Online News Popularity Prediction

Chitra Paryani
NU ID - 1869343

***Abstract* -** The significant numbers of articles are published using online platform like Medium, Facebook, NYTimes, etc. These platforms also give us an opportunity to share them in various social platforms like Facebook, twitter, LinkedIn, Reddit, Google+, etc. When these articles are shared by enormous number of people, articles become more popular as compared by those articles which are not shared.

In this project, I am analyzing articles popularity using target variable - shares which indicated number of times that article is shared. This can help online news companies who can check popularity of article before publishing it. Also, it can help businesses which rely heavily on social networks to grow and flourish their business.

I am viewing this problem as classification problem and applied various algorithms like logistic regression, Random Forest, SVM, Naïve Bayes, and Neural Network. Additionally, I have also applied regularization, hyper-parameter tuning, and cross-validation technique. Highest accuracy is achieved using Random Forest algorithm which is 68%

Keywords – Machine Learning, Logistic Regressor, Random Forest; Support Vector Machine, Naive Bayes, Multi-Layer Perceptron, News Popularity Prediction

I. INTRODUCTION

I am using Online News Popularity dataset from UCI Machine Learning Repository to predict Online News Popularity. The dataset consists of 39644 records and 61 attributes from which 2 are non-predictive attributes, 58 are predictive attributes and 1 is target variable which is Number of shares.

Dataset is provided by Mashable which is a global social news company. Dataset provides statistical summary of the articles instead of original articles. However, dataset consist of one non-predictive attribute – URL which consist of link to article and can be used for sentiment analysis in future.

Dataset contains target variable, shares which is a continuous output value ranging from 1 to 843300. To convert this problem into classification, I have used shares variable's median value 1400 and converted it to binary values 0 and 1 which has divided the dataset into two equal parts.

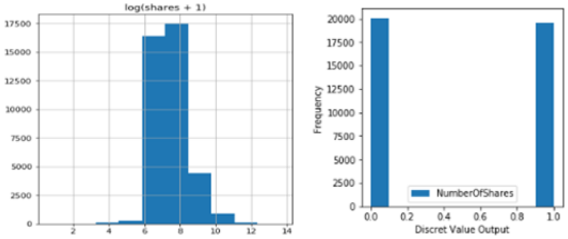

Fig 1 – Target value distribution

Initially, I have performed Exploratory Data Analysis to find null/missing values, plotted histogram to find missing values which are represented as zero. Additionally, created boxplot to find outliers, and heatmap to find co-relation between the data.

Dataset do not contain any missing value which is represented as NA. Further, while analyzing the dataset for 0 values, I found 0 values in rate_positive_words, rate_negative _words, and average_token_length. Removed all these values while doing data clean up.

s

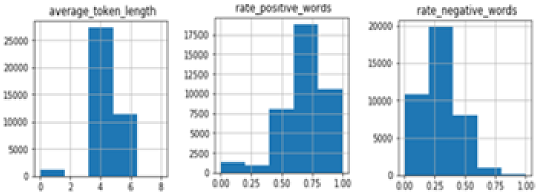

Fig 2 – Histogram

Outliers are present in few attributes in the dataset such as n_tokens_content which represents total number of words in the article whereas all other values like n_Unique_token, average_token_length

contains values between 0 and 1. Kw_Min_min which is total number of minimum worst words in the article whereas kw_avg_min represented between range 0 and 1. Removed all these values while doing data clean up.

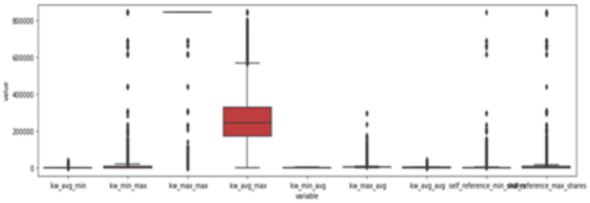

Fig 3 – Boxplot

After finding missing values and outliers, I tried to find correlation between the predictors w.r.t. target variable – Number of shares by using corr function in python and by plotting heatmap and found variables are not highly corelated w.r.t target variable shares.

Feature which is of maximum importance to target variable – shares contains value 0.028 and name of the predictor is kw_max_avg.

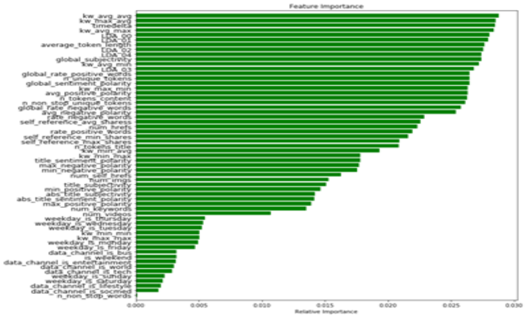

Fig 4 – Information Value Summary

Next, I have analyzed correlation between the predictors using heatmap. Dataset contains various binary predictors such as Is weekday Saturday, is_weekday_sunday replaced by predictor Is_weekend. Also, predictors such as Is_weekday_Monday, is_weekday _Tuesday, Is_weekday Wednesday, Is_weekday_Thursday, Is_weekday_Friday removed and replaced by Is_weekday.

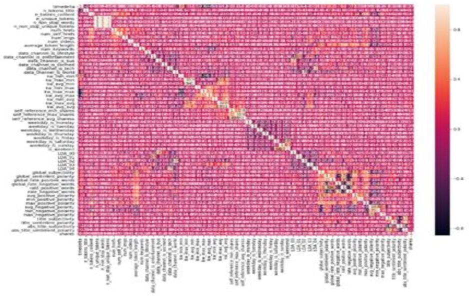

Fig 5 – Heatmap

I cleaned the data by removing missing values, outliers and other highly corelated columns. Finally, my dataset contains 56 predictors and 38458 records.

I divided the dataset into training set which contains 70% of the dataset and testing set which contains remaining 30% of the dataset. Training set is used for applying algorithms whereas testing set is used to predict accuracy of the model. At the end, compared the accuracy of all the models and found the best predicted model.

i) Logistic Regression

The first machine learning algorithm which I am applying to my dataset is Logistic Regression. I am using binary logistic model to estimate probability of Number of Shares based on 52 predictor features.

$$\hat{p} = \frac{\exp(B_0 + B_1 X)}{1 + \exp(B_0 + B_1 x)} = \frac{e^{B_0 + B_1 x}}{1 + e^{B_0 + B_1 x}}$$

s
s:

$$\hat{p} = \frac{\exp(B_0 + B_1 X)}{1 + \exp(B_0 + B_1 x)} = \frac{e^{B_0 + B_1 x}}{1 + e^{B_0 + B_1 x}}$$

The accuracy achieved by predicting logistic regressor model is 60%. Thereafter, I applied Lasso regularization, Tuned the hyperparameters and cross validated the model and achieved accuracy of 63% and AUC 0.66.

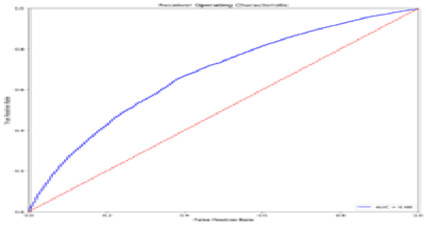

Fig 6 – ROC Curve (LR)

ii) Support Vector Machine

The next supervised machine learning model which I have applied to my dataset is Support Vector Machine. This algorithm outputs an optimal hyperplane dividing plane into two parts where in each class can lay in other side. In my dataset, Support Vector Machine divides hyperplane into two parts i.e. popular and not popular and provide optimal hyperplane.

The accuracy achieved by using Support Vector Machine is 51%. Then, I tuned hyper-parameters Kernel as linear, Regularization as 1 and gamma as 1 and cross validated the model and achieved accuracy of the models achieved as 61%.

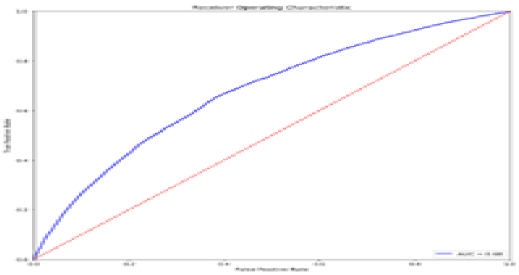

Fig 7 – ROC Curve (SVM)

iii) Random Forest

The next Ensemble learning algorithm which I applied to my dataset is Random Forest Classifier. Random Forest operate by constructing a multitude of decision tree at training time and outputting the class that is mode of the classes. Random Forest correct for decision tree habits of overfitting training set.

The accuracy achieved by using 400 trees in Random Forest algorithm is 67% which is highest in my dataset. Thereafter, I performed hyper-parameter tuning and cross-validated the model and achieved same accuracy of 67%.

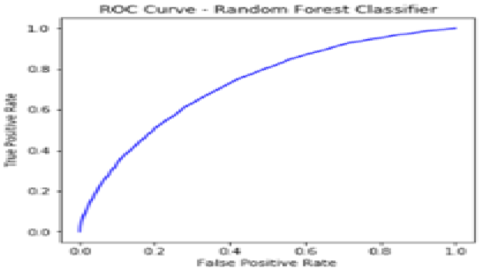

Fig 8 – ROC Curve (Random Forest)

iv) Naïve Bayes Algorithm

The next powerful algorithm which I have applied to my dataset is Naïve Bayes algorithm which belongs to the family of probabilistic classifiers. The Naïve Bayes classifier which I am using is Bernoulli models which is designed for binary/Boolean features.

The accuracy of model achieved by applying Naïve Bayes algorithm is 61%. Thereafter, performed hyperparameter tuning and cross validated the mode and achieved accuracy of 61%.

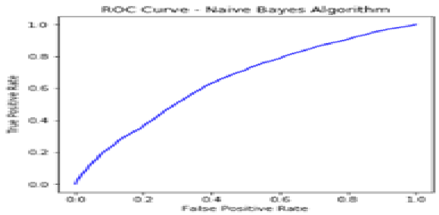

Fig 9 – ROC Curve (Naïve Bayes)

iv) Single layer Perceptron

In machine learning, perceptron is an algorithm that performs classification based on a linear predictor function for supervised learning in which an input is represented by a vector of numbers, and it predicts output vector by analyzing if target belongs to particular class or not.

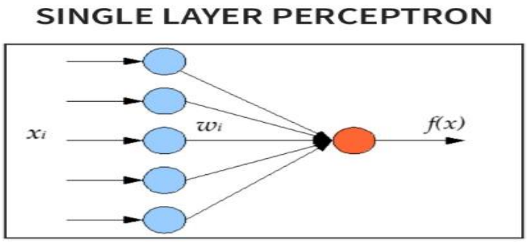

Fig 10 – Single Layer Perceptron

I applied Single Layer Perceptron to my dataset by adding some initial random weights and added bias to it. The accuracy achieved using SLP is 58%.

Then, I applied Backpropagation to Single Layer Perceptron by calculating cost function using cross entropy and applying gradient descent optimizer. Performed 10000 epochs and achieved accuracy of 64%

v) Multi Layer Perceptron

Multilayer Perceptron (MLP) includes at least one hidden layer other than input and output layer. Single layer sensors can only learn linear functions, while multi-layer sensors can learn from non-linear functions. To understand non-linear functions in my data, I applied multilayer perceptron.

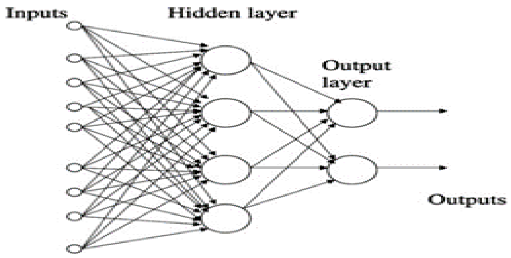

Fig 11 – Multi Layer Perceptron with a hidden layer

The multilayer perceptron learns from a training data and make accurate predictions based on the new data points given. In my dataset, I have used backpropogation algorithm which basically means it learns from its mistakes.

Input layer, intermediate hidden layer and output layer are connected with weights. The purpose of learning is to assign right weight to edges. By entering vectors, these weights can determine output vector.

The first step is forward transmission, second step is backpropogation and weight update and finally determining correct output.

I have applied MLP in my dataset. Initially, I have created two hidden layers, applied some random weight, and added bias to it. The error cost is calculated using cross entropy and optimized using gradient descent optimizer. For total 10000 epochs and each epoch is separated at 1000th step, at step 0, accuracy of model is 53% and at step 10000, accuracy achieved is 58%.

III RESULTS

I have applied five machine learning algorithms in my dataset. I also did in depth analysis on each algorithm by performing hyperparameter tuning and cross validation. First algorithm which I used for my analysis is logistic regression which give a descent accuracy of 60%. After performing hyperparameter tuning and cross validation, accuracy of model is increased to 64%.

Support Vector Machine with kernel as linear, regularization as 0.001 and gamma as 1 gave an accuracy of 60%. After performing hyperparameter tuning and cross validation, accuracy achieved is 61%.

Random Forest Classifier using 400 trees give the highest accuracy of 67%. After performing, hyperparameter tuning and cross validation, algorithm give an accuracy of same 67% only.

Naïve Bayes algorithm give an accuracy of 61%. After performing, hyperparameter tuning and cross validation, algorithm gave same accuracy of 61%

Using Neural Networks, Single Layer Perceptron gave an accuracy of 58%. However, using 10000 epochs gave an accuracy of 64%.

Using Neural Networks, Multi Layer Perceptron gave an accuracy of 53% at step 0 and 58% at step 10000.

Results		
Algorithm	**Predictors**	**Accuracy**
Logistic Regression	52 Predictors	60%
	Hyperparameter tuning and cross validation	64%
Random Forest	52 Predictors	67%
	Hyperparameter tuning and cross validation	67%
Support Vector Machine	52 Predictors	61%
	Hyperparameter tuning and cross validation	61%
Naïve Bayes	52 Predictors	61%
	Hyperparameter tuning and cross validation	61%
Neural Networks	Single Layer Perceptron	64%
	Multi Layer Perceptron	58%

IV DISCUSSION

To predict popularity of an article, initially, I performed Exploratory Data analysis to find missing values, outliers and correlation between predictors which was not an easy task as dataset have 62 predictors.

Then, I applied different five machine learning algorithms along with hyperparameter tuning and cross validation and received highest accuracy using random forest classifier which is 67% followed by logistic regressor which is 64%

As accuracy achieved is not great, a lot of future work can be done on this project.

Dataset contains URL attribute which is a link to original article that can be used to extract article content and perform text and sentiment analysis on the same.

Further, I applied only handful of parameters to tune the models due to resource constraints. A lot of different variations of parameters could be used to tune parameters and predict accuracy of the model.

Also, I just used two deep learning algorithms in my dataset, single layer perceptron and multilayer perceptron which manifested better accuracy enhancement as compared to other algorithms. Different deep learning algorithms with more variations in parameter could be tried for more accuracy as I was not able to perform due to resource constraints.

V References

http://www-scf.usc.edu/~jiayingg/

https://www.programcreek.com

https://ieeexplore.ieee.org/document/7802890

http://cs229.stanford.edu/proj2015/328_report.pdf

https://www.linkedin.com/pulse/online-news-popularity-trend-analysis-krunal-khatri/

http://inseaddataanalytics.github.io/INSEADAnalytics/groupprojects/January2017/MashableNews.html

https://github.com/nikbearbrown/INFO_7390/tree/master/Projects/Research_Papers

http://cs229.stanford.edu/proj2016/report/GengYuanWang-PredictingPopularityOfPostsOnHackerNews-report.pdf

http://cs229.stanford.edu/proj2016/report/JohnsonWeinberger-PredictingNewsSharing-report.pdf

http://cs229.stanford.edu/proj2015/328_report.pdf