iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** IV **Month of publication:** April 2019

DOI: https://doi.org/10.22214/ijraset.2019.4581

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue IV, Apr 2019- Available at www.ijraset.com

Predicting Online News Popularity

K. S. Naga Haritha[1], P. Vijaya Kumari[2], C. Nikhila Naga Jyothi[3], K. Manoj Kumar Naik[4]

Abstract: ***The consumption of online news increases day by day due to the wide spread adoption of smartphones and the rise of social networks. since it allows an easy and fast spread of information around the globe. Thus, predicting the popularity of online news is becoming a recent research trend. This project is aimed at giving the knowledge for authors, content providers, advertisers and even activists/politicians whether the article they publish is going to be popular or unpopular using python through machine learning. This research proposes various Machine learning approaches such as Logistic regression, Random Forest, Adaboost algorithms. In this project, we are going to take the day it is published, the category it belongs, and the features of article and predicts the popularity of the article.***

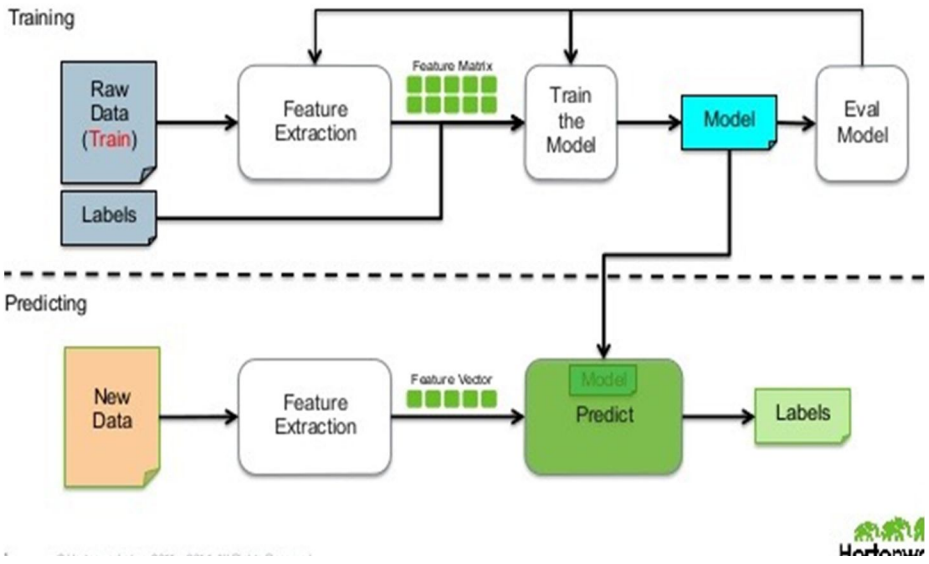
Keywords: Article, Number of shares, popularity, Classification algorithms.

I. INTRODUCTION

With the development of web 2.0,more and more people tend to read digital news rather than paper news ,which enables the online news industry to flourish and leads to substantial competition pressure in online news portals .if they know which kind of news will be popular, news portals can arrange news on their home pages accordingly and thus improve the competitiveness of their company. Besides, popular news represents the hot topics users care about, so advertisers can costumes more eye-catching ads as early as possible by predicting the popularity of online news. In addition ,governments can narrow down the search scope when identifying the potential harmful news and thus they can stop the respective articles release. Here in this paper ,the main aim is to predict the future popularity of news article prior to its publication estimating the number of shares, likes, and comments etc. (features of an article) .So the goal is to predict the online news article popularity. Here by using classification models to find the accuracy of each model and select the best model with high accuracy to predict the popularity.

II. LITERATURE STUDY

Many Data mining and Machine learning techniques has been developed for the early prediction of popularity of online news popularity. Some prediction approaches are based on analyzing early users' comments [1], or features about post contents and domains [2]. Another proposed method [3] predicted the article's popularity not only based on its own appeal, but also other articles that it is competing with. Prediction models with SVMs, Ranking SVMs [3], Naive Bayes are investigated, and more advanced algorithms such as Random Forest, Adaptive Boosting [4] could increase the precision.



ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue IV, Apr 2019- Available at www.ijraset.com

III. ARCHITECTURE

A. Data Analysis

Data Exploration 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. For each instance of the dataset :

Number of Attributes: 61 (58 predictive attributes, 2 non-predictive, 1 goal field)

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 encodingscheme, and the skewed feature like number of words in the article has been log transformed.

Data set

The dataset is publicly available at https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity# .

Dataset Information

Attributes

1) url: URL of the article (non-predictive)
2) timedelta: Days between the article publication and the dataset acquisition (non-predictive)
3) . n_tokens_title: Number of words in the title
4) n_tokens_content: Number of words in the content
5) n_unique_tokens: Rate of unique words in the content
6) n_non_stop_words: Rate of non-stop words in the content
7) n_non_stop_unique_tokens: Rate of unique non-stop words in the contents.
8) num_hrefs: Number of links
9) num_self_hrefs: Number of links to other articles published by Mashable
10) num_imgs: Number of images
11) num_videos: Number of videos
12) average_token_length: Average length of the words in the content
13) num_keywords: Number of keywords in the metadata
14) data_channel_is_lifestyle: Is data channel 'Lifestyle'?
15) data_channel_is_entertainment: Is data channel 'Entertainment'?
16) data_channel_is_bus: Is data channel ‘Business'?
17) data_channel_is_socmed: Is data channel 'Social Media’?
18) data_channel_is_tech: Is data channel 'Tech'?
19) data_channel_is_world: Is data channel ‘World'?
20) kw_min_min: Worst keyword (min. shares)
21) kw_max_min: Worst keyword (max. shares)
22) kw_avg_min: Worst keyword (avg. shares)
23) . kw_min_max: Best keyword (min. shares)
24) . kw_max_max: Best keyword (max. shares)
25) . kw_avg_max: Best keyword (avg. shares)
26) kw_min_avg: Avg. keyword (min. shares)
27) kw_max_avg: Avg. keyword (max. shares)
28) kw_avg_avg: Avg. keyword (avg. shares)
29) self_reference_min_shares: Min. shares of referenced articles in Mashable
30) self_reference_max_shares: Max. shares of referenced articles in Mashable
31) self_reference_avg_sharess: Avg. shares of referenced articles in Mashable
32) weekday_is_monday: Was the article published on a Monday?
33) weekday_is_tuesday: Was the article published on a Tuesday?
34) weekday_is_wednesday: Was the article published on a Wednesday?
35) weekday_is_thursday: Was the article published on a Thursday?
36) weekday_is_friday: Was the article published on a Friday?
37) weekday_is_saturday: Was the article published on a Saturday?
38) weekday_is_sunday: Was the article published on a Sunday?

39) is_weekend: Was the article published on the weekend?
40) LDA_00: Closeness to LDA topic 0
41) LDA_01: Closeness to LDA topic 1
42) LDA_02: Closeness to LDA topic 2
43) LDA_03: Closeness to LDA topic 3
44) . LDA_04: Closeness to LDA topic 4
45) global_subjectivity: Text subjectivity
46) . global_sentiment_polarity: Text sentiment polarity
47) global_rate_positive_words: Rate of positive words in the content
48) global_rate_negative_words: Rate of negative words in the content
49) rate_positive_words: Rate of positive words among non-neutral tokens
50) rate_negative_words: Rate of negative words among non-neutral tokens
51) avg_positive_polarity: Avg. polarity of positive words
52) min_positive_polarity: Min. polarity of positive words
53) max_positive_polarity: Max. polarity of positive words
54) avg_negative_polarity: Avg. polarity of negative words
55) . min_negative_polarity: Min. polarity of negative words
56) max_negative_polarity: Max. polarity of negative words
57) title_subjectivity: Title subjectivity
58) title_sentiment_polarity: Title polarity
59) abs_title_subjectivity: Absolute subjectivity level
60) abs_title_sentiment_polarity: Absolute polarity level
61) shares: Number of shares (target)

IV. ALGORITHMS AND TECHNIQUES

Since we formulate this problem as a binary classification problem three classification learning algorithms including Logistic Regression, RF, and Adaboost will be implemented.

A. *Logistic Regression*

Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts P(Y=1) as a function of X.

1) *Advantages:* Because of its efficient and straightforward nature, doesn't require high computation power, easy to implement, easily interpretable, used widely by data analyst and scientist. Also, it doesn't require scaling of features. Logistic regression provides a probability score for observations.
2) *Disadvantages:* Logistic regression is not able to handle a large number of categorical features /variables. It is vulnerable to over fitting. Also, can't solve the non-linear problem with the logistic regression that is why it requires a transformation of non-linear features. Logistic regression will not perform well with independent variables that are not correlated to the target variable and are very similar or correlated to each other.
3) *Parameters:* Class sklearn.linear_model.LogisticRegression (penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='warn', max_iter=100, multi_class='warn', verbose=0, warm_start=False, n_jobs=None)

B. *Random Forest*

Tree models are known to be high variance, low bias models. In consequence, they are prone to overfit the training data. This is catchy if we recapitulate what a tree model does if we do not prune it or introduce early stopping criteria like a minimum number of instances per leaf node. Well, it tries to split the data along the features until the instances are pure regarding the value of the target feature, there are no data left, or there are no features left to spit the dataset on. If one of the above holds true, we grow a leaf node. The consequence is that the tree model is grown to the maximal depth and there with tries to reshape the training data as precise as possible which can easily lead to over fitting. Another drawback of classical tree models like the (ID3 or CART) is that they are

relatively unstable. This instability can lead to the situation that a small change in the composition of the dataset leads to a completely different tree model.

1) *Parameters:* Class sklearn.ensemble.RandomForestClassifier (n_estimators='warn', criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1,min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None
2) *Advantages*
a) Reduction in over fitting: by averaging several trees, there is a significantly lower risk of over fitting.
b) Less variance: By using multiple trees, you reduce the chance of stumbling across a classifier that doesn't perform well because of the relationship between the train and test data.
3) *Disadvantages*
a) It takes more time to train samples.

C. ADABOOST

AdaBoost classifier is an ensemble classifier that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. sklearn provides a function called AdaBoostClassifier(). One tunable parameters is "n estimators", which is the maximum number of estimators at which boosting is terminated. Another is "learning rate", which means the contribution of each classifier shrinks by the value of "learning rate". There is a trade-off between "n estimators" and "learning rate". The boosting can help in learning complex non-linear hypothesis which is necessary because the dataset here is not linearly separable. In this project, we will first implement all three algorithms with the default hyperparameter settings, then we will use grid search method to refine the hyperparameters and thereby improve the model's performance. For logistic regression, we will tune the hyperparameter"C", which is the inverse of regularization strength. "C" is basically used for control overfitting. A smaller "C" means higher regularization over the model parameters, thus smaller "C" will decrease the magnitude of model parameters to prevent over-fitting. For RF, we will tune hyperparameter "n estimators", which is the maximum number of trees you can build before taking the maximum voting or averages of predictions. Higher number of trees can give better performance but makes code running slower. For Adaboost, the default base estimator is decision tree classifier. Similar to RF, hyperparameter "n estimators" represents the maximum number of trees you can build before termination. Higher number of trees may lead to better performance but will decrease the model running speed. The trade-off between "n estimators" and "learning rate" is as follows: a larger "learning rate" typically means smaller "n estimators" in Adaboost but might lead to over-fitting, while a smaller "learning rate" typically means larger "n estimators" and it will control over-fitting but slow down the running speed.

V. METRICS

In the data set the class labels (+, -) are very closely balanced. we are using the following metrics. (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

A. *Accuracy = true positives + true negatives/ dataset size (1)*

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.

The performance of a model cannot be assessed by considering only the accuracy, because there is a possibility for misleading. Therefore this experiment considers the F1 score along with the accuracy for evaluation.

(b) F1-score: F1-score is an unweighted measure for accuracy by taking harmonic mean of precision and recall, which can be computed as

B. *F1 = 2 · precision · recall/ precision + recall (2)*

Precision and Recall are defined as Precision=TP/ (TP+FP), Recall=TP/ (TP+FN), where TP=True Positive FP=False Positive FN=False Negative

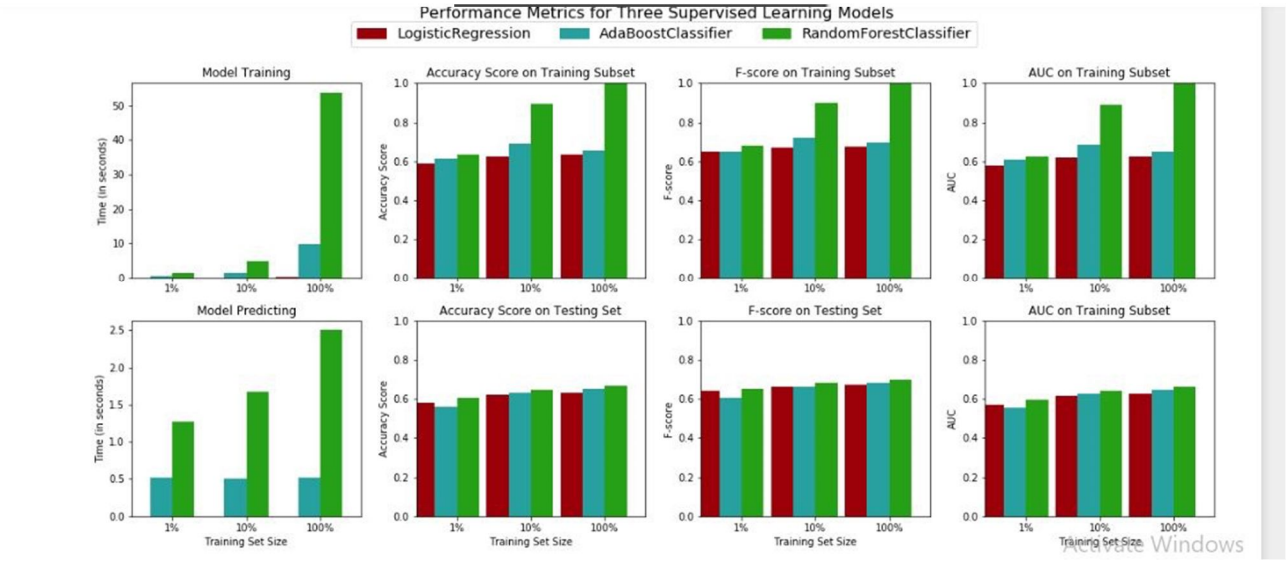
It is a robust measurement since it is independent of data class distribution.

For indicating the usefulness of the classifier the following metric(AUC) is used. (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 all above three metrics, the higher value of the metric means the better performance of model.

VI. CONCLUSION

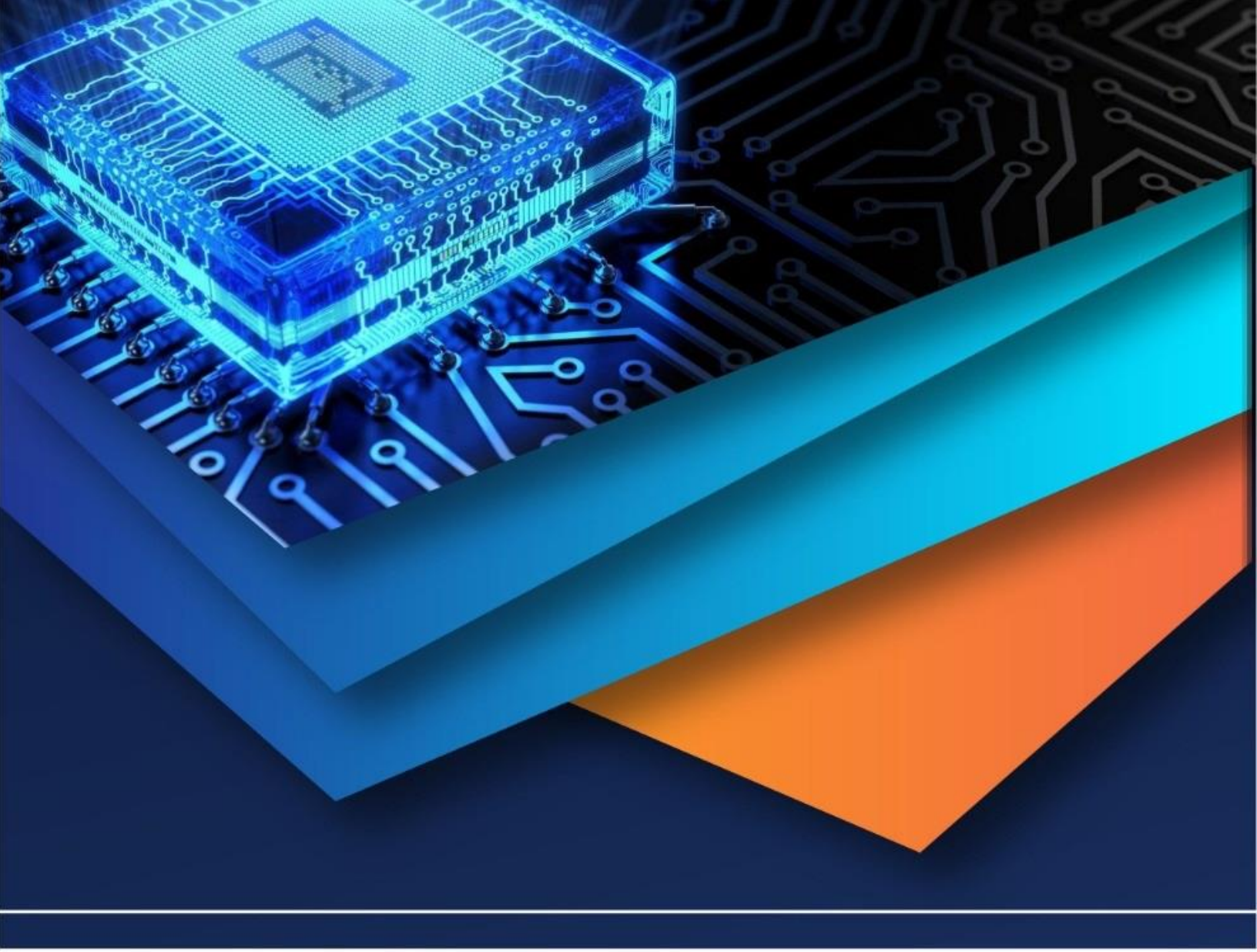
In this part, the performance of three classifiers with refined hyper parameters is finally displayed . As shown in the visualization, the training and testing time of RF is greatly increased since 500 trees are used in the forest, but it helps RF achieve the best performance in terms ACCURACY F1 SCORE and AUC



Finally we conclude that ADABOOST algorithm is best for predicting the online news popularity.

REFERENCES

[1] Tatar, Alexandru, et al. "Predicting the popularity of onlinearticles based on user comments." Proceedings of theInternational Conference on Web Intelligence, Mining andSemantics. ACM, 2011.
[2] "Predicting the Popularity of Social News Posts." 2013cs229 projects. Joe Maguire Scott Michelson.
[3] Hensinger, Elena, Ilias Flaounas, and Nello Cristianini."Modelling and predicting news popularity." Pattern Analysis and Applications 16.4 (2013): 623-635.
[4] K. Fernandes, P. Vinagre and P. Cortez. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. Proceedings of the 17th EPIA 2015- Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY