723885	0.0576127535062	0.0147415631154
Time for Model Building	Very Less (order of milli secs)	Very High (2 hours)	Very Less (order of milli secs)

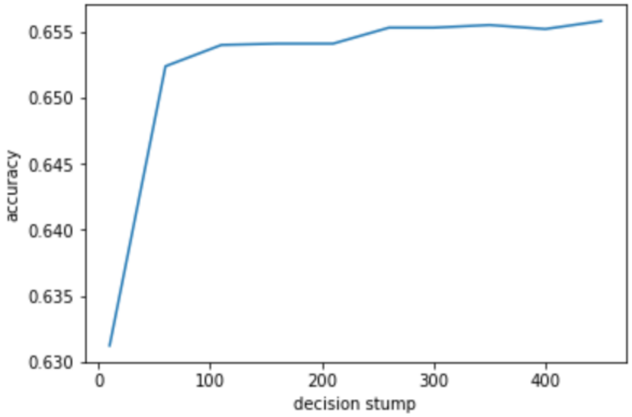
2.Classification

Classification of the given data set is performed by categorizing the news into two categories, i.e Popular and Unpopular based on the number of shares. Since the distribution for shares is right skewed, the median of the shares attribute is taken as a threshold for classification. All the news articles having shares greater than 1400 are considered as Popular news and those having shares lesser than 1400 are considered as Unpopular news. Several classification algorithms have been applied on the data set and the following section summarizes the same.

❑ ADA BOOST

To perform classification of the dataset, Adaboost Classifier is chosen with decision stump (decision tree with 1 level) as the base estimator. The AdaBoost Classifier internally ranks the features by calculating feature importances based on Gini Index. The following graph shows the importances calculated by AdaBoost Model for each feature.

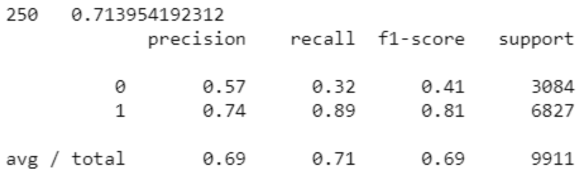




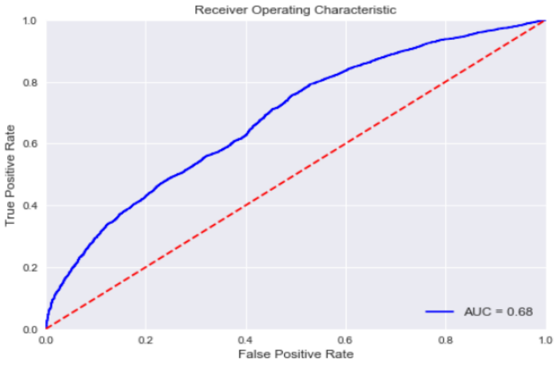
By varying the decision stumps for the model, we have observed that the accuracy for the model gets stabilized around 250 decision stumps giving the best performance.

By plugging in these values as parameters for the model, we have achieved comparatively high values for overall accuracy, precision, recall and f-measure

Shown below is the confusion matrix and ROC curve with the above mentioned optimal values taken as parameters for the AdaBoost model.

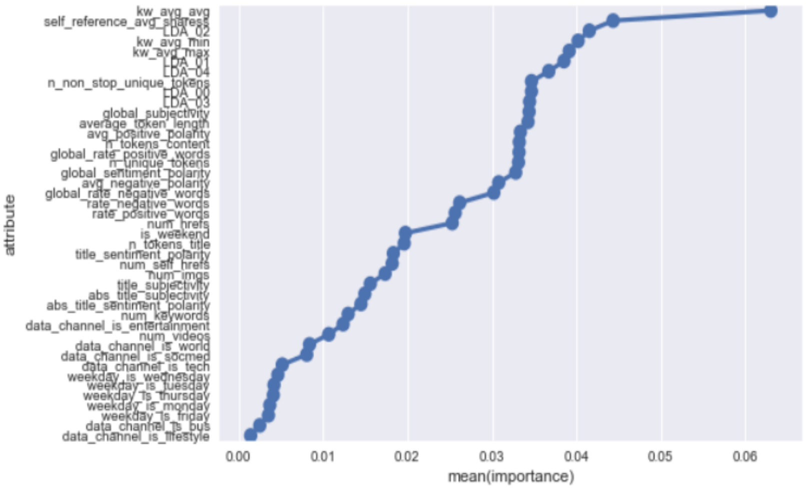


```
250   0.713954192312
             precision    recall  f1-score   support

          0       0.57      0.32      0.41      3084
          1       0.74      0.89      0.81      6827

avg / total       0.69      0.71      0.69      9911
```

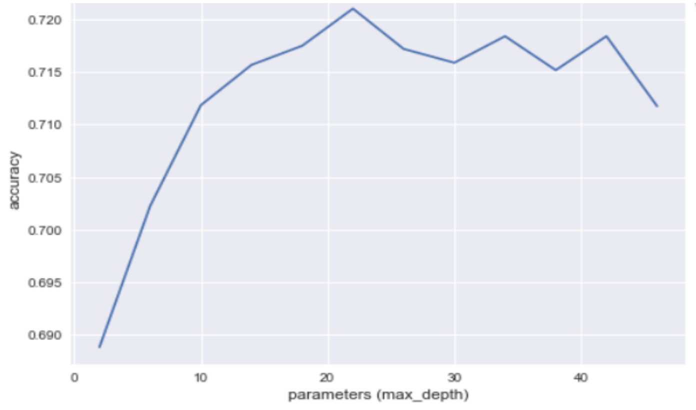


❑ Random Forest

A Random Forest, which is an ensemble of several decision trees is built on the data set. By using a number several decision trees, overfitting is controlled allowing the performance metrics to be improved. The sub-samples of the given data set are fed into each decision tree. The size of the input fed into each decision tree is kept constant by setting the parameter bootstrap = True. So, the size of each sub sample is the same and sampling is done with replacement. RandomForestClassifier package from scikit learn has been used for building the model. We have also calculated the feature importance based on mean decrease in impurity (Gini Impurity) at the node and this is averaged over all trees in the ensemble. The following graph plots the feature importances for each feature as calculated by the Random Forest Model.

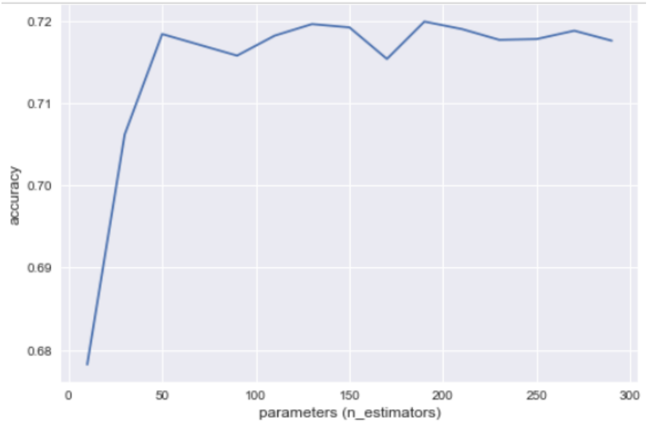


We have varied two parameters max-depth and n_estimators for the Random Forest model to see how the performance metrics are changing. Max_depth denotes the maximum depth of each decision tree used in the ensemble and n_estimators denote the number of decision trees in the ensemble.



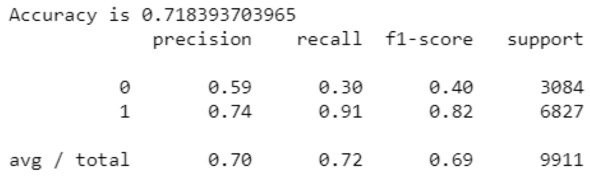
From the graph, when max_depth is increased beyond 22, no significant increase in accuracy has been observed. So, we have chosen 22 as the optimal depth for each tree in the ensemble.

The graph on right shows the relationship between number of decision trees in the ensemble and the accuracy. It can be seen from the graph that when n_estimators is increased beyond 50, the accuracy has not improved significantly. So, we have chosen 50 as the optimal value for the number of trees in the ensemble.

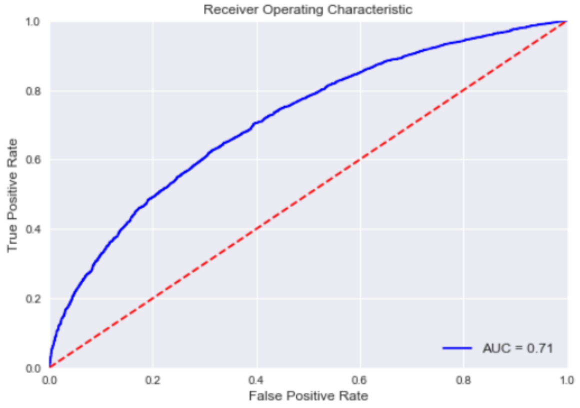


By plugging in these values as parameters for the model, we have achieved comparatively high values for overall accuracy, precision, recall and f-measure

Shown below is the confusion matrix and ROC curve with the above mentioned optimal values taken as parameters for the RandomForest model.



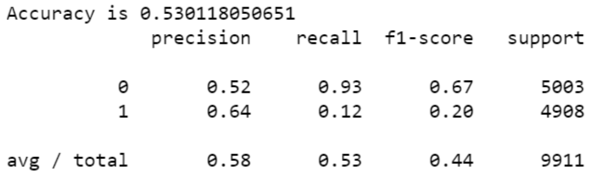
```
Accuracy is 0.718393703965
                  precision    recall  f1-score   support

              0       0.59      0.30      0.40      3084
              1       0.74      0.91      0.82      6827

    avg / total       0.70      0.72      0.69      9911
```



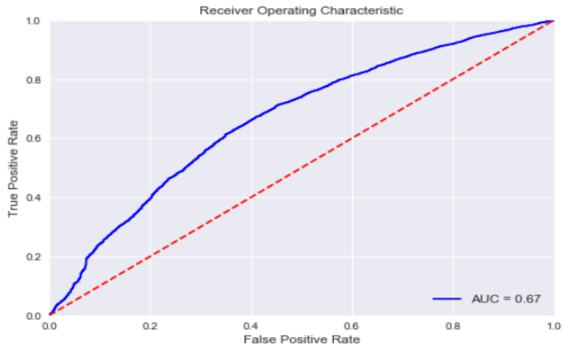
❑ Naive Bayes

Since our data set about news articles includes results from text mining and sentiment analysis, we have tried to create a Naive Bayes Classifier for our data set. The confusion matrix is shown below. The observed accuracy and f-score are not that satisfactory. So, we have concluded that the Naive Bayes Model does not fit our data set as expected. The failure can be attributed to the naive assumption about class conditional independence. It is possible that the features in our data set are not independent.

The confusion matrix and ROC curve with the above mentioned optimal values taken as parameters for the AdaBoost model are calculated.

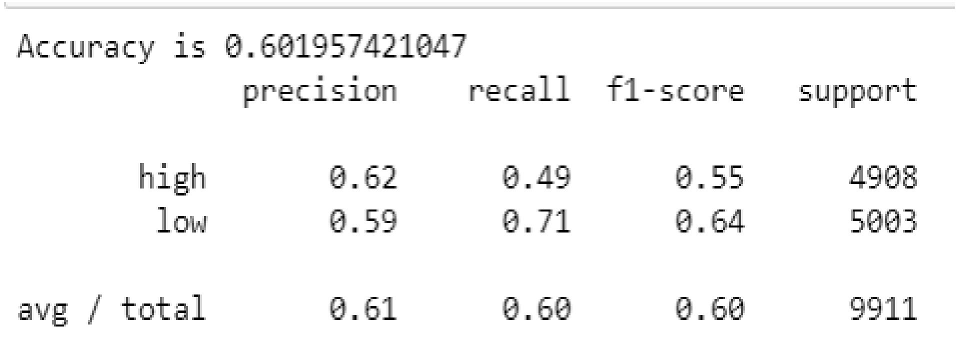


```
Accuracy is 0.530118050651
                  precision    recall  f1-score   support

              0       0.52      0.93      0.67      5003
              1       0.64      0.12      0.20      4908

    avg / total       0.58      0.53      0.44      9911
```



❑ Support Vector Machine

Support Vector Machines provide us with some major advantages while dealing with classification problems where data set has high dimensions. Hence, we have decided to try it for our data set. It provides a regularization parameter which prevents the model from overfitting the data. By engineering the kernel, we will be able to build expert knowledge about the problem. The algorithm also uses certain instances in the training data as support vectors to identify the decision function. We have used SVC (Support Vector Classifier) package from scikit learn. We have chosen the kernel function to be a polynomial of degree 3. The model is validated on the test data and the confusion matrix is given below.



```
Accuracy is 0.601957421047
             precision    recall  f1-score   support

       high       0.62      0.49      0.55      4908
        low       0.59      0.71      0.64      5003

avg / total       0.61      0.60      0.60      9911
```

From the confusion matrix we can see that the performance metrics on the test data are reasonably moderate. One major drawback of using Support Vector Machine with our data is that the time for building the model is very high (more than 3 hours). So because of this, we were not able to vary the parameters for the model (kernel function, degree) to see how the performance metrics are changing.

Classification Summary

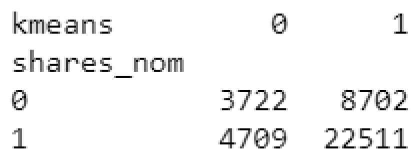
We have applied four classification models AdaBoost, Random Forest, Naive Bayes and Support Vector Machine to the data set. From our observation the ensemble classifiers (AdaBoost and Random Forest) have comparatively better performance than the remaining two. The reason for the failure of Naive Bayes method could be because of naive assumption regarding class conditional independence. Support Vector Machine has moderate performance, but the time needed for model building is very high.

3. Clustering

To view how well the data can be clustered, Clustering algorithms are applied on the dataset. After discretizing the ‘shares’ attribute to popular and unpopular we have experimented with identifying whether the given data set can be divided into two clusters. K-Means Clustering and Hierarchical Clustering have been applied on the data set and the following sections summarize the same.

❑ K-MEANS

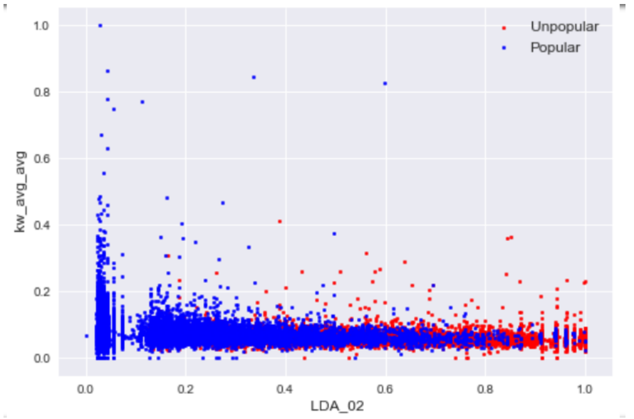
KMeans package from scikit learn has been used to create clusters for the data. Since we want to categorize the news articles into Popular/Unpopular we chose k=2 for the model. New labels have been given for the data based on the clusters. The cross-tabulation matrix showing true clusters and assigned clusters is given below.

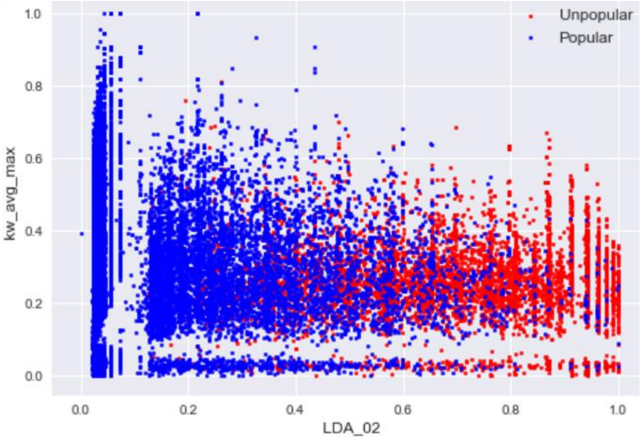


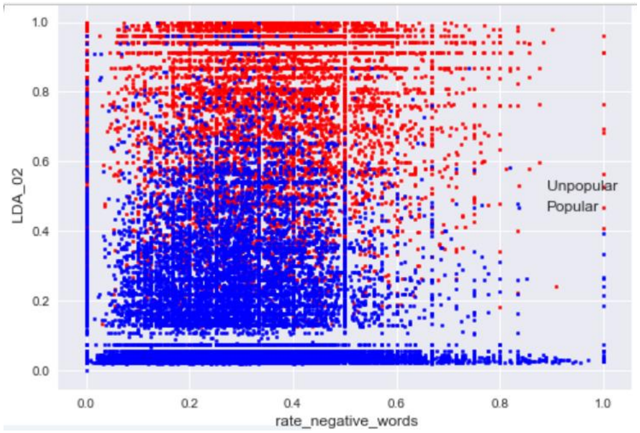
```
kmeans          0      1
shares_nom
0            3722   8702
1            4709  22511
```

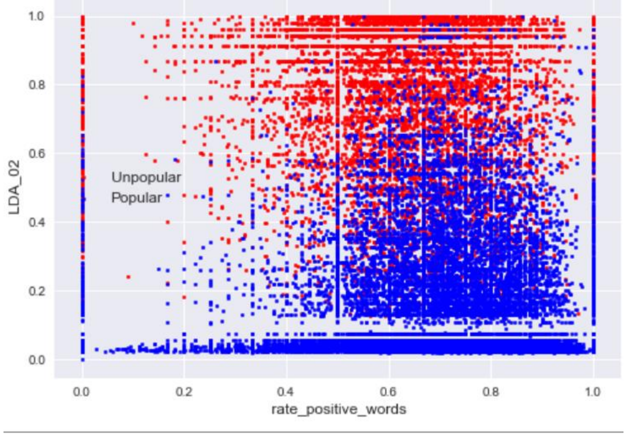
From the cross-tab we can see that most of the popular news articles are clustered into one group, but the unpopular ones got distributed between the two clusters.

After obtaining the cluster assignments, plots have been created for some pairs of attributes to view the cluster assignments. All the plots had the clusters overlapped with no separation. Few plots for reference are given as below.



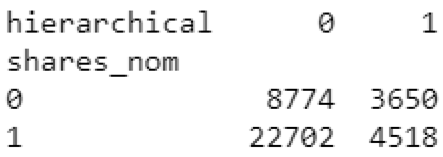






❑ Hierarchical Clustering

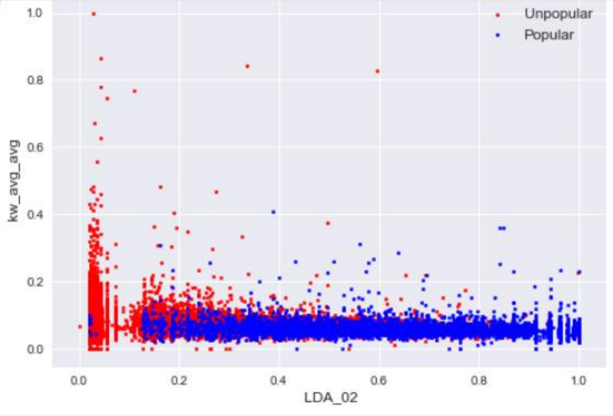
Since K means clustering did not give us satisfiable results we have decided to see how hierarchical clustering performs. AgglomerativeClustering package from scikit learn has been used to create the model. The number of clusters to keep is again chosen as two. The cross tabulation matrix showing true clusters and assigned clusters is given below.

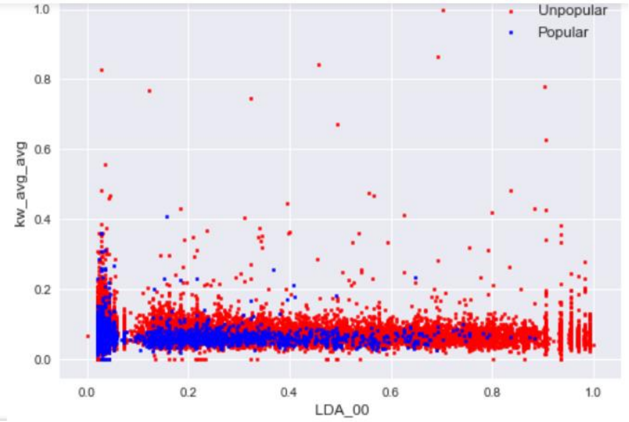


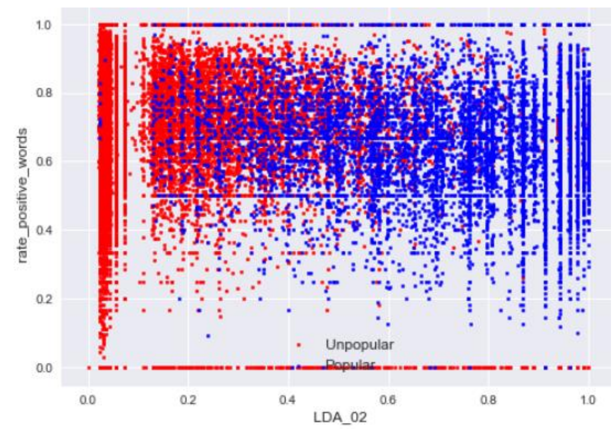
```
hierarchical        0     1
shares_nom
0                8774  3650
1               22702  4518
```

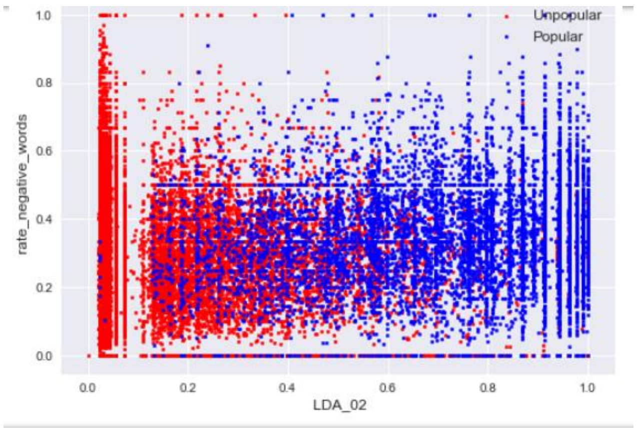
From the given cross-tab we can see that majority of the popular news are grouped into one cluster (0). Also, most of the unpopular news also got clustered into the same cluster.

After creating the cluster assignments, plots have been created for some pairs of attributes showing the cluster assignments and few plots are shown below.









From the above results we can conclude that, neither of the clustering algorithms perform exceptionally well. However, hierarchical clustering performs slightly better than the k-means clustering.

Summary

	REGRESSION	CLASSIFICATION	CLUSTERING
MODELS USED	• Ordinary Least Squares Regression • Logistic Regression • Lasso Regression	• AdaBoost • RandomForest • Naïve Bayes • SVM	• K-Means clustering • Hierarchical Clustering
BEST MODEL	Lasso Regression	RandomForest, AdaBoost	Hierarchical Clustering(slightly)
PERFORMANCE ON MODEL	Average	Average	Poor
CONS	High regression coefficients for OLS and high computational time for logistic regression.	Due to the large volume of dataset, SVM took lot of time for model building	Based on the results, the dataset doesn't seem to perform well on clustering

Individual Contributions

Pavani	Vedasri
• Correlation Analysis	• Principal Component Analysis
• Feature Subset Selection	• Data Transformation
• OLS Regression	• Data Cleaning
• Lasso Regression	• Logistic Regression
• Support Vector Machine	• Random Forest
• AdaBoost	• Support Vector Machine
• Naive Bayes	• Clustering
• Evaluation	• Evaluation