Article Popularity Classification

KENNEDY AGUSI, University of Rochester, USA

SAM BLUMBERG, University of Rochester, USA

1 INTRODUCTION

For our project, we decided we wanted to work with the online news popularity data set which was scrubbed from mashable.com. The dataset contains 39,645 tuples and 61 attributes about numerous online news articles. with the target attribute being predicting the number of shares the article will receive on social networks. Among the features are 22 Boolean attributes and 39 continuous valued attributes. There were no missing values in our dataset. The goal of our project is to be able to take an online article and be able to predict how popular it will be. We believe this is a very interesting topic, and this project will be able to help understand what aspects of an article lead to it being popular.

2 RELATED WORK

This dataset was used by K. Fernandes, P. Vinagre and P. Cortez in their research titled “A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News.” This was presented at the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence. During their research they used Random Forest, Adaboost, SVM, KNN and Naïve Bayes, with Random Forest performing the best with 67% accuracy.

3 METHODOLOGY

Data cleaning: We used Z-score with threshold of "3" to detect outliers in our dataset. This means if the Z-score value is greater than or less than 3, that data point will be identified as outliers.

Data normalization: We normalized all the continuous valued attribute using range of 0-1. For the class attribute (shares), we did something different. We divided it into two binary (0 and 1) bins with a threshold of 1400. This means if number of shares is less than 1400, we assign “0” as the value (meaning the article is not popular and if the number of shares is above 1400, we assign “1” as the value meaning that the article is popular.

Attribute selection: We first used “Forward Selection Wrapper method” for our feature selection, but we found out it was taking too long, so we used LASSO feature selection to reduce the number of attributes from 61 to 34.

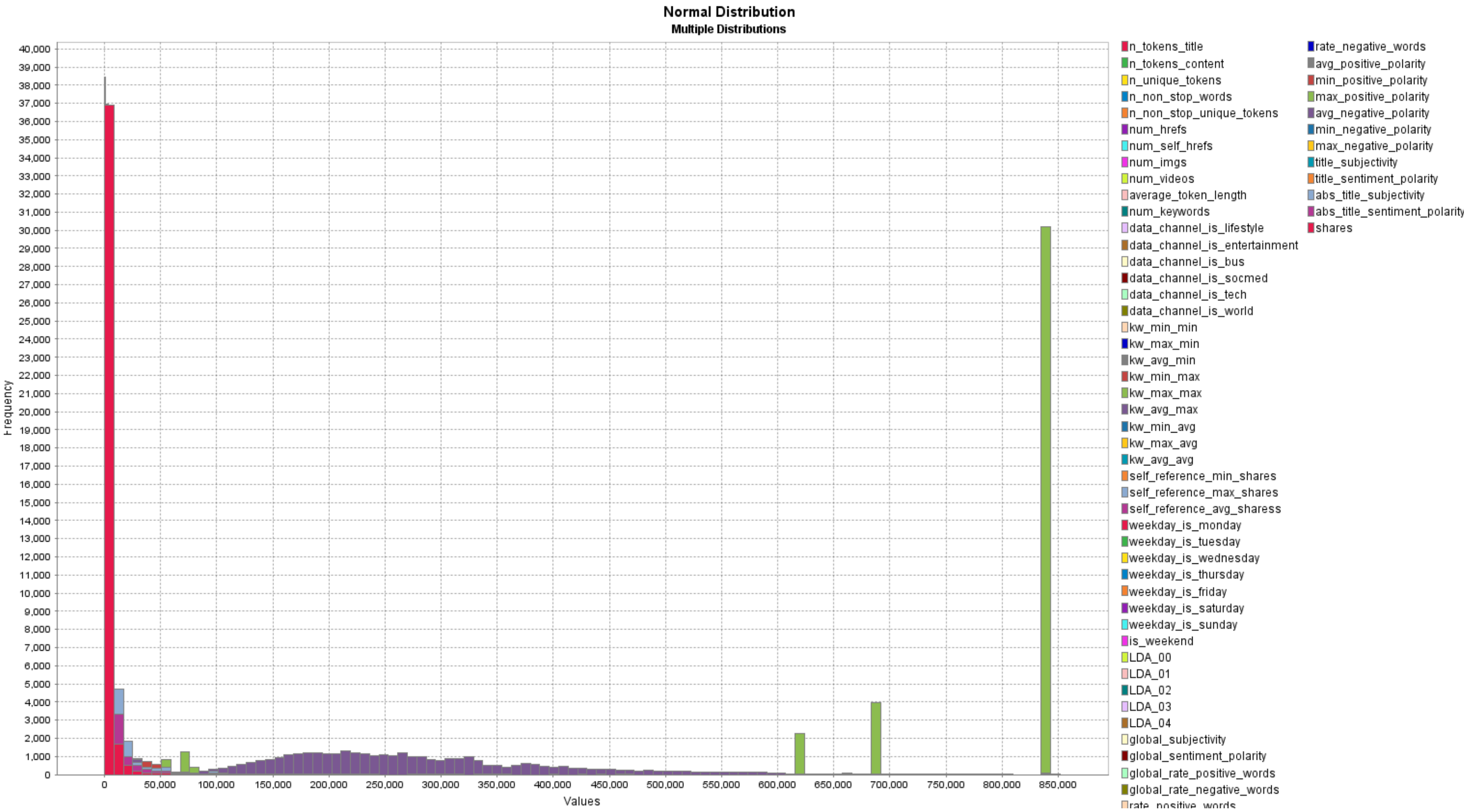
Dataset Statistics summary:

The summary statistics of our dataset is described in the table below

ATTRIBUTES	MEAN	VARIANCE	STD_DEV	MIN	MAX
n_tokens_title	10.39875	4.469152	2.114037	2	23
n_tokens_content	546.5147	221942.3	471.1075	0	8474
n_unique_tokens	0.548216	12.39539	3.520708	0	701
n_non_stop_words	0.996469	27.36578	5.231231	0	1042
n_non_stop_unique_tokens	0.689175	10.65903	3.264816	0	650
num_hrefs	10.88369	128.4146	11.33202	0	304
num_self_hrefs	3.293638	14.86211	3.855141	0	116
num_imgs	4.544143	69.04669	8.309434	0	128
num_videos	1.249874	16.87447	4.107855	0	91
average_token_length	4.548239	0.713021	0.844406	0	8.041534
num_keywords	7.223767	3.644779	1.90913	1	10
data_channel_is_lifestyle	0.052946	0.050144	0.223929	0	1
data_channel_is_entertainment	0.178009	0.146326	0.382525	0	1
data_channel_is_bus	0.157855	0.13294	0.36461	0	1
data_channel_is_socmed	0.058597	0.055164	0.234871	0	1
data_channel_is_tech	0.185299	0.150967	0.388545	0	1
data_channel_is_world	0.212567	0.167386	0.409129	0	1
kw_min_min	26.1068	4848.785	69.63322	-1	377
kw_max_min	1153.952	14884094	3857.991	0	298400
kw_avg_min	312.367	385372.6	620.7839	-1	42827.86
kw_min_max	13612.35	3.36E+09	57986.03	0	843300
kw_max_max	752324.1	4.6E+10	214502.1	0	843300
kw_avg_max	259281.9	1.83E+10	135102.2	0	843300
kw_min_avg	1117.147	1293808	1137.457	-1	3613.04
kw_max_avg	5657.211	37196239	6098.872	0	298400
kw_avg_avg	3135.859	1737520	1318.15	0	43567.66
self_reference_min_shares	3998.755	3.9E+08	19738.67	0	843300
self_reference_max_shares	10329.21	1.68E+09	41027.58	0	843300
self_reference_avg_sharess	6401.698	5.86E+08	24211.33	0	843300
weekday_is_monday	0.16802	0.139793	0.373889	0	1
weekday_is_tuesday	0.186409	0.151665	0.389441	0	1
weekday_is_wednesday	0.187544	0.152375	0.390353	0	1

weekday_is_thursday	0.183306	0.149709	0.386922	0	1
weekday_is_friday	0.143805	0.123128	0.350896	0	1
weekday_is_saturday	0.061876	0.058049	0.240933	0	1
weekday_is_sunday	0.069039	0.064275	0.253524	0	1
is_weekend	0.130915	0.113779	0.337312	0	1
LDA_00	0.184599	0.069156	0.262975	0	0.926994
LDA_01s	0.141256	0.048271	0.219707	0	0.925947
LDA_02	0.216321	0.079606	0.282145	0	0.919999
LDA_03	0.22377	0.087138	0.295191	0	0.926534
LDA_04	0.234029	0.083627	0.289183	0	0.927191
global_subjectivity	0.44337	0.013615	0.116685	0	1
global_sentiment_polarity	0.119309	0.009396	0.096931	-0.39375	0.727841
global_rate_positive_words	0.039625	0.000304	0.017429	0	0.155488
global_rate_negative_words	0.016612	0.000117	0.010828	0	0.184932
rate_positive_words	0.68215	0.036178	0.190206	0	1
rate_negative_words	0.287934	0.024385	0.156156	0	1
avg_positive_polarity	0.353825	0.010929	0.104542	0	1
min_positive_polarity	0.095446	0.005086	0.071315	0	1
max_positive_polarity	0.756728	0.061398	0.247786	0	1
avg_negative_polarity	-0.25952	0.016314	0.127726	-1	0
min_negative_polarity	-0.52194	0.084268	0.29029	-1	0
max_negative_polarity	-0.1075	0.009096	0.095373	-1	0
title_subjectivity	0.282353	0.105136	0.324247	0	1
title_sentiment_polarity	0.071425	0.070464	0.26545	-1	1
abs_title_subjectivity	0.341843	0.035642	0.188791	0	0.5
abs_title_sentiment_polarity	0.156064	0.051209	0.226294	0	1
shares	3395.38	1.35E+08	11626.95	1	843300

Normal Distribution of Various Attributes



4 EXPERIMENT

We used two classification models for our prediction: A Naïve Bayes classifier and a Neural Network (both coded from scratch without any framework). We divided the dataset into two parts – 70% training set and 30% test set.

We got a recognition rate of 50.78% and error rate of 0.49% with our Naïve Bayes classifier.

For our neural network we used a learning rate of 0.7 and a momentum constant of 0.9. We first ran it without doing feature selection on the dataset and we also did not remove any outliers. We used a hidden layer of 60 neurons and ran it for 3,000 epochs. This resulted with a prediction rate of 55% and an error rate of 45%. We then reduced the number of attributes by using LASSO to do feature selection. We kept the number of hidden nodes at 60. After running for 3,000 epochs we got a recognition rate of 57% and an error rate of 43%. We were not satisfied with this result because it was below what the previous researchers got while using this dataset.

As a next step we tried removed outliers from our dataset using the method described under data preprocessing. We then ran the program with the same number of hidden layers. After 1,000 epochs, we got a recognition rate of 65% and error rate of 35%. Finally, we decided to add a second hidden layer with 150 total neurons (100 neurons in 1st hidden layer and 50 neurons in the next layer). After running it for under 2,000 epochs we got a recognition rate of 79.1% and error rate of 20.9% which is higher then what previous researchers achieved. After running our neural network for several more epochs, we concluded that it performed better than our Naïve Bayes classifier for our chosen dataset.

5 CONCLUSION

We learned a lot while doing this project. Especially by coding our models from scratch we learned various optimization tricks you can apply to backpropagation to make it more efficient at learning. For example, we learned how use a momentum constant to make the gradient descent smoother. Another powerful lesson we learned while doing this project is the importance of outlier removal. From the experiment we presented above, one can clearly see that removing outliers can improve both the model's learning rate as well as accuracy.

Given the recent growth of digital media, we feel that a possible next stage of this research might be to explore the correlation between the authenticity of an article and its popularity.

BIBLIOGRAPHY

[1] 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.