10/30/2019

PGP DSE: Capstone Project

Online News Popularity Data Set

GROUP H

Mentor: Animesh Tiwari

Members:

Shrey Kumar

Tarun Tyagi

Tanu Vishnoi

Nikhil Jain

Neeraj Kushwaha

GREAT LAKES INSTITUTE OF MANAGEMENT (GREAT LEARNING)

Contents

Acknowledgements

This report was made with the help and support of Great Lakes (Great Learning) Institute of Management. We would firstly like to thank and express my deepest gratitude to our mentor Animesh Tiwari for this project. We want to thank him for guiding us and assisting us whenever we needed.

Finally, our deepest appreciation goes to all our parents and siblings. We are thankful to have them by our side regardless of everything.

This achievement wouldn't have been possible without each and one of them.

Thankyou.

Abstract and Problem Statement

We have a dataset which summarizes a set of features about the articles which have been published by the company Mashable in a short period of time of two years. It is an online news popularity dataset.

The dataset has 39644 number of instances and 61 attributes. There are no missing values.

The problem is a regression problem and a classification problem as well. The objective in a regression problem is to predict the number of shares (target column) of an article. Whereas in a classification problem, we have set the threshold of 1400 shares (below 1400 shares would be an Unpopular Article and above 1400 shares would be a Popular Article), so we need to predict on this basis.

Exploratory Data Analysis

Univariate Analysis

The main aim behind univariate analysis is to visualize the spread of our data to understand the distribution it follows and to recognize the outliers or any anomalies that may be present in data.

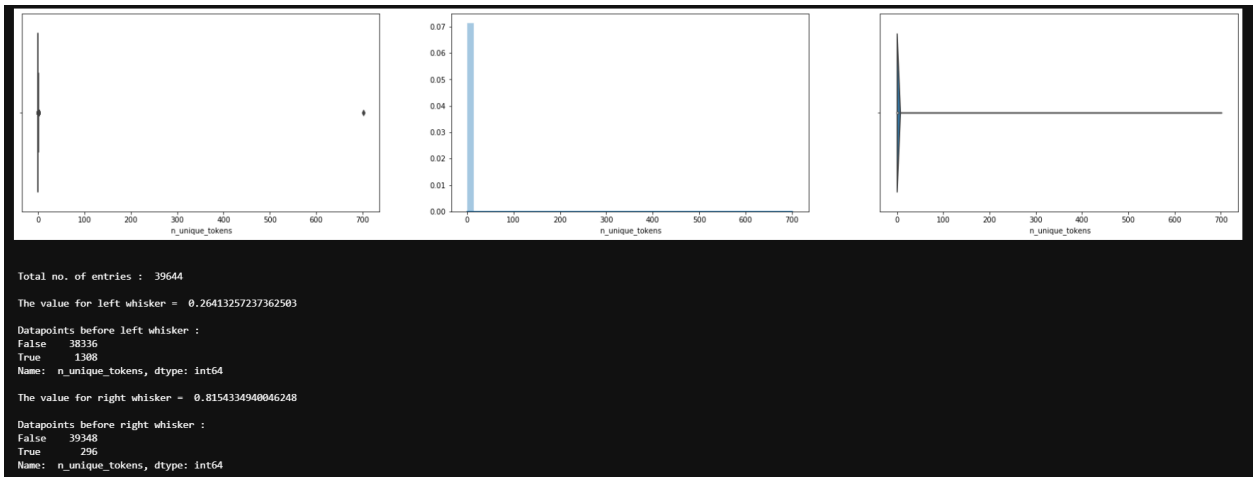
For analysis of same, 3 graphs are used namely distribution plot: to for better visualization of skewness, boxplot: for better understanding of outlier's existence and lastly violin plot: to further help with our analysis. These graphs are easily imported from seaborn library and are then used inside a function.

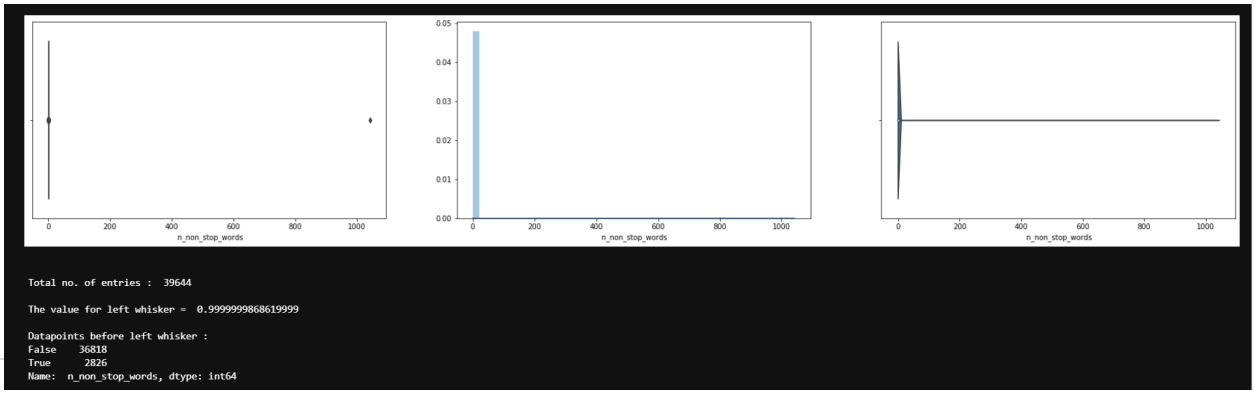
From the previous data analysis and data dictionary we recognized that there are many columns that are the dummies version of some pre-existing feature and so in order to analyse that we used count plot to quantify the occurrence of binary labels such as 1s and 0s.

Along with the graphical visualizations, some important statistical inferences such as no. of outlier's present, values at different percentiles, mean, median, values count etc. are also extracted from every column.

Observation:

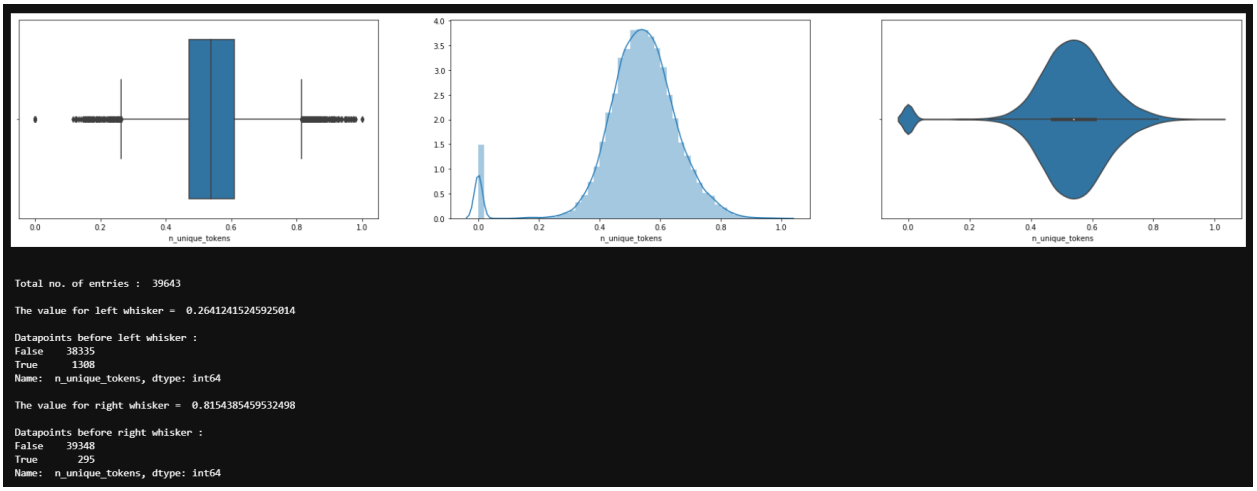
There are noticeable outliers that are present in majority of columns (excluding the ones





that contains binary labels) however there are some columns that have some anomalies,

This anomaly is found in several other columns however the cause of this anomaly is a single row which we decide to drop in order to rectify it.



After rectification:

Bivariate Analysis

For this analysis we take every independent feature and plot it against our target variable to understand the relation between the same.

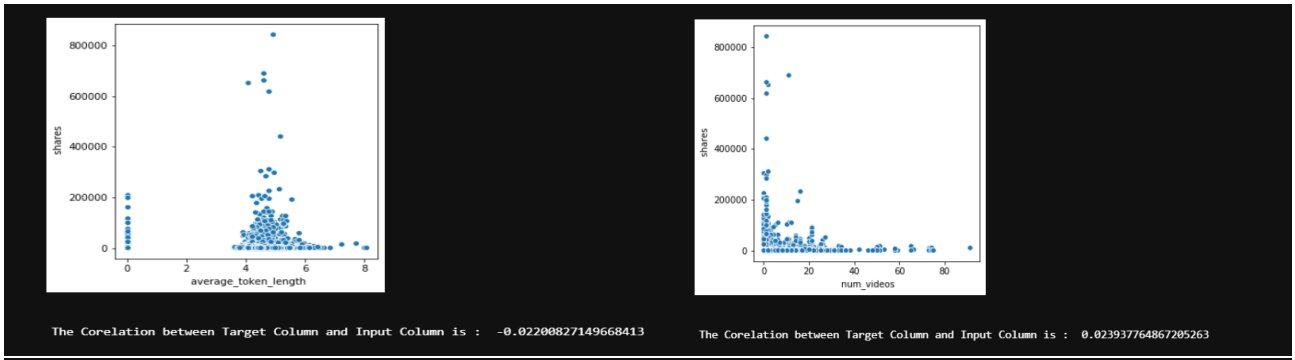
Scatter plot is very useful plot to visualize the 2 or more features together and hence we use the same for our analysis too, library is again seaborn for scatterplots.

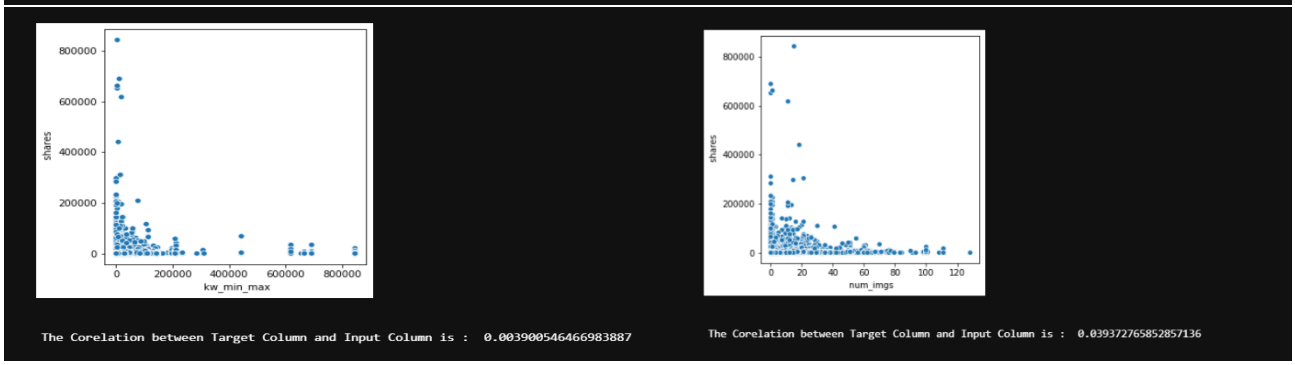
We would also like to do a scatter plot between every feature however due to large no. of features in our data it would be difficult to analyse the pair plot. So, we stick with scatter plot between target and independent feature and would rely on other tools such as correlation matrix, OLS statistical model to understand the dependencies of one variable with another.

Observations:

The main observation that we inferred from our analysis is that most of the data points are forming a cluster, and rest of them are dispersed in a linear manner by the axes.

The impact of outliers can also be seen below as there are several datapoints that exists at higher position towards axes.





Upon further analysis using correlation matrix we were also able to identify pairs of have

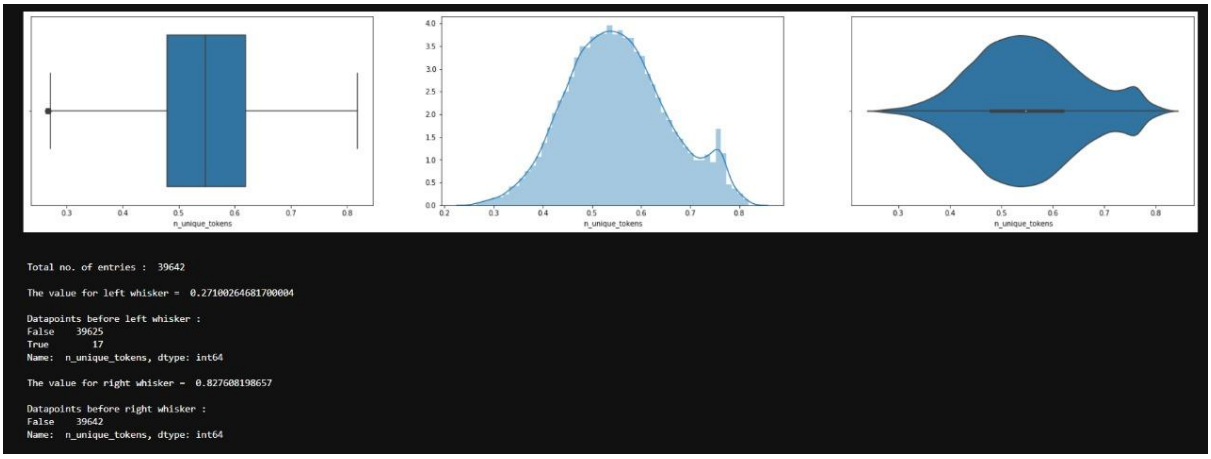
n_non_stop_unique_tokens	n_unique_tokens	0.999852
n_unique_tokens	n_non_stop_unique_tokens	0.999852
n_non_stop_words	n_unique_tokens	0.999572
n_unique_tokens	n_non_stop_words	0.999572
n_non_stop_words	n_non_stop_unique_tokens	0.999532
n_non_stop_unique_tokens	n_non_stop_words	0.999532
kw_avg_min	kw_max_min	0.940529
kw_max_min	kw_avg_min	0.940529

that have very high correlation with each other.

After doing both univariate and bivariate analysis on our data we were able to infer several important points such as , the data is highly unevenly spread and occurrence of outliers is a major problem that we have to address and occurrence of highly corelated column also needs to be fixed as same they don't add any new information to our target variable but may hinder with our accuracy scores.

Data Cleaning

As we have observed, there were no missing values and no special characters which needed to be cleaned in the dataset. Although there were a high number of outliers present in the dataset. With the help of Inter Quartile Range (IQR), we detected the outliers present in the dataset. We detected them by taking out the upper limit by Q3 + 1.5* IQR and the lower limit by Q1 – 1.5* IQR. There were 2.818 % of outliers present in the dataset excluding the target column. We converted all the outliers into null values. We treated the outliers by the Multivariate Imputation by Chained Equations (MICE) algorithm. We did not treat them through mean median or mode because it has a huge disadvantage that it reduces the variance and the standard error in the imputed values, and it does not regard the correlations between the variables. The methodological working behind the MICE algorithm is that numerous imputations are done in small steps, and in every step, it requires indicative checking which helps in imputing one single value in the best way. It checks and learns the missing data pattern and then impute the missing values n number of times and then checks the quality of the imputed values by analysing it if it's the best fit. It works through predictive mean matching and all the regression models for comparison.



We can see in this figure the outliers are treated using mice and there is substantial reduction in outliers.

Principal Component Analysis

As we know dealing with numerous amounts of features is tedious. We have applied PCA to our dataset. PCA is a dimensionality reduction method. Dimensionality reduction is the process of reducing the columns/features in the dataset and it helps in compressing the data, it speeds up the computation time of algorithms. It removes the features which provide the least informative features and keeps the ones which has the highest of explained variance ratio. It minimizes n number of dimensions of data into 2D or 3D and

hence it becomes easy for us to visualize the data accurately. PCA deals with large datasets, it removes noise and improves signal in our dataset; dimensionality reduction can help the model in learning more efficiently.

It is often useful to measure data in terms of its principal components rather than on a normal x-y axis. It gives the features which has the most variance, the features where the data is most spread out. We have done PCA with n_components (number of components) as 2, 4 and 10 with solver auto where the metrics achieved with R2 Score as 13.87 % and Root Mean Squared Error as 0.5581.

Ordinary Least Squares Regression

We applied Ordinary Least Squares regression method to our dataset. It is a statistical technique to predict the relationship between the predictors and the target feature. It evaluates by minimizing the difference between the predicted values and the actual values of the target column (shares). Here, the null hypothesis is framed in such a manner that the null hypothesis would be rejected. The null hypothesis here would be that there is no relationship between the predictor features and the target feature. The alternate hypothesis is that there is a relationship between the predictor features and the target features. When we apply OLS regression, we consider the p-value as our significance level for our features for us to know if a predictor feature has a relationship with the target or not. P-value is basically the probability of the null hypothesis being rejected. Here, we have considered our p-value as 0.05 (5%), which means if the p-value of a feature is above 0.05, we will fail to reject the null hypothesis meaning that the feature has no relationship with the target feature.

We apply backward/forward elimination to remove the predictor variables which are the least significant based on p-value. Backward Elimination has been chosen by us for this project as it is effective, it is a feature selection technique. After applying OLS regression successfully, we achieved a R-Squared of 13.9% and an Adjusted R-Square of 13.8% with degrees of freedom of the model being 40.

Standard Scaler

Since different features have different ranges of values, some features may affect the prediction more than others. Therefore, a common practice is to standardize and normalize the data for many machine learning estimators: they might give poor prediction results if the individual features are not normally distributed. Scikit-learn provides many variations of such scaling. The most suited method for this project is found to be the Standard Scaler Scaling. Standard Scaler standardize features by removing the mean and scaling to unit variance. We have applied this scaler to our dataset.

Evaluation Metrics

As we have a Regression and a Classification problem, we would be having different metrics for both problems. For the Regression problem, we will be looking at R2 score and Root Mean Squared Error. We will make different models and compare those models based on both of those metrics. For the Classification problem, we will be comparing our various models through the Classification Report (Precision, Recall, Accuracy, F1 Score) and Confusion Matrix (False Positives, False Negatives, True Positives and True Negatives) and Cohen's Kappa Score.

Literature Review

Benchmark for Regression

After thorough searching we could only find two approach which has attempted this problem as a regression problem. This was attempted by Syed Sadiq Ali Naqvi. Here they have predicted the number of shares using the following models. Their corresponding RMSE and Mean Absolute error have been provided in the image below:

Model	Root Mean Squared Error	Mean Absolute Error
Random Forest Regressor (with 10 trees)	15144	3707
Random Forest Regressor (with 50 trees)	14724	3469
Linear Regression	14509	3154
Ridge Regression (learning rate=0.5)	14502	3116
Lasso	14502	3111
Bayesian Ridge	14503	3125
Ensemble of Bayesian Ridge, Ridge, Lasso and Linear Regression	**14501**	**3114**

Here as we can see that the best approach is ensemble of different methods which include Bayesian Ridge, Ridge, Lasso and Linear Regression which gives the RMSE of 14501.They have not mentioned their R2 score.

However, another approach done by Quan Yang and He Ren have mentioned that their R2 score was 20%, but they have not provided RMSE for it.

Benchmark for Classification

Previously, several machine learning experts have done their analysis to predict number of shares of an article in the Online News Popularity dataset given the input parameters. Likewise, many other media companies' successes depend on the popularity of articles and one of the key metrics to measure popularity is no. of shares done on article.

They firstly, apply the approach to find the threshold value for target variable and most of them observe that median (50%) have value 1400 that means balanced classes. Hence, 1400 is chosen as threshold value for share variable when share > 1400 that means it considered as popular share, when share <= 1400 that means it considered as unpopular by visualizing. Also, they visualize summary statistics of shares. Secondly, they normalize the numerical features by using all numerical value variable are taken into 'numerical' list and scaled by importing Min-Max-Scaler**.** They split the dataset into training and testing dataset and then they use many types of models and algorithm to accomplish the best accuracy like : Random Forest, Naïve Bayes(With PCA and Without PCA), Decision Tree, SVM, GridSearchCV(With PCA and Without PCA), XGBoost, KNN.

After observing all the plots and implemented all the models mentioned above, they predict that the classifier model of Random Forest has the highest accuracy on training set of 67% than the others classifier models. Random Forest is the better algorithm which can be used to predict whether the news article is popular or not.

Challenges Faced

There were majorly two challenges faced by us whilst analysing this dataset. We faced challenges when we were applying the MICE algorithm as it's a very complex algorithm and its working is very tough to understand. As MICE algorithm is considered to be the most effective and the best algorithm for null imputation in the data science industry, it needs high computation power hence it takes a very long time to execute it on our dataset as it has 61 columns. We tried to apply MICE on 61 columns, as it took several hours, we decided to apply on the columns which are not dummificated or in the range of 0 to 1 (24 columns in total). Another critical challenge we faced was about the R2 score and the mean squared error. Even after applying several models like OLS Regression model, Linear Regression, Ridge Regression, Lasso Regression, Gradient Boosting Regressor, XGBoost Regressor; the best R2 score we achieved was 13.8% with the OLS regression model.

Models Applied

We have applied several Regression and Classification models on our dataset as we mentioned in our problem statement such as Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, XGBoost and Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier, KNN Classifier, XGBoost Classifier.

Regression Models

Linear Regression is an approach for predicting a quantitative dependent variable y based on many independent predictors (x1, x2, x3…xn). It assumes there is a linear relationship between x and y.

In our project we applied linear regression two times, one at base model and one after backward elimination. We found that the R2 score was very low on base model as well as after applying backward elimination.

R2 Score on base model: 8.0%

R2 Score after backward elimination: -0.0033%

This is because our data doesn't follow assumptions of Linear Regression as shown below:

Summary of Assumptions:

1. For the residual plot there is a clear pattern between residuals and target column (shares). Therefore, this assumption is not fulfilled.
2. For checking the normality of the data, we applied Shapiro's test and found out that our p value was 0.0. Thus, we reject our null hypothesis and accept the alternate hypothesis which means our data is not normal. This assumption was also not fulfilled.
3. For checking the heteroscedasticity, we applied the Goldfeld quandt test. Here we got our p value as 9.027698342437254e-13, which means our data is heteroscedasticity and shows an increasing pattern
4. For checking the auto correlation in errors, we applied the Durbin-Watson test and it returned the value of 2.007. Therefore, there is almost no correlation between in errors. This assumption thus was fulfilled.
5. Data appears to be non-linear in nature as the rainbow test p value is 0.0026.

Thus, we see that our model doesn't meet many of the assumptions of the linear

Regression. Therefore, it is not preferable to apply it on this dataset. Thereafter, we applied ensemble models.

Models	R2 Score	Root Mean Squared Error
Linear Regression	**8.0**	**1000**
Decision Tree Regressor	**6.0**	**1010**
Random Forest Regressor	**5.30**	**1015**
Gradient Boosting Regressor	**14.30**	**1980**
XGBoost Regressor	**18.0**	**950**

Classification Models

Models	Accuracy (%)	F1-Score (Popular|Unpopular) (%)
Logistic Regression	**64.71**	**63.0|66.0**
Decision Tree Classifier	**57.78**	**57.0|58.0**
Random Forest Classifier	**61.19**	**63.0|59.0**
KNN Classifier	**57.42**	**56.0|59.0**
Gradient Boosting Classifier	**66.36**	**65.0|67.0**
XGBoost Classifier	**66.21**	**65.0|67.0**

Conclusion

From the above applied models and exploratory data analysis, it seems that the data is unevenly spread so our models are not able to fit properly which is responsible for low observed r-square values and root mean squared errors, hence the independent features are not able to explain the target column. Our classification accuracies are more inclined towards then benchmark accuracies rather than our regression accuracies. It would be better to treat the problem as a classification problem than a regression one.

Steps to Be Taken to Improve the Predictive Model

As we can see the R2 score and RMSE are not up to the mark, our next goal to focus on for now would be to increase these two metrics. So, we will still be exploring the data more for analysis to extract some more useful insights the dataset has to provide, and we will be focusing more on feature engineering. We will also be applying transformation. We are going to apply cross validation techniques and grid search for hyperparameter tuning which would help us get the optimum values by tuning our hyperparameters in our applied models which would give us better training and testing accuracies with low bias and low variance errors for both our problems(regression and classification). The interpretation of the R2 score

after applying PCA also is that our model explains 13.87% of the variance in the Shares of Online News Popularity. Also, increase/decrease principal components by tuning the hyperparameter which is n_components so that we can get better explained variance in the data which would give us better metrics for training and testing sets.

References

Archive.ics.uci.edu. (2019). *UCI Machine Learning Repository: Online News Popularity Data Set*. [online] Available at: https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity# [Accessed 30 Oct. 2019].

Azur, M., Stuart, E., Frangakis, C. and Leaf, P. (2012). *Multiple imputation by chained equations: what is it and how does it work?*.

Cs229.stanford.edu. (2019). [online] Available at: http://cs229.stanford.edu/proj2015/328_report.pdf [Accessed 30 Oct. 2019].

Medium. (2018). *Predicting popularity of Online News Articles — A Data Scientist's Report*. [online] Available at: https://medium.com/@syedsadiqalinaqvi/predicting-popularity-of-online-news-articles-a-data-scientists-report-fac298466e7 [Accessed 30 Oct. 2019].

Medium. (2019). *Understanding Principal Component Analysis*. [online] Available at: https://medium.com/@aptrishu/understanding-principle-component-analysis-e32be0253ef0 [Accessed 30 Oct. 2019].

Medium. (2019). *6 Different Ways to Compensate for Missing Values In a Dataset (Data Imputation with examples)*. [online] Available at: https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779 [Accessed 28 Oct. 2019].

Rdocumentation.org. (2018). *mice function | R Documentation*. [online] Available at: https://www.rdocumentation.org/packages/mice/versions/3.6.0/topics/mice [Accessed 28 Oct. 2019].

Techniques, B. (2015). *Beginners Guide To Learn Dimensionality Reduction Techniques*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2015/07/dimension-reduction-methods/ [Accessed 28 Oct. 2019].

Appendix

Data Dictionary

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

Attribute Information:
0. url: URL of the article (non-predictive)
1. timedelta: Days between the article publication and the dataset acquisition (non-predictive)
2. n_tokens_title: Number of words in the title
3. n_tokens_content: Number of words in the content
4. n_unique_tokens: Rate of unique words in the content
5. n_non_stop_words: Rate of non-stop words in the content
6. n_non_stop_unique_tokens: Rate of unique non-stop words in the content
7. num_hrefs: Number of links
8. num_self_hrefs: Number of links to other articles published by Mashable
9. num_imgs: Number of images
10. num_videos: Number of videos
11. average_token_length: Average length of the words in the content
12. num_keywords: Number of keywords in the metadata
13. data_channel_is_lifestyle: Is data channel 'Lifestyle'?
14. data_channel_is_entertainment: Is data channel 'Entertainment'?
15. data_channel_is_bus: Is data channel 'Business'?
16. data_channel_is_socmed: Is data channel 'Social Media'?
17. data_channel_is_tech: Is data channel 'Tech'?
18. data_channel_is_world: Is data channel 'World'?
19. kw_min_min: Worst keyword (min. shares)
20. kw_max_min: Worst keyword (max. shares)
21. kw_avg_min: Worst keyword (avg. shares)
22. kw_min_max: Best keyword (min. shares)
23. kw_max_max: Best keyword (max. shares)
24. kw_avg_max: Best keyword (avg. shares)
25. kw_min_avg: Avg. keyword (min. shares)
26. kw_max_avg: Avg. keyword (max. shares)
27. kw_avg_avg: Avg. keyword (avg. shares)
28. self_reference_min_shares: Min. shares of referenced articles in Mashable
29. self_reference_max_shares: Max. shares of referenced articles in Mashable
30. self_reference_avg_sharess: Avg. shares of referenced articles in Mashable
31. weekday_is_monday: Was the article published on a Monday?
32. weekday_is_tuesday: Was the article published on a Tuesday?
33. weekday_is_wednesday: Was the article published on a Wednesday?
34. weekday_is_thursday: Was the article published on a Thursday?
35. weekday_is_friday: Was the article published on a Friday?
36. weekday_is_saturday: Was the article published on a Saturday?
37. weekday_is_sunday: Was the article published on a Sunday?
38. is_weekend: Was the article published on the weekend?
39. LDA_00: Closeness to LDA topic 0

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

Raw Codes (Jupyter Notebook) – GitHub Link

https://github.com/Tarun-nano/Capstone_project