ONLINE NEWS POPULARITY ANALYSIS



Report by:

Gunjan Lalwani

Rishabh Jain

Ann Sajee

INDEX

1. **OVERVIEW** …………………………………………………………………………………………………… 3
2. **PROBLEM STATEMENT** ………………………………………………………………………………… 4
3. **METRICS** ………………………………………………………………………………………………………. 5
4. **ANALYSIS** …………………………………………………………………………………………………….. 6
   1. **DATA EXPLORATION AND VISUALIZATION** …………………………………………… 6
      1. **USING MATPLOTLIB, SEABORN, PLOTLY** ………………………………………. 6
      2. **USING TABLEAU** …………………………………………………………………………… 12
   2. **ALGORITHMS AND TECHNIQUES** …………………………………………………………. 15
5. **METHODOLOGY** …………………………………………………………………………………………..16

5.1 **DATA PREPROCESSING** …………………………………………………………………………. 16

5.2 **IMPLEMENTATION** ……………………………………………………………………………….. 20

5.2.1 **MANUALLY CREATED CLUSTERS** ……………………………………………………. 20

5.2.1 **CLUSTERS USING K-MEANS** …………………………………………………………… 21

5.3 **HYPERPARAMETER TUNING** …………………………………………………………………. 25

1. **SUMMARY** ………………………………………………………………………………………………….. 27
2. **CONCLUSION** ………………………………………………………………………………………………. 28
3. **OVERVIEW:**

With the help of Internet, the online news can be instantly spread around the world. Most of peoples now have the habit of reading and sharing news online, for instance, using social media like Twitter and Facebook. Typically, the news popularity can be indicated by the number of reads, likes or shares. For the online news stake holders such as content providers or advertisers, it's very valuable if the popularity of the news articles can be accurately predicted prior to the publication. Thus, it is interesting and meaningful to use the machine learning techniques to predict the popularity of online news articles. We have dataset with 39,643 articles from the Mashable website. In this project, based on the dataset including 39,643 news articles from website Mashable, we will try to find the best classification learning algorithm to accurately predict if a news article will become popular or not prior to publication.

1. **PROBLEM STATEMENT:**

In this project, we have used machine learning techniques to solve a binary classification problem, which is to predict if an online news article will become popular or not prior to publication. The popularity is characterized by the number of shares. If the number of shares is higher than a pre-defined threshold, the article is labeled as popular, otherwise it is labeled as unpopular. Thus, the problem is to utilize a list of article’s features and find the best machine learning model to accurately classify the target label (popular/unpopular) of the articles to be published. As the problem can be formulated as a binary classification problem, we have implemented and compared three classification learning algorithms including Logistic Regression, RF, and Adaboost. The best model is selected based on the metrics introduced in next part.

1. **METRICS:**

As a classification task, we have calculated the following three evaluation metrics: accuracy,

F1-score and AUC. For all three metrics, the higher value of the metric means the better

performance of model.

(a) Accuracy: Accuracy is direct indication of the proportion of correct classification.

It considers both true positives and true negatives with equal weight and it can be computed as

accuracy = (true positives + true negatives)/ dataset size

Although the measure of accuracy might be naive when the data class distribution is highly skewed, but it is still an intuitive indication of model's performance.

(b) F1-score: F1-score is an unweighted measure for accuracy by taking harmonic mean of

precision and recall, which can be computed as

F1 = (2 \*precision\*recall)/ (precision + recall)

It is a robust measurement since it is independent of data class distribution.

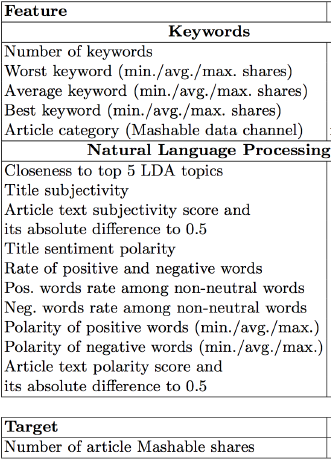
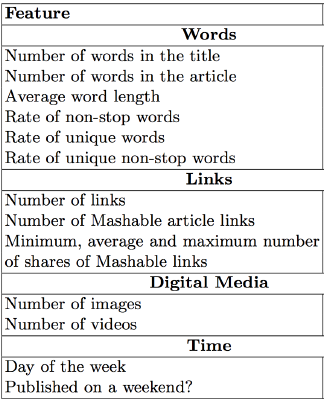
(c) AUC: The AUC is the area under the ROC (Receiver Operating Characteristics) curve, which is a plot of the True Positive Rate versus the False Positive Rate. AUC value is a good measure of classifier's discrimination power and it is a more robust measure for model performance.

1. **ANALYSIS**
   1. **DATA EXPLORATION AND VISUALIZATION**

**4.1.1 USING MATPLOTLIB, SEABORN, PLOTLY**

The dataset is consisted of 39,643 news articles from an online news website called Mashable collected over 2 years from Jan. 2013 to Jan. 2015. It is downloaded from UCI Machine Learning Repository as <https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity> and this dataset is generously denoted by the author. For each instance of the dataset, it has 61 attributes which includes 1 target attribute (number of shares), 2 non-predictive features (URL of the article and Days between the article publication and the dataset acquisition) and 58 predictive features as shown in Fig. below.

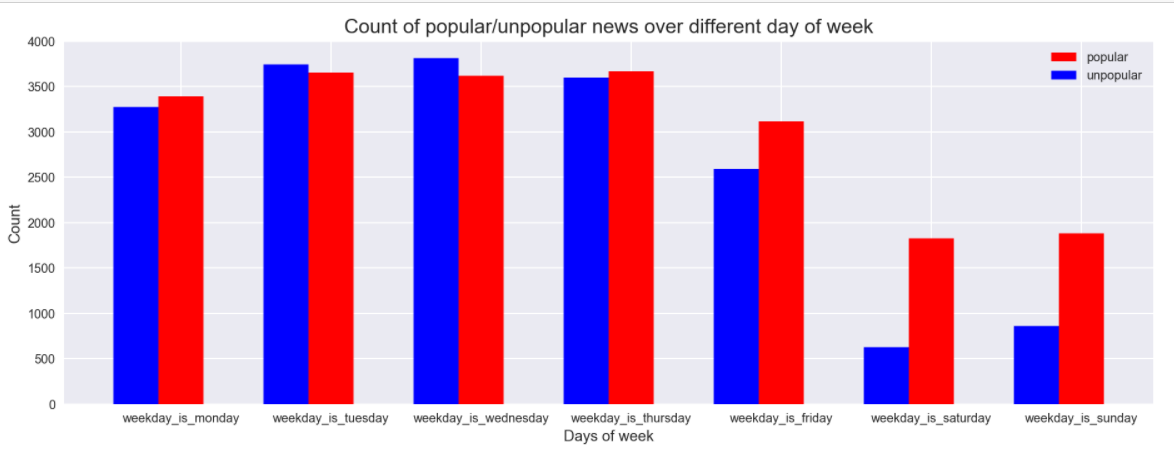
The dataset has already been initially preprocessed. For examples, the categorical features like the published day of the week and article category have been transformed by one-hot encoding scheme, and the skewed feature like number of words in the article has been log- transformed.



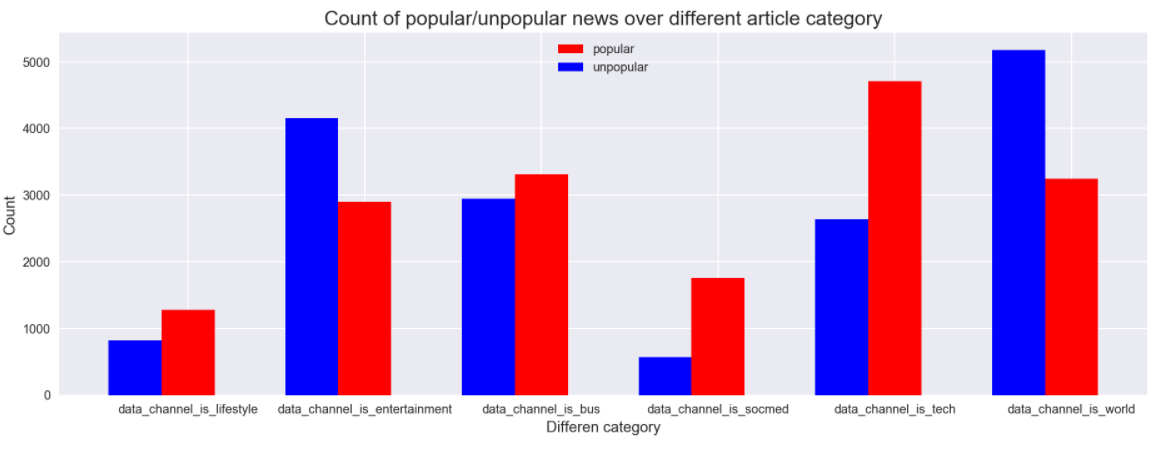
Firstly, we determined the appropriate threshold for number of shares to discriminate the news to be popular or unpopular. So, we show the statistics of the target attribute "shares" in below fig., and we find the median of target attribute is 1,400, thus it is reasonable if we take 1,400 as a threshold. Then we used this threshold to convert the continuous number target attribute into a Boolean label.



By observing various features, I think there are several relevant features like day of the week and article category. In Fig. 3, the count of popular/unpopular news over different day of the week is plotted. We can clearly find that the articles published over the weekends has larger potential to be popular. It makes sense because it is very likely that people will spend more time online browsing the news over the weekends. In Fig below, the count of popular/unpopular news over different article category is plotted. We can observer that in category of technology ("data channel is tech") and social media ("data channel is socmed"),the proportion of popular news is much larger the unpopular ones, and in category of world ("data channel is world") and entertainment ("data channel is entertainment"), the proportion of unpopular news is larger the popular ones. This might reflect that the readers of Mashable prefer the channel of technology and social media much over the channel of world and entertainment.



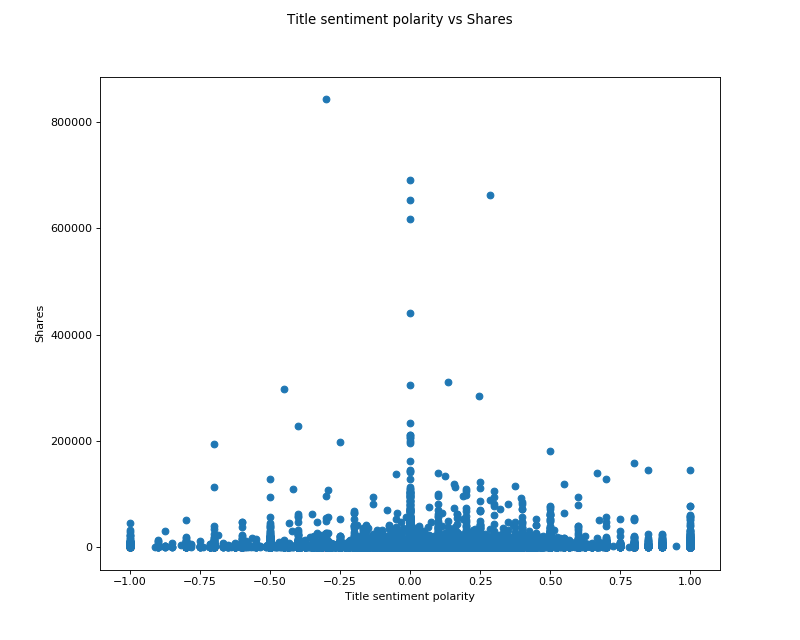
Days of the week vs Shares for popular and unpopular articles



Category for popular and unpopular articles

**Title Sentiment Polarity:**

How does the title of an article affect its shares? What sentiments should be expressed in title?



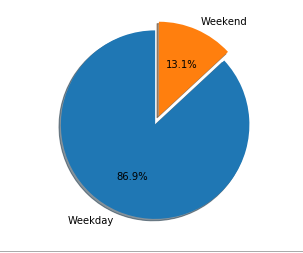
Title sentiment Polarity vs Shares

**RESULT:**

Mostly the articles have titles which are not too positive or negative. It lies within the range of -0.5 to 0.5. However highest concentration can be seen in the 0 axis i.e. high no. of articles is neutral in nature.

**Weekday vs Weekend:**

When are most articles published? Is it more on weekdays or weekends?



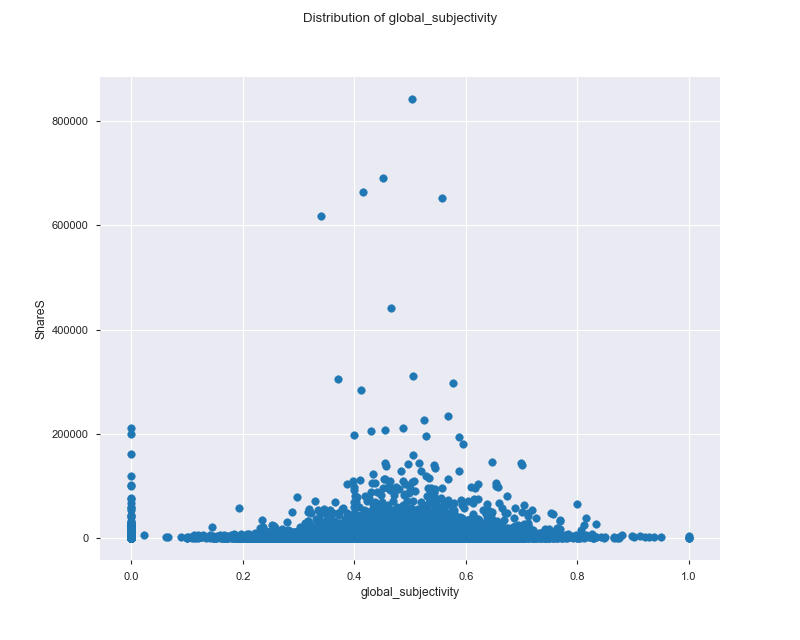
Weekends vs Weekday Pie chart

**RESULT:**

The above graph shows that most of the articles are published in Weekdays. However, in order to get more popularity on must publish the article on weekends.

**Global subjectivity:**

Understanding the global subjectivity of an article is extremely important when it is being read by millions of people across the globe. Will it appeal to masses?



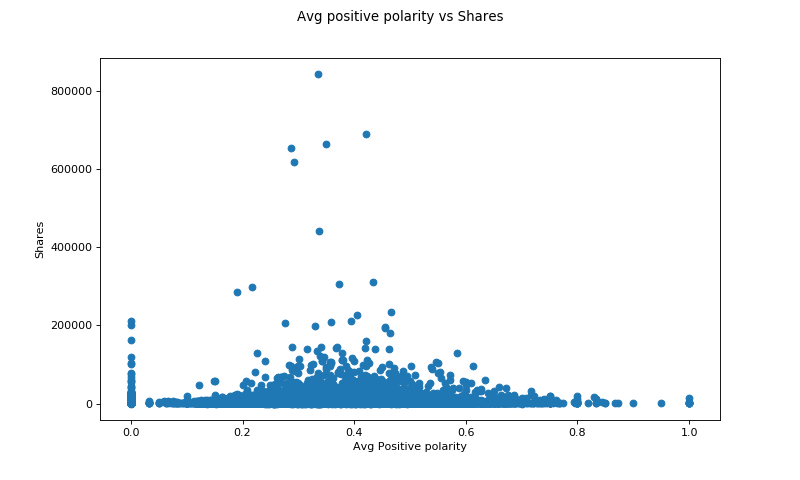
Distribution of Global subjectivity

**RESULT:**

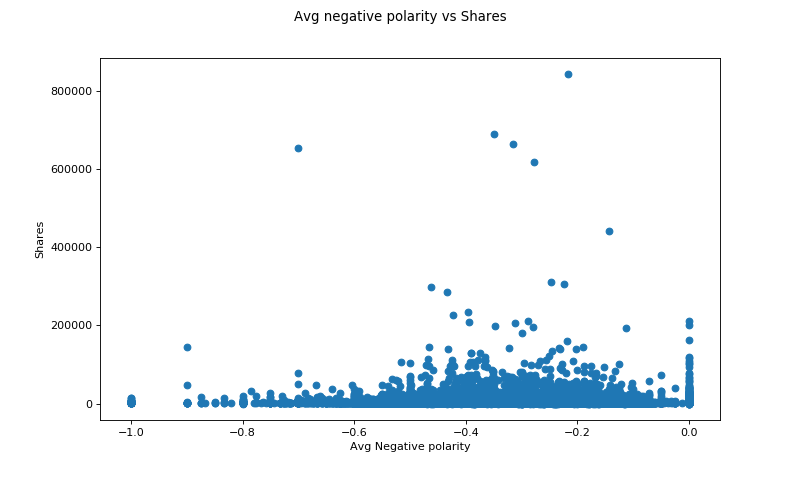
Maximum of global subjectivity lies from 0.2 to 0.8. A significant outlier lies at global subjectivity of 0.503 with share of 957. Hence, we conclude that most of the articles with medium global subjectivity are maximum shares.

**Average Polarity:**

How is the content of the article? Is it highly positive or negative?



Average Positive Polarity



Average Negative polarity

**RESULT:**

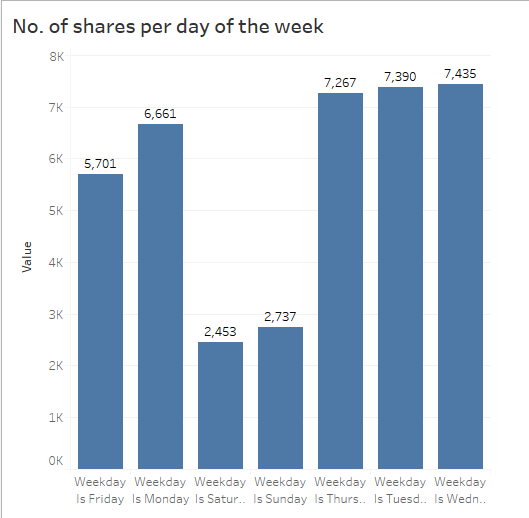
Most of the shared articles are slightly to medium positive or mostly neutral in nature.

* + 1. **USING TABLEAU**

Tableau is a great data visualizing tool using which one can explore data and get useful insights from it. We have visualized the dataset in Tableau as well and found useful insights from the dataset.

1. **Number of shares per day of the week:**

On which days are most of the articles shared?



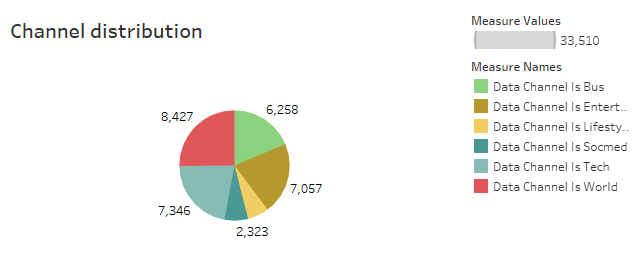
Number of shares per day of the week

**RESULT:**

Most of the articles are shared on Weekdays rather than weekends. However, after our analysis, we have found that publishing an article on weekends will result in more shares as more people read online news during weekends.

1. **Channel distribution:**

How is the channel distribution amongst shares? Which channel is most popular?



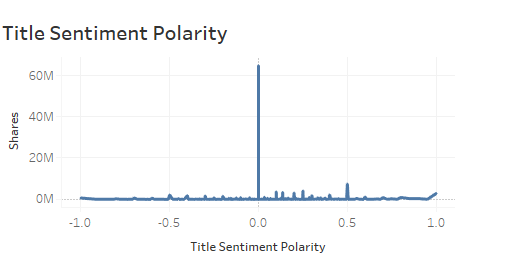
Channel distribution

**RESULT:**

We can conclude that most of the articles published are related to world news

1. **Title sentiment polarity:**

How is the polarity of title? Is it positive, negative or neutral?

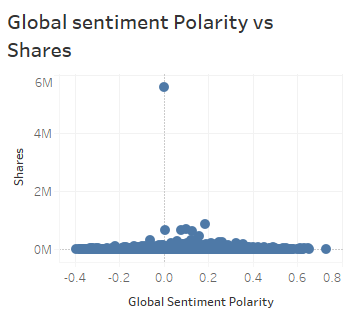
****

**RESULT:**

Most of the articles have their title as neutral and rest positive in nature.

1. **Global sentiment Polarity:**

Is the article appealing to the masses across the globe?

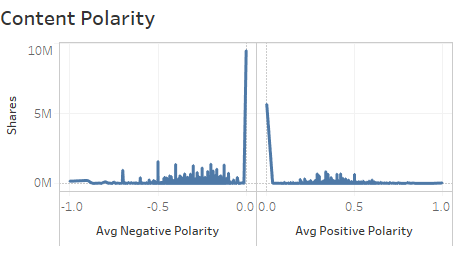
****

**RESULT:**

Most of the articles lie in the range of -0.2 to 0.3. That means it slightly negative to neutral to slightly positive.

1. **Content polarity:**

How is the content of the article? Is it positive, negative or neutral?

****

**RESULT:**

Most of the articles have their content neutral.

**4.2 ALGORITHMS AND TECHNIQUES**

Since we formulate this problem as a binary classification problem.

In this project, three classification learning algorithms including Logistic Regression, RF,

and Adaboost, Decision Tree, Support Vector Machine, etc are implemented and compared based on the evaluation metric such as accuracy, F1-score and Area Under ROC Curve (AUC). All the learning algorithm are implemented by sklearn toolbox.

1. **METHODOLOGY**
   1. **DATA PREPROCESSING**

As mentioned above, some data preprocessing works have been done by the data's

donator. The categorical features like the published day of the week and article category

have been transformed by one-hot encoding scheme, and the skewed feature like number of

words in the article has been log-transformed. Based on this, I further preprocess the dataset

by normalizing the numerical feature to the interval [0; 1] such that each feature is treated

equally when applying supervised learning. I also select the median of target attribute as

The threshold to convert the continuous target attribute to Boolean label.

Since there are 58 features in the dataset, it is reasonable to conduct a feature selection to

reduce the data noise and increase the algorithm running speed. One effective way is using

recursive feature elimination with cross validation (RFECV) to automatically select the most

Significant features for certain classifier. Sklearn provides a function called REFCV() that

can help us.

Firstly, we run RFECV with a logistic regression estimator. The cross validation score versus the number of feature selected. From the figure, we can find there is drop of score when number of features is 29. Thus RFECV algorithm selects 29 most relevant features from 58 original features. The selected 29 features are listed below figure.

Interestingly, the day of week and article category features are included in these 29 features

as we discussed and visualized in above section.

**Logistic Regression:**

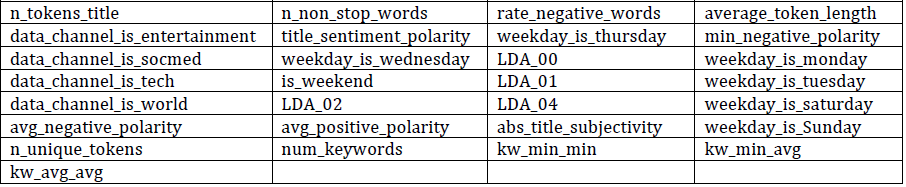
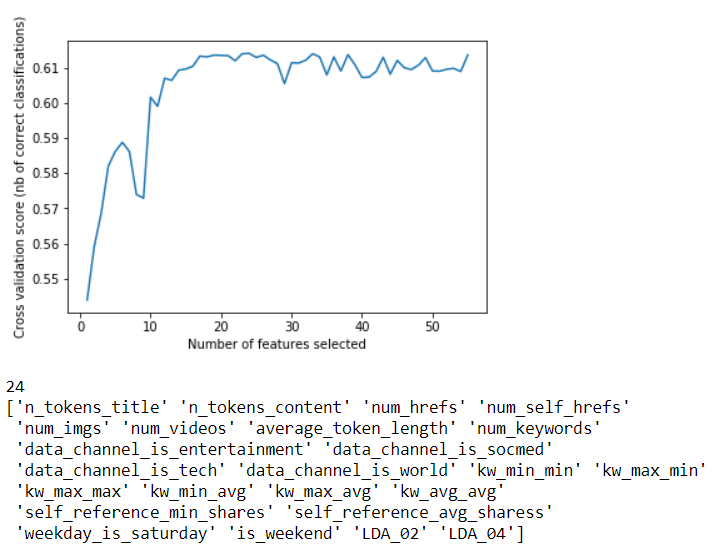


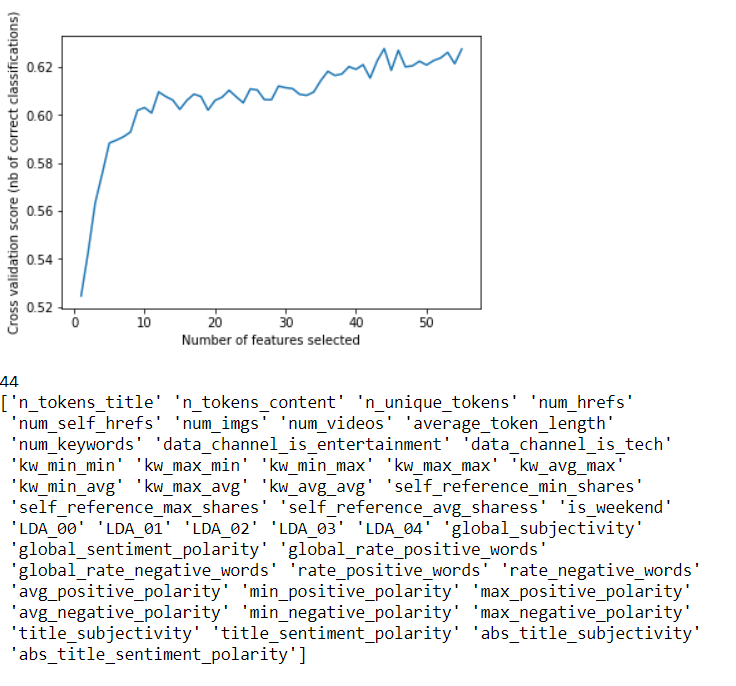
Figure: 29 features selected using RFECV with logistic regression estimator.



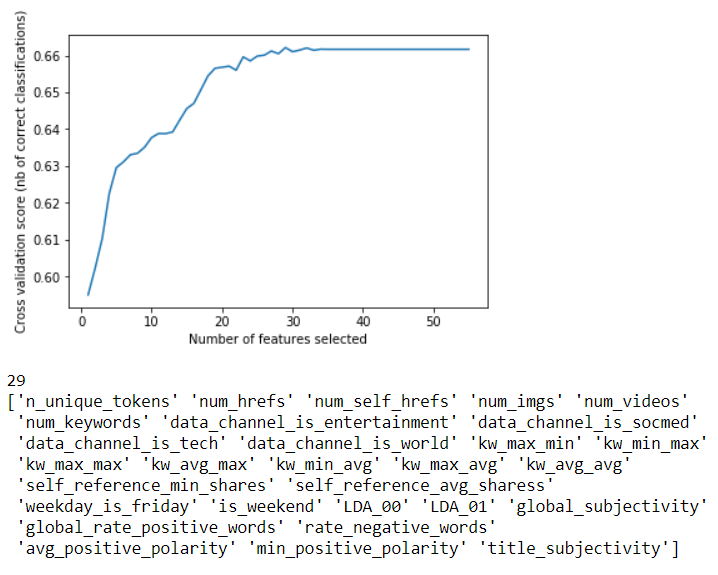
**Random Forest:**

Next, we run RFECV with a RF estimator. The cross-validation score versus the number of feature selected is shown in below fig. We find that RFECV selects 56 features for RF, which

is almost the full features in dataset. The only two features RFECV excludes for RF is ['n nonstop words', ' data channel is lifestyle'].



**Adaptive Boosting:**



* 1. **IMPLEMENTATION**

**5.2.1 MANUALLY CREATED CLUSTERS**

Before algorithm implementation, for each algorithm, I also randomly split dataset with its

own selected features into training set (90%) and testing set (10%). The logistic regression,

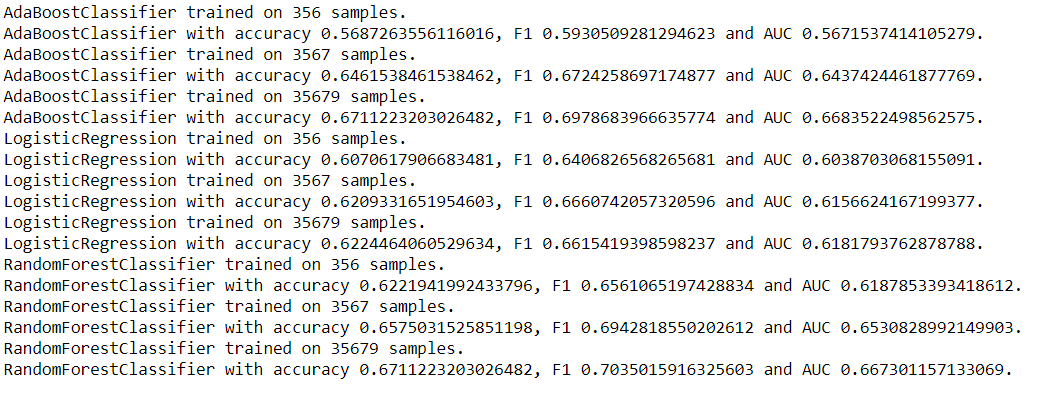
RF and Adaboost are implemented by the sklearn function LogisticRegression(), Random-

ForestClassifier() and AdaBoostClassifier(), respectively. At this stage, I have used the

default setting for model hyperparameters.



**Evaluation Metrics:**



* + 1. **CLUSTERS CREATED USING K-MEANS**

**Cluster Description-**

We got 5 clusters with the help of K-Means

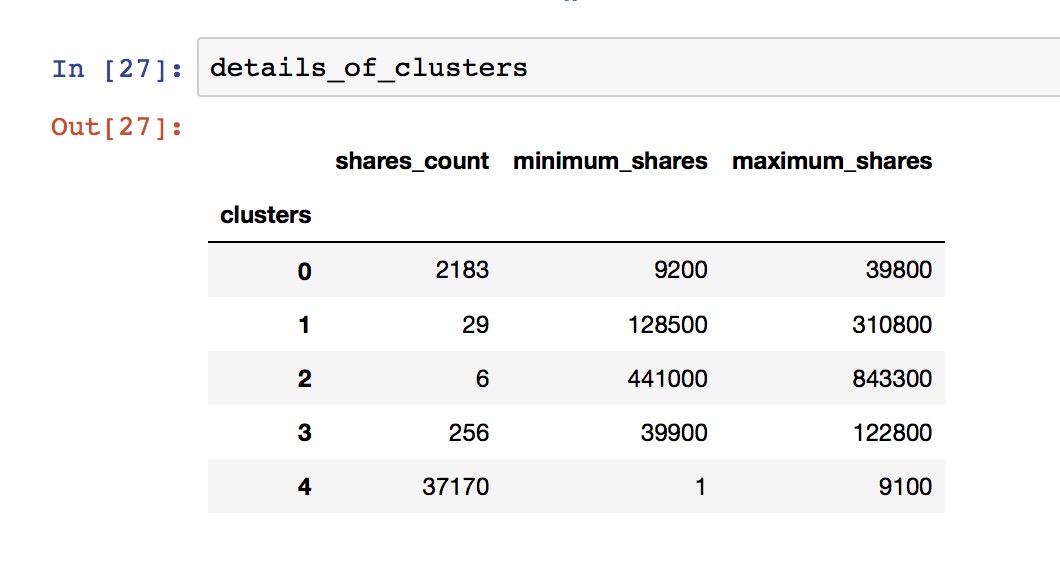
**Obscure-** Obscure Cluster contains of minimum shares

**Mediocre-** Mediocre consist of shares higher than obscure but lower than Popular

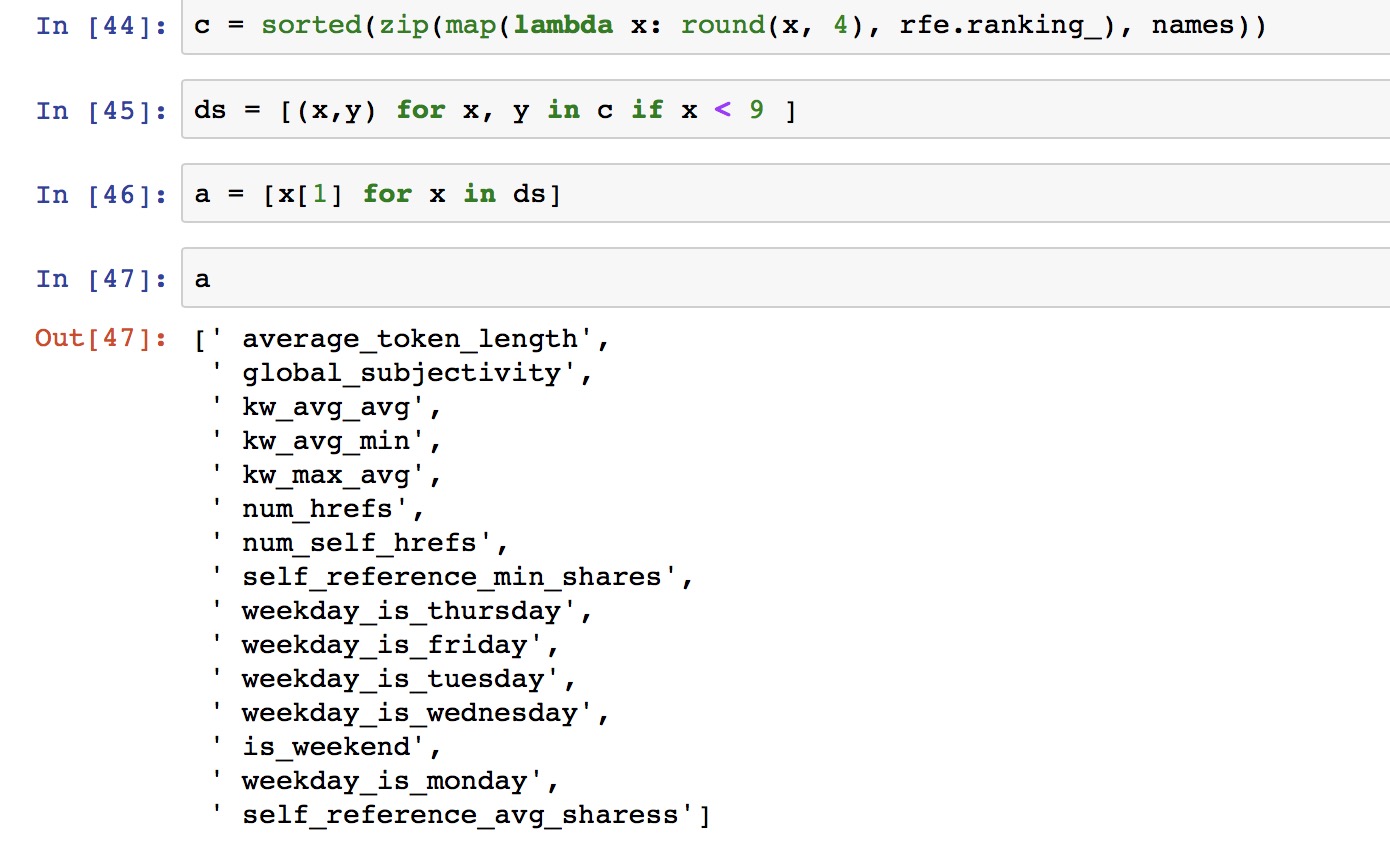
**Popular-** It consists of shares that are popular.

**Super Popular-** They are less popular than viral

**Viral-** Viral cluster shows shares of online news which were viral

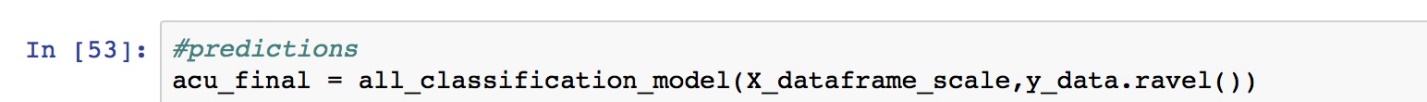


Best Feature Selected Using RFE



**Model Training & Testing-**

We have used a single function for training and testing the models which calls each model recursively. The Precision, Recall, F1-Score, Accuracy are calculated with the help of classification report





**Accuracy:** Accuracy = TP+TN/TP+FP+FN+TN

**Precision:** The question that this Precision metric answer is of all passengers that labeled as survived, how many actually survived?

Precision = TP/TP+FP

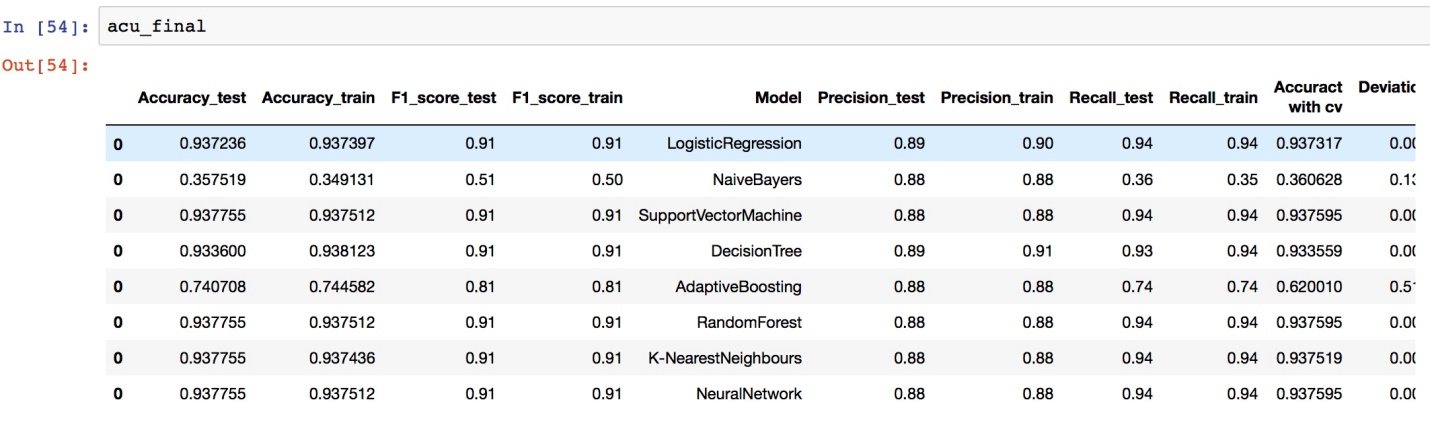
**Recall:** The question recall answer is: Of all the passengers that truly survived, how many did we label?

Recall = TP/TP+FN

**F1 score:** Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different

F1 Score = 2\*(Recall \* Precision) / (Recall + Precision)





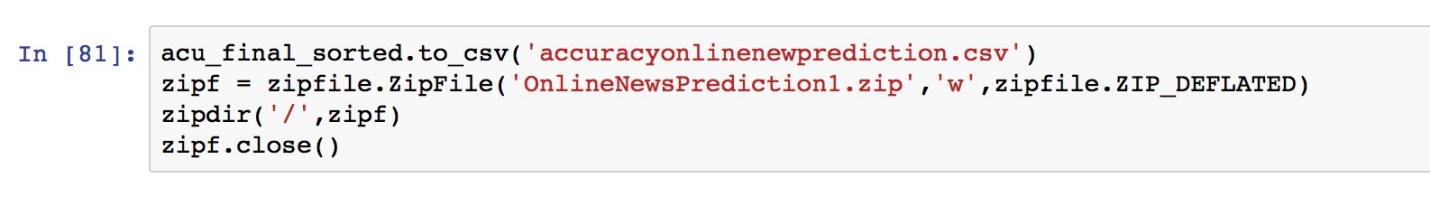
**Ranking Of Models-**



**Pickled Models-**



**Pickled File Stored in Zip Format**



**5.3 HYPER-PARAMETERS TUNING**

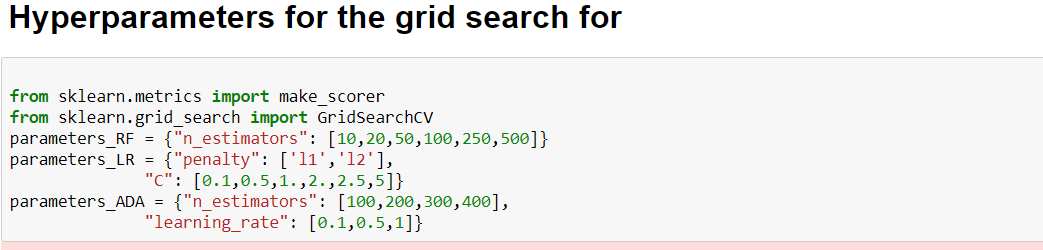
In this part, I have used the grid search method for each of the three classifiers to tune their hyperparameters. The grid search method exhaustively search through all possible

combinations of model parameters, cross validate the model and then determine which set

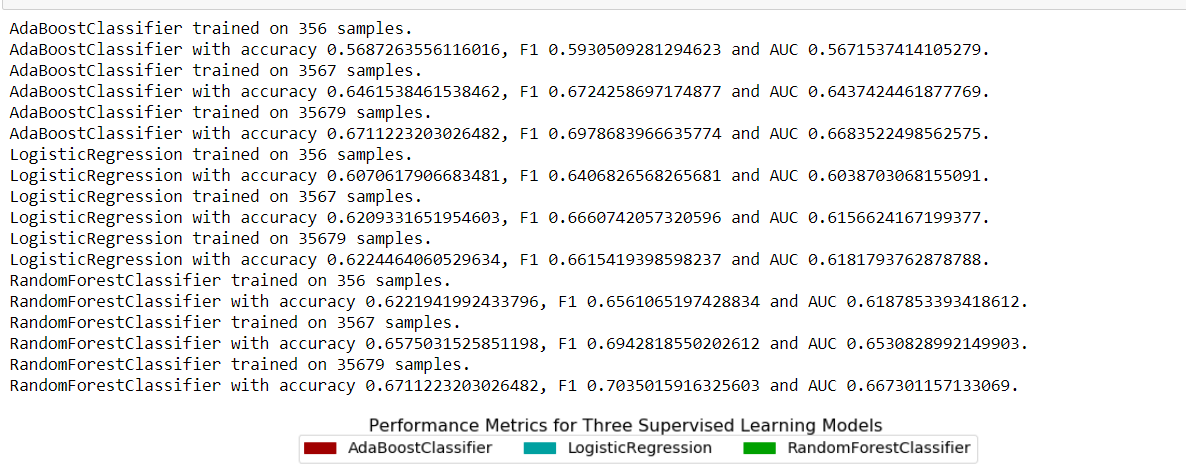
of model parameters gives the best performance. Since the grid search can help select an

optimal model parameter, thus the learning algorithm can be optimized. In sklearn, we can

use the function GridSearchCV() to implement grid search.







Compared with the metrics in fir above, we can find the metrics of logistic regression and Adaboost are just slightly improved, but the metrics of random forest are almost same after tuning by grid search.

1. **SUMMARY**

To summarize, the project is conducted through following steps:

(a) **Data collection:** The dataset of some 40,000 online news articles are downloaded from

UCI Machine Learning Repository, which is originally collected and donated by the

author .

(b) **Data preprocessing**: Based on the initial data preprocessing in original dataset, I fur-

ther preprocess the dataset by normalizing the numerical feature such that each feature

is treated equally when applying supervised learning. I also select the median of the

target attribute (number of shares) as an appropriate threshold to label all the data

as either popular or unpopular.

(c) **Data exploration and visualization**: I explore the relevance of certain feature by visu-

alization. And I also visualize the data distribution by PCA.

(d) **Feature selection:** To select the most relevant features among all 58 features, I use

RFECV to select the most relevant features for each of the classifier.

(e) **Classifier implementation and hyperparamter tuning:** The classification algorithms including Logistic Regression, RF and Adaboost ,Support Vector Machine, Neural Networks, Decision Tree are implemented using sklearn. Then the model's hyperparameters are tuned by grid search method.

(f) **Model evaluation and validation**: The refined models are evaluated and compared using

three metrics (accuracy, F1-score, AUC).

1. **CONCLUSION:**

We have used **2 Approaches** for Online News Popularity-

1. **Manually Created Cluster-**

Popular, Unpopular

1. **Creation of Cluster with the help of K-Means-**

Obscure, Mediocre, Popular, Super Popular, Viral

With the help of manually created clusters we got the best Model as RandomForestClassifier

In K-Means Cluster we got the best Model as RandomForestClassifier:

F1-score = 0.91

Precision = 0.88

Recall as 0.94