Online News Popularity Analysis using Machine Learning Techniques

IST 718 Big Data Analytics Project Report,
Syracuse University

Group 7

Juilee Salunkhe
Akshay Bhala
Yeswanth Reddy
Sai Praharsha Devalla

TABLE OF CONTENTS

Project Abstract

To predict whether an article is going to be a hit or not before publishing and predicting the number of shares an article will be shared are our two aims in the project. Random Forest, Gradient Boosting, Logistic Regression, and others are the most common mining algorithms used for classification. In this research, we aimed to find the best model to predict the popularity of online news, using machine-learning techniques, and implement various data mining algorithms tuning hyperparameters. The data source was Mashable, a well-known online news website. AUC Score, Accuracy, Precision, Recall, and F-measure were used to evaluate the results and their results were compared to find the better one. It is treated as a classification problem for predicting the popularity and as a regression problem to predict the number of shares. The project intends to implement three classification models and evaluate the best performing model by measuring performance based on various metrics.

.

1. Introduction

1.1 Project Overview

Digital media marketing is the use of various online platforms to connect with your audience to build your brand, increase sales, and drive website traffic. How they do it?
Well, there are various techniques used such as targeted advertising, influencer marketing, content creation, and optimization. In recent years, digital marketers, bloggers, and businesses of all sizes have been busy creating content of all types to engage their target audience. Whether it's in the form of informative blog posts, customer testimonial videos, or recorded webinars, content is everywhere online. It's very important to know and optimize the content to target the right audience. Machine Learning tools are immensely helpful in analyzing what type of content, keywords, and phrases are most relevant to your desired audience.
The project is a case of a Content optimization technique that uses machine learning algorithms to predict what articles are likely going to be a hit. Typically, the news popularity can be indicated by the number of reads, likes, or shares. For the online news stakeholders such as content providers, it's very valuable if the popularity of the news articles can be accurately predicted before the publication. It is a binary classification problem which helps us in predicting whether an article is going to be popular or not and to find the best classification learning algorithm to accurately predict if a news article will become popular or not before publication. Also, data is treated as a regression problem to predict the number of shares an article can get. The project intends to help Mashable decide which articles should they publish since they can predict which articles will be having the maximum number of shares and their popularity and how to increase the popularity. Classification models like Logistic Regression, Gradient Boosting, Random Forest will be implemented and compared. The models will be tuned, and the best model will be selected based on the metrics used.

1.2 About the data

Data is produced by 'Mashable' where they collected data of around 39000 articles. There are about precisely 39644 observations and 61 variables. The Mashable dataset consists of articles data information mainly as a number of unique words, number of non-stop words, the postpositive polarity of words, negative polarity of words, and so on. For each instance of the dataset, it has 61 attributes which include 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.1. The dataset has already been initially preprocessed. For example, the categorical features like the published day of the week and article category have been transformed by a one-hot encoding scheme, and the skewed feature like the number of words in the article has been log-transformed.
The project intends to transform it into a classification problem by setting a decision threshold on shares attributes. Number of images, videos published on an article, different categories of channels like Lifestyle, Entertainment, etc. where articles are published, the day of publication, URLs, Positive and negative polarity of the words in publication, all these attributes contribute to the variation and characteristics of the article. These attributes

can help us to find answers to some of the questions like, “On what week-day what type of article should Mashable post more?”. For different categories of articles what should be their min and max-content length?” and so on.

Dataset Predictors Overview:

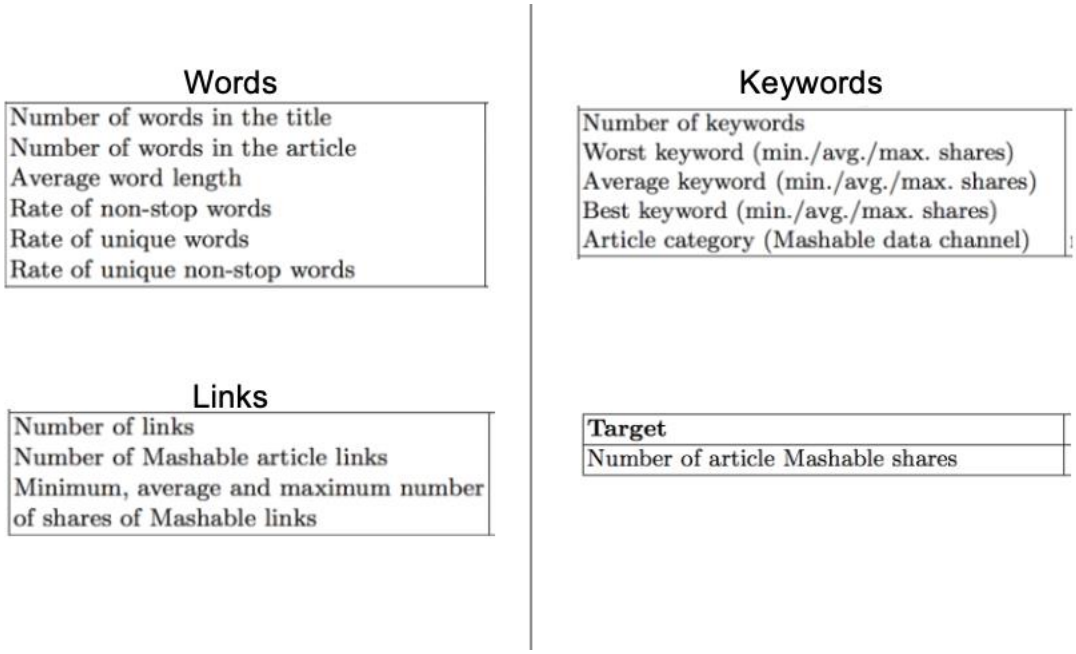

Fig.1. List of Predictive attributes

Others include Digital Media variables like Number of Videos, Images, Time related like Day of the week and whether it is been published on weekend and natural language processing related variables like Title subjectivity, sentiment polarity, rate of positive and negative words, Polarity of positive and negative words (min, max, average) and so on.
Will try to answer some business questions like what features contribute best in making a content good, how it can be achieved, etc.

2. Methodology

2.1 Data Pre-processing

Data Pre-processing started with checking for noisy, inconsistent entries in the dataset. 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 a one-hot encoding scheme, and the skewed feature like the number of words in the article has been log-transformed. After exploring the data for missing values, no missing values were found. Also, there were no duplicate values detected since each article is unique and the number of observations equals to the number of articles.

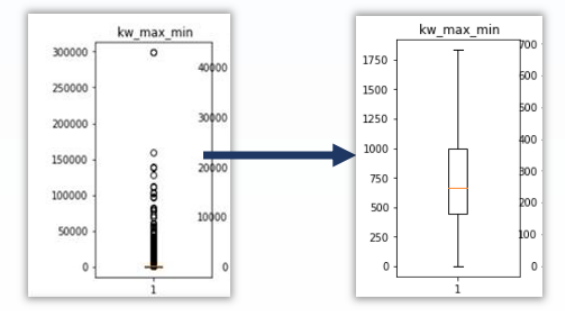

Fig.2. Outlier detection by box-plot

Data was checked for any outliers. Wherever applicable outliers can be excluded to eliminate noisy data from the dataset. If not removed they can alter the accuracy of models resulting in poorer outputs. With the use of boxplots, the distribution of data was visualized. Outliers are those values of a variable that fall far from the central point, the median. Outliers were detected using IQR (Inter-Quartile Range). Outlier Detection was done by identifying the lower-bound and upper-bound of the data. Any values which were less than lower-bound or greater than upper bound were deleted.

```
count     39644.000000
mean       3395.380184
std       11626.950749
min           1.000000
25%         946.000000
50%        1400.000000
75%        2800.000000
max      843300.000000
Name: shares, dtype: float64
```

Fig.3. Summary Statistics of target variable 'shares'

Descriptive statistics were done to determine the appropriate threshold for the number of shares to discriminate the news to be popular or unpopular. The statistics of the target attribute "shares" in Fig.2. showed the median of the target attribute, which is 1,400, thus it is reasonable to take 1,400 as a threshold. This threshold is used to convert the continuous number target attribute into a boolean label. Further, the data was pre-processed by normalizing the numerical feature to the interval [0,1] such that each feature is treated equally when applying supervised learning.

2.2. Exploratory Data Analysis

Data Pre-processing was followed by Exploratory Data Analysis. From the plot, we can

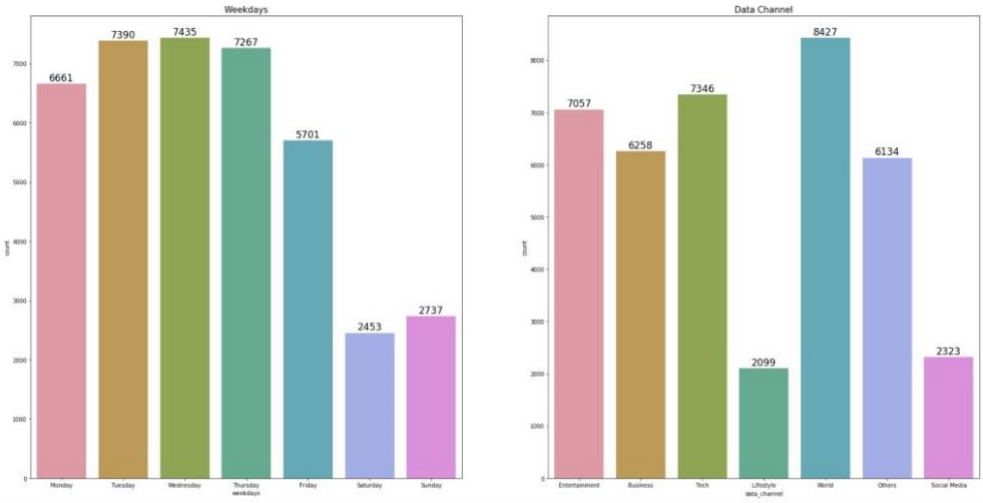

Fig.4. Distribution of Weekdays and Data Channel

observe, the weekdays have the highest number of shares. Wednesday, Tuesday, Thursday, and Monday have the highest number of shares, with Friday having a significant drop. We can also see which category or topic does the best. Here is a bar chart for the seven

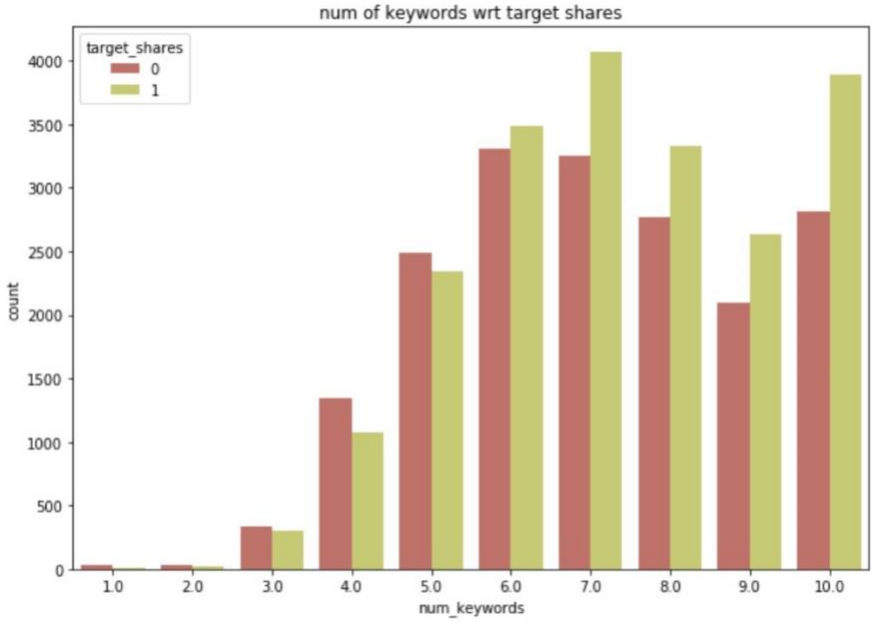

Fig.5. Number of keywords v/s Target shares

categories Tech, Entertainment, World, Business, Social Media, Lifestyle, and Others. The best performing category is World, followed by Tech, Entertainment, Business. The least popular categories are Social Media and Lifestyle.

Looking at the number of keywords in an article we can observe an increasing trend i.e., the number of shares increases when there are more keywords in an article.

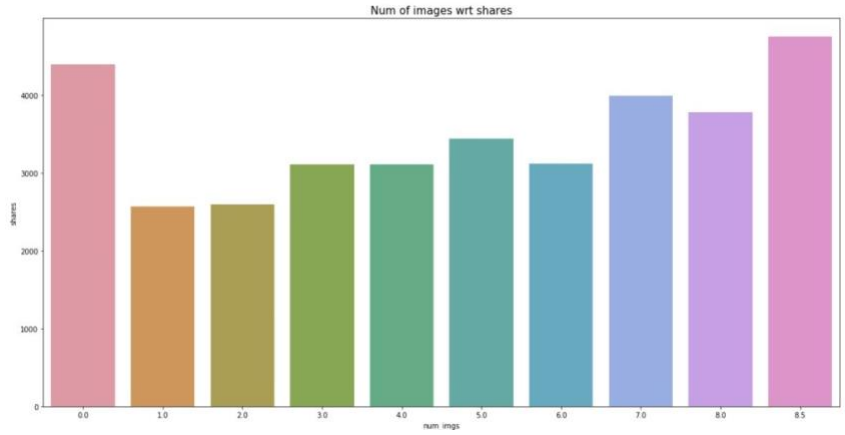

Fig.6. Number of Images v/s shares

This chart here shows the relationship between the number of images and the num of times the article is shared. The popularity seems to increase when there are a significant number of images in the article.

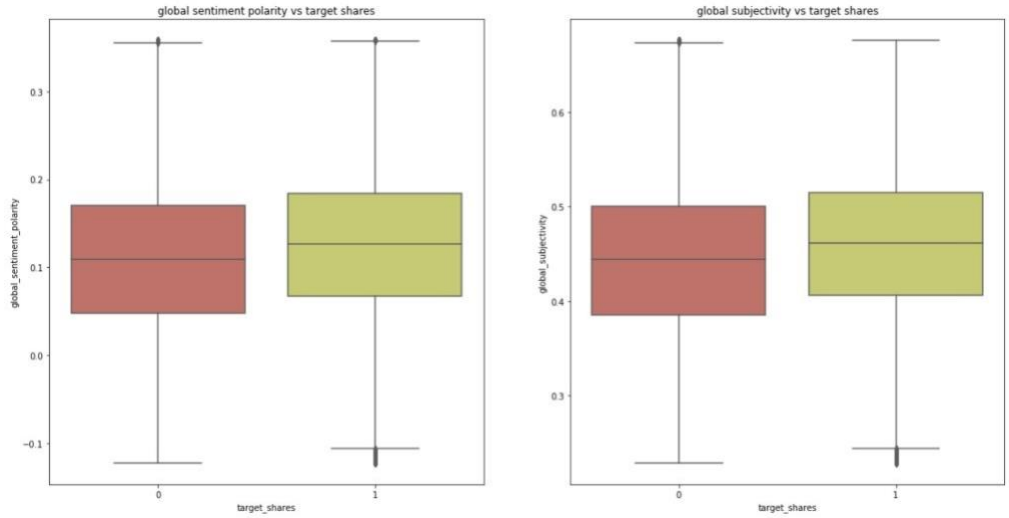

Fig.7. Global sentiment polarity and subjectivity v/s shares

Polarity in sentiment analysis refers to identifying sentiment orientation (positive, neutral, and negative) in written or spoken language. Other types of sentiment analysis include fine-grained sentiment analysis which provides more precision in the level of polarity (e.g. very positive, positive, neutral, negative, and very negative) and emotion analysis which aims to identify emotions in expressions (e.g. happiness, sadness, frustration, surprise). Language can contain expressions that are objective or subjective. Objective expressions are facts. Subjective expressions are opinions that describe people's feelings towards a specific subject or topic. Here global sentiment Polarity is calculated by using sentiment analysis on the news articles. Subjectivity refers to the quality of being based on or influenced by personal feelings, tastes, or opinions. The articles with more subjectivity and polarity tend to be more popular among the online news.

2.3. Data Modeling and Evaluation

Machine learning is the process of mathematical algorithms learning patterns or trends on previously recorded data observations and then makes a prediction or classification.

2.3.1. Classification: In this work, we are examining only binary classification (e.g. Y = 1,0), which is a form of supervised learning in which an algorithm aims to classify which category an input belongs to. Supervised learning can be described as taking an input vector comprised of n-features and mapping it to an associated target value or class label. Models were developed using different classification algorithms and then selecting the one(s) that present better performance indicators. Because the label "shares" could only assume binary values (0: no; 1: yes), the following two-class classification algorithms were chosen:

1. Logistic Regression
2. Random Forest Classifier
3. Gradient Boosting

1. Logistic Regression

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Logistic regression predicts whether something is True or False, instead of predicting something continuous. Also, instead of fitting a line to the data, logistic regression fits an "S" shaped logistic function (sigmoid). The curve goes from zero to one and in our example, Logistic regression uses a sigmoid function to classify whether the share would be popular (1) or not (0). Logistic regression doesn't have the concept of residual, so it can't use least squares and can't calculate R square. Instead, it uses Maximum Likelihood. Logistic Regression picks up a probability, scaled by features of a popular share, and then use that to calculate the likelihood of observing a non-popular share. Then we calculate the likelihood of all the shares. Lastly, we multiply all of those likelihoods together. Then we shift the curve and calculate the new likelihood of the data and we continue to shift the curve until we get the curve with maximum likelihood. We started by running the base model of logistic regression keeping the default values and got the results as shown in the image below in fig.8.

The model performed with an accuracy of 66%. Performance metric chosen were AUC score and accuracy. To improve AUC score and accuracy, hyperparameters are tuned. The Hyperparameters in logistic regression are regularization parameters and elastic net parameter. As we know Regularization adds penalties to more complex models and then sorts potential models from least overfit to greatest; The model with the lowest "overfitting" score is usually the best choice for predictive power. The two types of penalties are L1 (absolute value of the magnitude of coefficients) and L2 (square of the magnitude of coefficients). Elastic Net is an alpha parameter that is somewhere in between L1 and L2. Lambda is a term that controls the learning rate. If lambda value is too high, the model will be simple but can run the risk of underfitting and if lambda value is too low, the model will be more complex and can run the risk of overfitting the data as well. The best hyperparameters we got for our model are shown below:

AUC Score	Accuracy	Precision	Recall	F1 Score
0.709	0.66	0.65	0.66	0.65

regParam	elasticNetParam
0.1	0.01

AUC Score	Accuracy	Precision	Recall	F1 Score
0.709	0.66	0.65	0.66	0.65

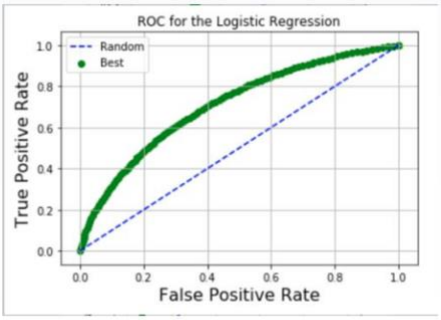

Fig.8. Logistic Regression Hyperparameter Tuning and ROC Curve

After tuning the model using grid-search our accuracy and remained constant with accuracy as 66% and AUC score as 0.709. As seen in fig.8., the Roc curve shows the curve away from the dotted lines, which proves our model to be a good fit.

2. Random Forest Classifier

Random Forest classifier is an ensemble method that combines the predictions made by multiple decision trees and each tree is generated using a selected number of input features. The decision trees with bagging form a special case of random forest which gives the randomness to the model compared to bagging by randomly selecting the samples to build the individual tree with replacement. As the random forest is nothing but a group of decision trees, it uses the splitting criteria as Gini Index or Entropy. The decision tree works in a way that entropy is maximum at the root node and will become zero at the leaf node. The node splitting will be based on a feature that has the highest Information Gain. Information Gain can be defined as Entropy before splitting and weighted average of entropy after splitting. To reduce the correlation between trees, the random forest selects P input features randomly to split at each node of the decision tree. As a result, instead of using all the available features, the decision tree will split a node based on these P features. In the random forest, the tree can grow to the fullest without any pruning. This may help reduce the bias present in the resulting tree. Once the trees have been constructed, the predictions are combined using a majority voting scheme. Ideal number of features that should be considered while splitting a node for Random forest can vary from log2(input features) to sqrt (input features). The number of features required for each decision tree to split(featureSubsetStrategy), max_depth, and the number of decision trees are the hyperparameters that can be tuned. The trees parameter specifies the number of trees in the forest of the model. The lesser the number of trees there might be over-fitting due to high variance. So a higher number of trees ensures the random forest classifier does not overfit the data. max_depth is the how many splits deep you want each tree to go. Even though high value for this reduces the bias but gives the high variance for that tree. The number of features(featureSubsetStrategy) tells each tree how many features to check when looking for the best split to make. This will not allow the tree to fit the data closely. To ensure the model is not over-fitting (i.e. high variance and low bias) number of trees should be more and the number of features should be optimal value as mentioned in the above discussion. There should be a proper trade-off between variance and bias to build a low variance and

low bias model. The base model was implemented with default hyperparameters values and got an accuracy of 68.9%. So, after trying different combinations, we had to use 150 trees and 8 features to get the best results of accuracy 73% and recall of 73% as well.

AUC Score	Accuracy	Precision	Recall	F1 Score
0.689	0.66	0.66	0.65	0.66

maxDepth	numTrees	featureSubsetStrategy
9	150	auto

AUC Score	Accuracy	Precision	Recall	F1 Score
0.725	0.73	0.73	0.73	0.72

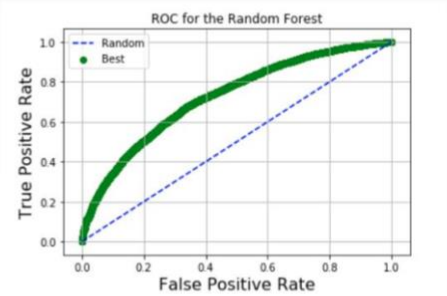

Fig.9. Random Forest Hyperparameter Tuning and ROC Curve

Using Random Forest Feature importance was performed. It helps to make sense of the features and their importance. Looking at the graph below, we see that some features are not used at all, while some impact the performance greatly like the Average number of keywords, whether an article that belongs to entertainment, tech and the world, whether the article is published on weekend, whether an article has less self-referencing share links.

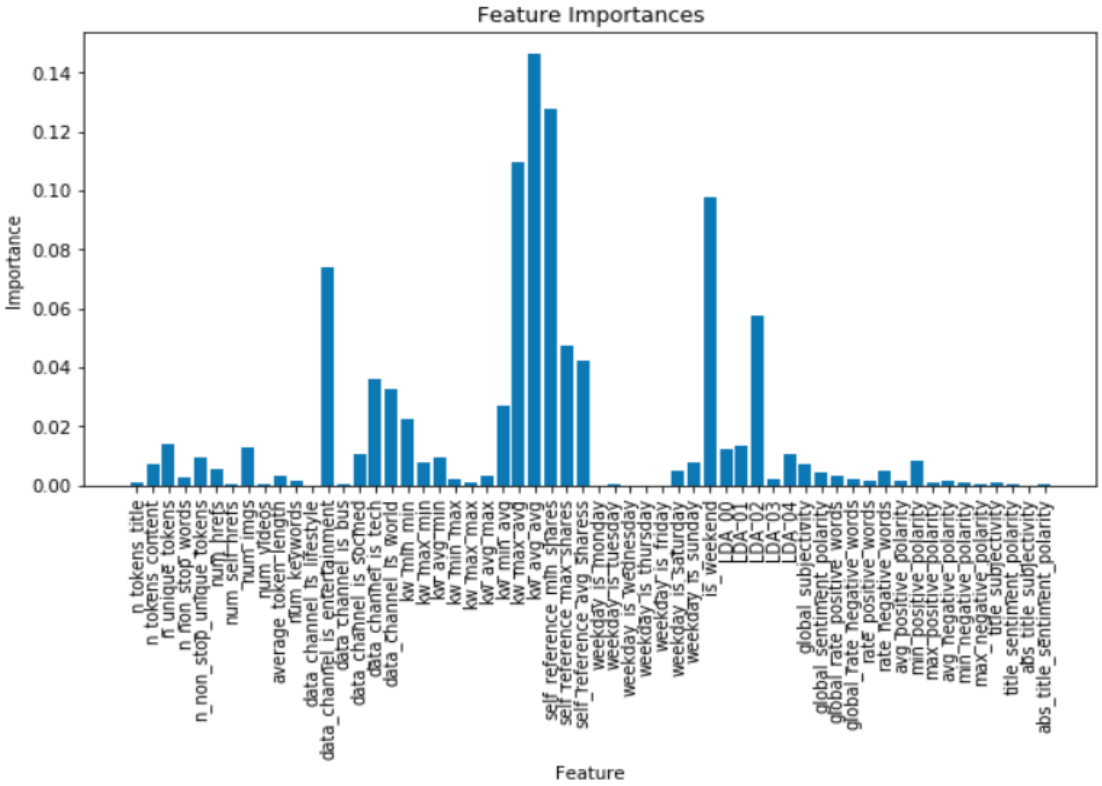

Fig.10. Feature Importance by Random Forest

3. Gradient Boosting

Gradient Boosting Machine is an ensemble method consisting of multiple decision trees where each decision tree is built sequentially one after the other and the final ensemble model is produced by taking the weighted average of predictions made by each base classifier. GBTs iteratively train decision trees to minimize a loss function. Like decision trees, GBTs handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and can capture non-linearities and feature interactions. On each iteration, the algorithm uses the current ensemble to predict the label of each training instance and then compares the prediction with the true label. The dataset is re-labeled to put more emphasis on training instances with poor predictions. Thus, in the next iteration, the decision tree will help correct for previous mistakes. The specific mechanism for re-labeling instances is defined by a loss function. With each iteration, GBTs further reduce this loss function on the training data. GBT uses Log Loss for classification Tasks. We begin running the base model with default values and got the results as shown in fig.11.

AUC Score	Accuracy	Precision	Recall	F1 Score
0.707	0.64	0.64	0.64	0.64

maxDepth	minInstancesPerNode	featureSubsetStrategy	stepSize
5	3	All	0.1

AUC Score	Accuracy	Precision	Recall	F1 Score
0.709	0.65	0.65	0.65	0.65

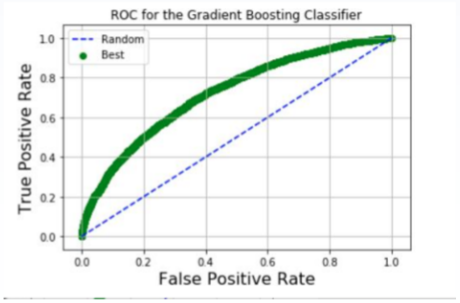

Fig.11. Gradient Boosting Hyperparameter Tuning and ROC Curve

The model was tuned to improve Accuracy and AUC score considering hyperparameters such as maxDepth, minInstancesPerNode, featureSubsetStrategy, and stepSize. While tuning the hyperparameters we observed Gradient boosting can overfit when trained with more trees. Max depth explains to us that the deeper the tree, the more the splits and it captures more information which leads to overfitting. minInstancesPerNode is the minimum number of samples required to split an internal node. It underfits when we consider all of the samples at each node. Step size also known as learning rate, Higher values of the learning rate for the model tend to overfit (high variance). The learning rate with lower values will not allow the tree to fit the data closely. featureSubsetStrategy is the number of features to use as candidates for splitting at each tree node. After applying a grid search, we got the best parameters as shown in fig.11. Using the best parameters our model accuracy and AUC score remained constant at accuracy as 65% and AUC score as 0.709 with no signs of overfitting. To prove our model to be a good fit we plotted the roc curve as shown in fig.11.

2.3.2. Regression: The project is also treated as a regression problem wherein the intention is to predict the number of shares an article can be shared. This will be very helpful for Mashable to decide which articles should they publish because they can predict which articles will be having the maximum number of shares. Random forest regression has been used to predict the number of shares.

4. Random Forest Regressor

Random forests Regressor is also an ensemble learning method for regression, other tasks that operate by constructing a multitude of decision trees at training time and outputting mean prediction (regression) of the individual trees. Random decision forests correct for decision tree's habit of overfitting to their training set. The RF regressor also uses the same parameters as of RF classifier, the only difference is that it uses splitting criteria of RMSE and MSE. It uses the sum squared error residuals to make the decision. For our problem, the range of shares varies from 1 to 843,300 and the mean is around 3400 shares. In a tree, while splitting data, the feature which has the least sum of squared residuals of all that will become the root node. In this case, keywords have minimum SSE. So, the hyperparameters tuned are shown in fig.10. The metric used to evaluate the model was RMSE. RMSE decreased.

Hyper Parameters Tuned:

maxDepth = [1, 3, 5]

numTrees = [100, 200, 300]

featureSubsetStrategy = ['log2', 'auto']

Fig.11. Random Forest Regressor Hyperparameters

RMSE
13463.4

maxDepth	numTrees	featureSubsetStrategy
5	200	log2

RMSE
13360

Fig.12. Hyperparameters tuning and RMSE

3. Results

Model Comparison and Metrics used:

Since the problem is balanced using AUC Score is usually a good start as a metric. We used AUC Score as our primary metric Also, we used Accuracy along with Precision, Recall, and F1 score.

Metric	Formula
Accuracy	(TP + TN) / (TP + TN + FP + FN)
True Positive Rate (TPR)	TP / (TP + FN)
False Positive Rate (FPR)	FP / (FP + TN)
Precision	TP / (TP + FP)
Recall	TP / (TP + FN)
Area Under the Curve (AUC)	Integral area of plotting TPR vs FPR

Fig.13. Performance Evaluation Metrics

Algorithms	AUC Score	Accuracy	Precision	Recall	F1 Score
Random Forest	0.725	0.73	0.73	0.73	0.72
Gradient Boosting	0.709	0.65	0.65	0.65	0.65
Logistic Regression	0.709	0.66	0.65	0.66	0.65

Fig.14. Comparison of Classification Models

Under the default parameter setting, Gradient Boosting performs better than Random Forest by considering the AUC score as the metric. For comparison, AUC and Accuracy can be used together with most importantly as the problem is balanced. The best performing model with the best accuracy and AUC is the Random Forest Classifier with 72.5% Accuracy and an AUC score of 0.73. As for training and testing speed, logistic regression is much faster than the other two models. No problems concerning over or underfitting were encountered as the cross-validation AUC score and the testing AUC score were close to each other. If cross-validation AUC Score is greater than testing AUC score by a huge margin there can be overfitting and if it is less than cross-validation AUC Score, there can be underfitting. Also, for an article to be a hit the mashable should be focusing on how their content should be optimized. By increasing the number of keywords, the number of links embedded, the number of images, by including reference articles with high popularity, by setting a more subjective and positive title we can enhance the content of an article and focus more on social media articles during the weekdays.

4.Conclusion

In this study, we applied four different data mining algorithms on an Online News Population dataset that is published in the UCI repository. Accuracy, AUC Score, Precision, Recall, and F-measure are calculated to evaluate the applied algorithms. Random Forests with 150 number of trees, 9 maximum depth, and featureSubsetStrategy as auto obtained the best result with the AUC Score of about 0.725. However, there is still a possibility for improvement, by implementing a hybrid model using Bagging or AdaBoost with Random Forests and trying more advanced cross-validation methods although it may increase the training time. Also, more relevant features to the original dataset can be engineered and added. For instance, we could use all the words in an article as additional features, and then try the classifier such as Naive Bayes to see if it can achieve a better performance. The days of the week and the article categories along with few features of digital media and words were found to be the most contributing features in making an article popular.

4.1 References

[1] K. Fernandes, P. Vinagre, and P. Cortez, “A proactive intelligent decision support system for predicting the popularity of online news," in Portuguese Conference on Artificial Intelligence. Springer, 2015, pp. 535–546.

[2] An Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani Introduction to Data Mining by Tan, Steinbach, Kumar First Edition.

[3] Anon, 2016. UCI Machine Learning Repository: Online News Popularity Data Set. Archive.ics.uci.edu.

[4] Crisci, C., Ghattas, B. and Perera, G., 2012. A review of supervised machine learning algorithms and their applications to ecological data. Ecological Modelling, 240, pp.113-122.