PROJECT REPORT ON

Popularity Prediction of Online News Articles

Submitted towards partial fulfilment of the criteria
For the award of PGPDSE by the Great Lakes Institute of Management

SUBMITTED BY

Group No. 9 [Batch: ONL AUG 2021-A]

GROUP MEMBERS

1. Likhitha Karumanchi (Team Lead)
2. Ruchika Madanmohan Polasa
3. Anish Narayanna
4. VeenaSri M
5. Raghavan Kannan
6. Pramod M A

RESEARCH SUPERVISOR

Mr. Srikar Muppidi

GREAT LAKES EXECUTIVE LEARNING

Great Lakes Institute of Management

ABSTRACT

Online platforms like Medium, Buzzfeed, Mashable and so forth distribute many articles regularly. These articles have a place with specific classes like entertainment, sports, technology, innovation, etc. This project aims to predict the popularity of an online news article before it is published. Such a model will help publishers and editors in amplifying the fame of their articles and selling advertisements.

News associations have progressively come to depend on media analytics as a method for drawing in and retaining readers. It has become evident with advertisement income falling this year, because of COVID-19 and other pre-pandemic patterns. It's become crucial for media organizations to know which news stories resonate with users, and which articles do not. Considering this, the aim is to discover what makes a news story popular or unpopular

- Techniques:
  - Predictive Modelling
  - Supervised Machine Learning Models
  - PCA
  - Feature Selection

- Tools:
  - Python
  - IDE - Jupyter Notebook

- Domain:
  - Data Science
  - Social Media Marketing

ACKNOWLEDGEMENT

We hereby certify that the work done by us for the implementation and completion of this project is original and to the best of our knowledge.

Date: 27-04-2022

Place: Online

CERTIFICATE OF COMPLETION

This is to certify that the project titled **“Popularity Prediction of Online News Articles”** for case resolution was undertaken and completed under the supervision of Mr. Srikar Muppidi for the Post Graduate Program in Data Science and Engineering (PGP – DSE)

Mentor: Mr. Srikar Muppidi

TABLE OF CONTENTS

ABBREVIATIONS

S. No.	Full-Form	Abbreviation
1.	Latent Dirichlet Allocation	LDA
2.	Inter quartile Range	IQR
3.	Exploratory Data Analysis	EDA
4.	Recursive feature Elimination	RFE
5.	Principal Component Analysis	PCA
6.	Logistic Regression	LR
7.	Random Forest	RF
8.	Cross-Validation	CV
9.	Area Under the ROC Curve	AUC

CHAPTER 1

1.1 EXECUTIVE SUMMARY

Consuming news articles is an integral part of our daily lives and news agencies expend tremendous effort in providing high-quality reading experiences for their readers. Journalists and editors are faced with the task of determining which articles will become popular so that they can efficiently allocate resources to support a better reading experience. The reasons behind the popularity of news articles are typically varied and might involve contemporariness, writing quality, and other latent factors.

In our project, we cast the problem of popularity prediction problem as classification, engineer several classes of features (metadata, contextual or content-based, temporal, and social), and build models for forecasting popularity. With the help of the Internet, online news can be instantly spread around the world. Most people now have the habit of reading and sharing news online, for instance, using social media like Twitter and Facebook. Typically, the news popularity can be indicated by the number of reads, likes, or shares. For the online news stakeholders such as content providers or advertisers, it’s very valuable if the popularity of the news articles can be accurately predicted before the publication. Thus, it is interesting and meaningful to use machine learning techniques to predict the popularity of online news articles. Various works have been done in the prediction of online news popularity.

In this project, based on the dataset including 39,643 news articles from the website Mashable, we will try to find the best classification learning algorithm to accurately predict if a news article will become popular or not before publication.

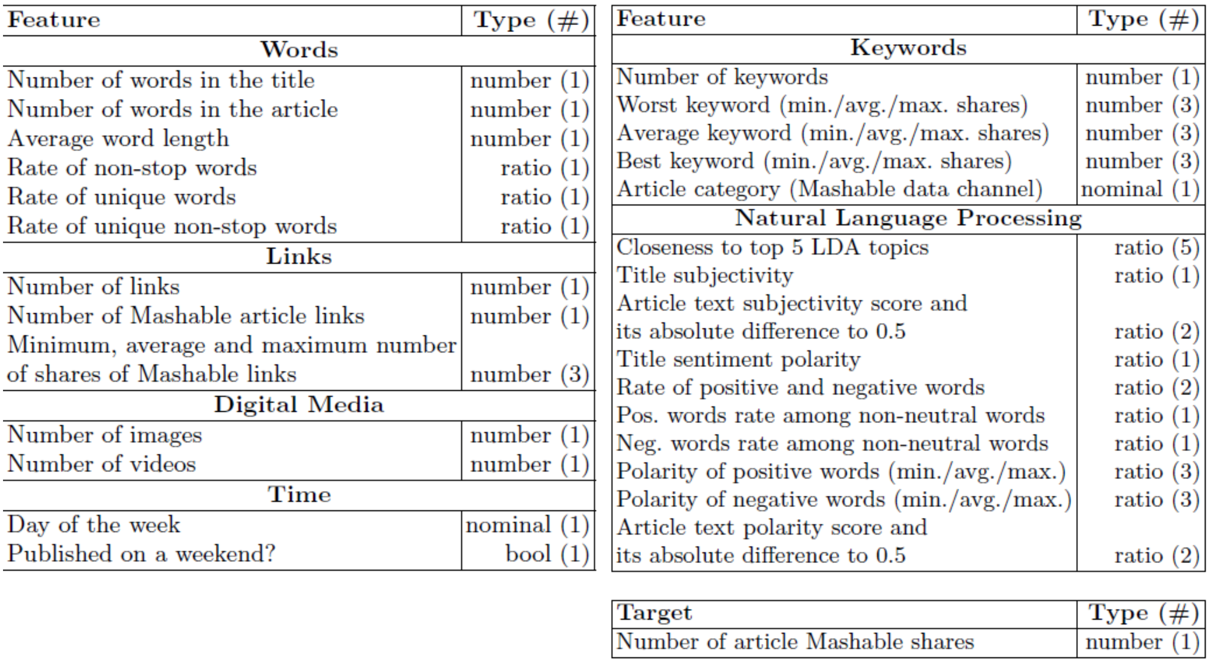
CHAPTER 2

2.1 DATA SET INFORMATION

The chosen dataset summarizes a heterogeneous set of features about articles published by Mashable in two years. The dataset consists of 39,643 news articles from an online news website called Mashable collected over 2 years from Jan. 2013 to Jan. 2015.

The data set used for the project is available at the UCI repository. 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 Table 1.

The features contain information about URLs, temporal data statistics, and sentiment analysis. The dataset has already been initially pre-processed. For example, the categorical features like the published day of the week and article category have been transformed by a one-hot encoding scheme.



Feature	Type (#)
Words	
Number of words in the title	number (1)
Number of words in the article	number (1)
Average word length	number (1)
Rate of non-stop words	ratio (1)
Rate of unique words	ratio (1)
Rate of unique non-stop words	ratio (1)
Links	
Number of links	number (1)
Number of Mashable article links	number (1)
Minimum, average and maximum number of shares of Mashable links	number (3)
Digital Media	
Number of images	number (1)
Number of videos	number (1)
Time	
Day of the week	nominal (1)
Published on a weekend?	bool (1)

Feature	Type (#)
Keywords	
Number of keywords	number (1)
Worst keyword (min./avg./max. shares)	number (3)
Average keyword (min./avg./max. shares)	number (3)
Best keyword (min./avg./max. shares)	number (3)
Article category (Mashable data channel)	nominal (1)
Natural Language Processing	
Closeness to top 5 LDA topics	ratio (5)
Title subjectivity	ratio (1)
Article text subjectivity score and its absolute difference to 0.5	ratio (2)
Title sentiment polarity	ratio (1)
Rate of positive and negative words	ratio (2)
Pos. words rate among non-neutral words	ratio (1)
Neg. words rate among non-neutral words	ratio (1)
Polarity of positive words (min./avg./max.)	ratio (3)
Polarity of negative words (min./avg./max.)	ratio (3)
Article text polarity score and its absolute difference to 0.5	ratio (2)

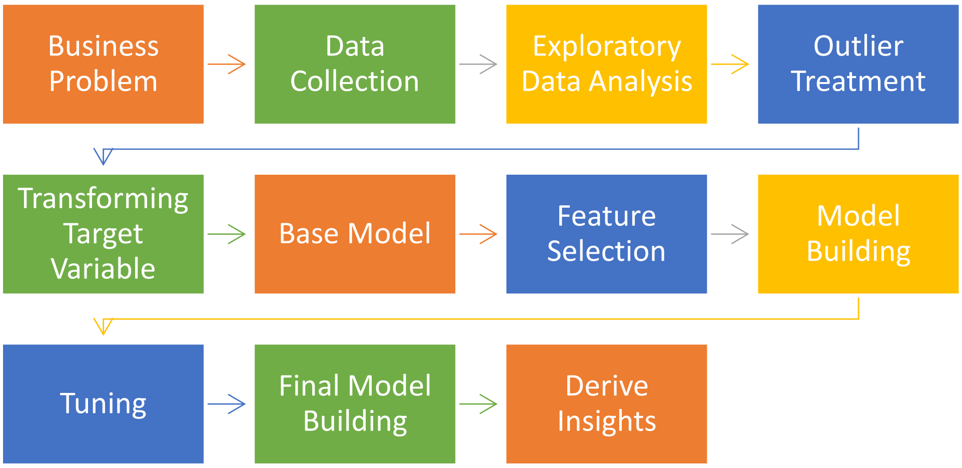
Target	Type (#)
Number of article Mashable shares	number (1)

Table 1: List of Features in the dataset

Data set sources:
https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity

https://in.mashable.com/

2.2 PROJECT FLOW



The process we chose to work on is an iterative process and is as below:

- We define the business problem.
- We collect data from different relevant sources.
- In Exploratory Data Analysis, we try to understand the structure of the data and the nature of the variables. The information is present in the data. Statistical summary of the variables. Special attention is paid to the target variable distribution.
- The other step of data preparation is Outlier Treatment. It is important to treat the outliers as they can affect the mean of the data in question and our evaluation of the data could be way off than the actual values.
- We built baseline models to get the idea of the minimum amount of performance that we can get from each of the models we have decided to use in the future.
- Feature Selection is a method to select features that have the maximum impact on the metrics of the models. The features which don’t influence the performance of the model much are dropped.
- Recursive model building is one of the key highlights of our methodology as we are going to build models, evaluate their performance, find out the best features hyperparameters behind the performance of the model and then rework the focus

areas and make our model better to meet benchmark accuracy which in real-world will be decided by the actual business partner.

- Final Model building will be done when we are equipped with the information as to what drives the performance of the models and what we want from our model, such as model interpretability, explainability, performance, etc.
- We can also derive insights from the selected features, like, as which features to focus on the most in the future.

2.3 DATA PREPROCESSING

Dropping Columns:

The data set has two attributes 'url' and 'timedelta' which denote the url of the published news article and the number of days between article publication and the data set acquisition. We can see that these two attributes do not possess any predictive power and cannot contribute to predicting the popularity of a new news article. So, these features have been eliminated from the data for model building.

Outlier Analysis

After removing the attributes mentioned earlier the data set has been checked for any potential outliers. A box plot has been created to visualize these outliers from a few attributes. It can be observed that most of the data is shown as outliers in the plot.

The news articles are checked for outliers using the 1.5*IQR rule i.e., values that are beyond Q3 + 1.5*IQR and values that are below Q1 - 1.5*IQR are considered outliers. A total of 37,533(i.e., 95% of the data) records are detected as outliers by applying this rule. Hence all the data shown as outliers in the plot cannot be considered erroneous data and should not be eliminated.

Converting negative values to positive:

There were a lot of negative values in the dataset and though the transformation techniques do not work on negative values we had to convert them to positive ones. We followed the below steps:

1. We identified the features with negative values.
2. We added a constant to the features to convert negative values to positive values.

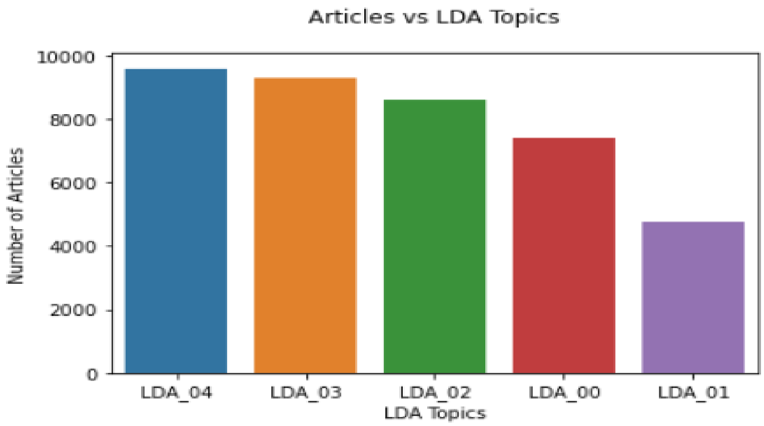
Transformation techniques:

Transformation techniques are used for transforming the data in the presence of outliers where we decide not to remove the outliers. We applied different transformation techniques like log transformation, square-root transformation & box-cox transformation. All the above box-cox transformations gave the best results.

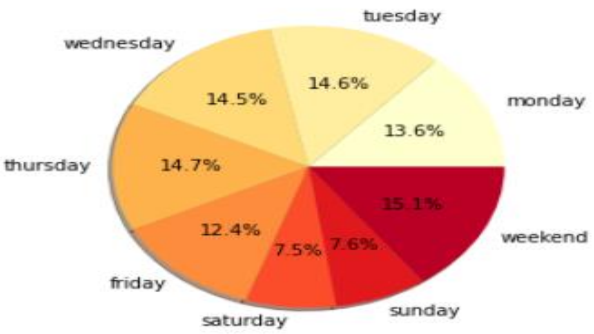
CHAPTER 3

3.1 EDA AND INSIGHTS

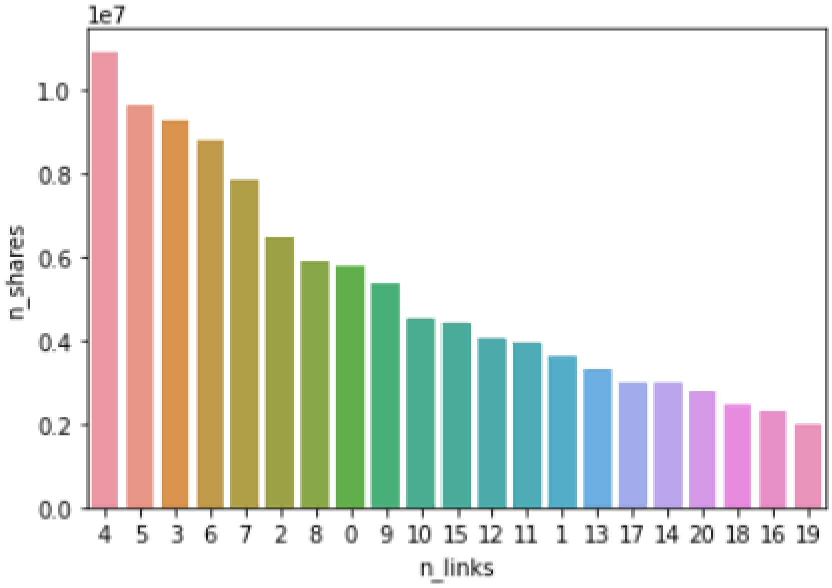
The following graphs show the relationship between different independent features and the target variable.



Weekday wise distribution of articles



Number of shares vs num_hrefs(Number of links)



INSIGHTS:

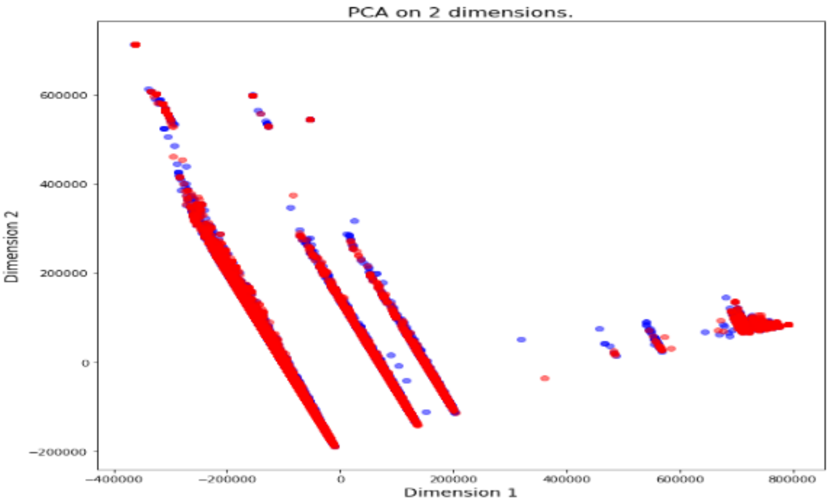
1. Higher number of articles are related to topic 4, whereas fewer articles are related to topic 1.
2. The articles with article categories such as business, technology, world, entertainment, and others are closely related to topic 0, topic 4, topic 2, topic 1, and topic 3 respectively.
3. More than 10% of the articles having article categories such as lifestyle, entertainment, social media, and others are slightly related to topic 4, topic 3, topic 0, and topic 1 respectively.
4. Monday, Tuesday, and Wednesday have almost equal shares.
5. Thursday, Friday, and weekend are the same shares.
6. Saturday and Sunday are equal shares, also equal to the weekend.

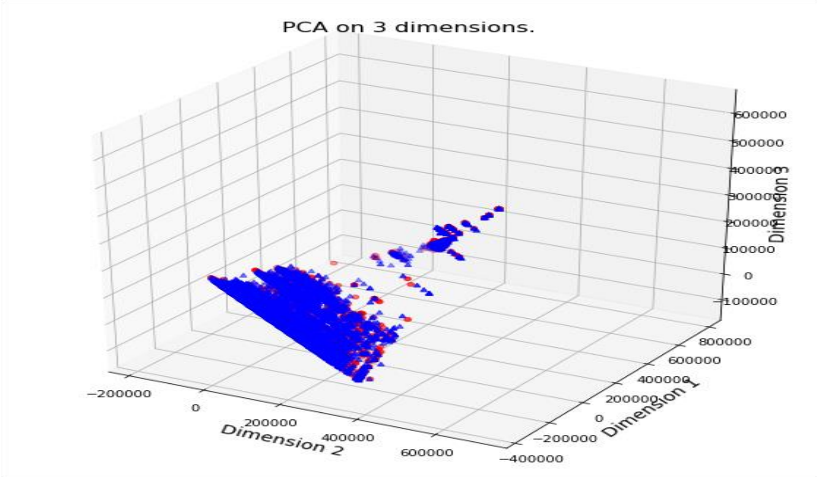
3.2 PCA

Next, we do the principal component analysis (PCA) to visualize the data. When there are many input attributes, it is difficult to visualize the data. There is a very famous term ‘Curse of dimensionality in the machine learning domain. It refers to the fact that a higher number of attributes in a dataset adversely affects the accuracy and training time of the machine learning model.

Principal Component Analysis (PCA) is a way to address this issue and is used for better data visualization and improving accuracy.

As shown below: project the data point onto the first 2 and 3 principal components, respectively. The dataset is not linearly separable in PCA space.

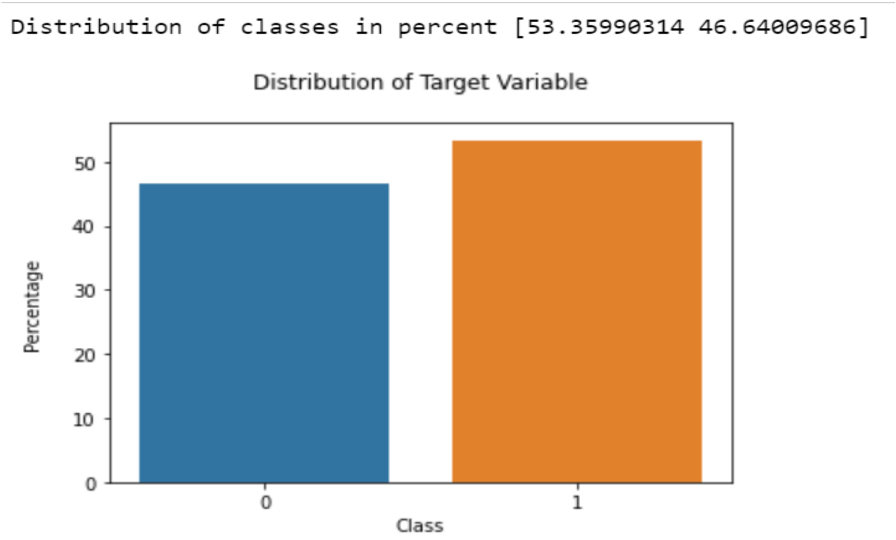




3.3 DISTRIBUTION OF TARGET VARIABLE

We transform our target variable from a discrete variable to a categorical variable. The first step to transforming our target variable to categorical variable is to check for "class imbalance".

We considered the median of the target attribute (number of shares) as an appropriate threshold to label all the data as either popular or unpopular. There is no class imbalance present in our data.



```
Distribution of classes in percent [53.35990314 46.64009686]
```

3.4 BASELINE MODEL BUILDING

Train-Test Split & Scaling:

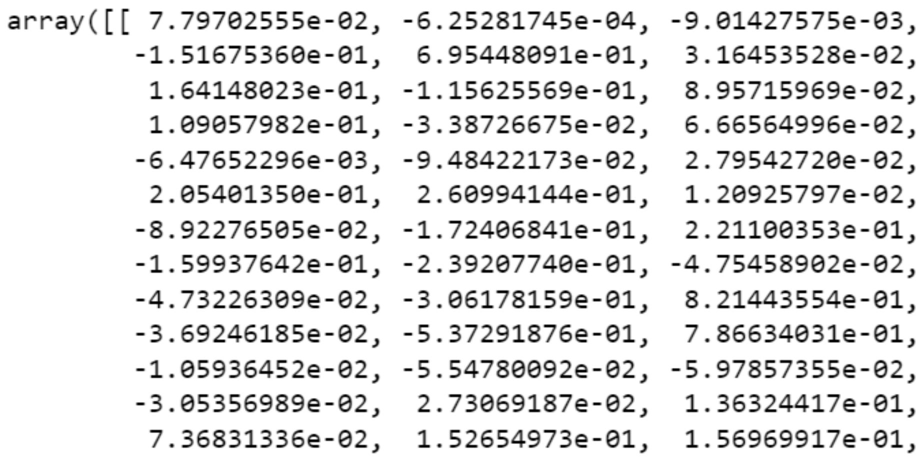
We need to split a dataset into train and test sets to evaluate how well our machine learning model performs. The train set is used to fit the model, and the statistics of the train set are known. The second set is called the test data set, this set is solely used for predictions. 80% of data has been used as a train set, while the remaining 20% was used as the test set.

Standard scalar from sklearn has been implemented to fit, transform train data and transform the test data.

Logistic Regression:

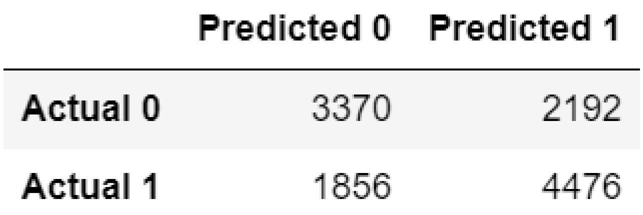
Logistic regression is easier to implement, interpret, and very efficient to train, it also makes no assumptions. Classification of the given data set is performed by categorizing the news into two categories, i.e., Popular and Unpopular based on the number of shares. Since the distribution for shares is right-skewed, the median of the shares attribute is taken as a threshold for classification. All the news articles having shares greater than 1400 are considered Popular news and those having shares lesser than 1400 are considered Unpopular news. Several classification algorithms have been applied to the data set and the following section summarizes the same.

We built a logistic regression model using stats models and sklearn learn packages. We got an accuracy score of 0.65 for our logistic regression model.



```
array([[ 7.79702555e-02, -6.25281745e-04, -9.01427575e-03,
        -1.51675360e-01,  6.95448091e-01,  3.16453528e-02,
         1.64148023e-01, -1.15625569e-01,  8.95715969e-02,
         1.09057982e-01, -3.38726675e-02,  6.66564996e-02,
        -6.47652296e-03, -9.48422173e-02,  2.79542720e-02,
         2.05401350e-01,  2.60994144e-01,  1.20925797e-02,
        -8.92276505e-02, -1.72406841e-01,  2.21100353e-01,
        -1.59937642e-01, -2.39207740e-01, -4.75458902e-02,
        -4.73226309e-02, -3.06178159e-01,  8.21443554e-01,
        -3.69246185e-02, -5.37291876e-01,  7.86634031e-01,
        -1.05936452e-02, -5.54780092e-02, -5.97857355e-02,
        -3.05356989e-02,  2.73069187e-02,  1.36324417e-01,
         7.36831336e-02,  1.52654973e-01,  1.56969917e-01,
```

A confusion matrix is used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

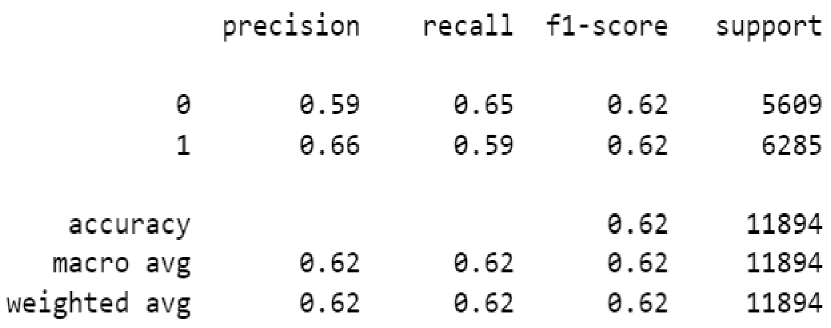


	Predicted 0	Predicted 1
Actual 0	3370	2192
Actual 1	1856	4476

Random Forest:

A Random Forest, which is an ensemble of several decision trees is built on the data set. By using several decision trees, overfitting is controlled allowing the performance metrics to be improved. The sub-samples of the given data set are fed into each decision tree. The size of the input fed into each decision tree is kept constant by setting the parameter bootstrap = True. So, the size of each subsample is the same, and sampling is done with replacement.

The below figure shows the accuracy and other performance metrics like precision, recall, and F1 score of our model.



```
              precision    recall  f1-score   support

           0       0.59      0.65      0.62      5609
           1       0.66      0.59      0.62      6285

    accuracy                           0.62     11894
   macro avg       0.62      0.62      0.62     11894
weighted avg       0.62      0.62      0.62     11894
```

AdaBoost:

AdaBoost is one of the first boosting algorithms to be adapted in solving practices. AdaBoost helps you combine multiple “weak classifiers” into a single “strong classifier”. The weak learners in AdaBoost are decision trees with a single split, called decision stumps. AdaBoost works by putting more weight on difficult to classify instances and less on those already handled well.

The three metrics (accuracy, F1-score, and AUC) are summarized in Fig 12. Under default parameter setting, Adaboost performs best in all three metrics, RF performs better than logistic regression in AUC while logistic regression performs better than RF in accuracy and F1-score. As for the training and testing speed, logistic regression is much faster than the other two, and RF runs faster than Adaboost.

Classifier	**Accuracy**	**F1-score**	**AUC**
Logistic Regression	0.63	0.69	0.64
RF	0.64	0.69	0.65
Adaboost	0.63	0.68	0.63

Figure: Metrics score of three classifiers including all features

3.5 FEATURE SELECTION PROCESS

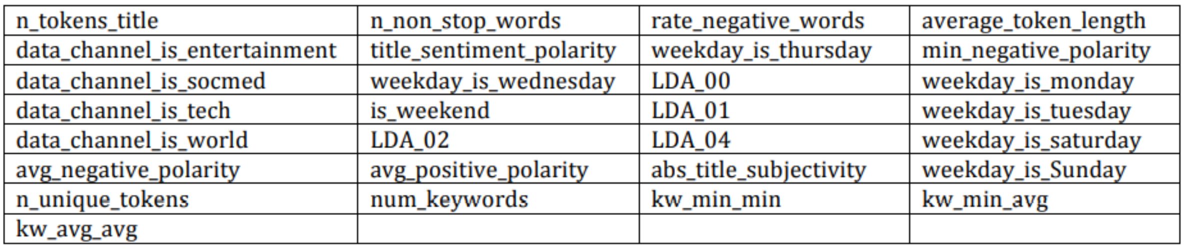
“***MLE can be defined as a method for estimating population parameters (such as the mean and variance for Normal, rate (lambda) for Poisson, etc.) from sample data such that the probability (likelihood) of obtaining the observed data is maximized”***.

We used Recursive Feature Elimination (RFE) to find the top-performing features affecting the target variables and then applied logistic regression to these variables to predict the target variables. RFE as its title suggests recursively removes features, builds a model using the remaining attributes, and calculates model accuracy.

Three benefits of performing feature selection before modelling your data are:

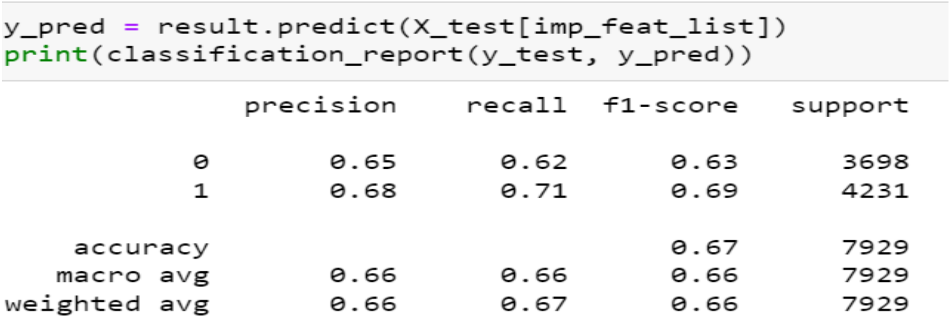
- *Reduces Overfitting*: Less redundant data means less opportunity to make decisions based on noise.
- *Improves Accuracy*: Less misleading data means modeling accuracy improves.
- *Reduces Training Time*: Fewer data means that algorithms train faster.

Since there are 58 features in the dataset, it is reasonable to conduct a feature selection to reduce the data noise, reduce the model complexity and improve the algorithm training time. One effective way is using recursive feature elimination with cross-validation (RFE) to automatically select the most significant features for certain classifiers. sklearn provides a function called RFE () that can help us. RFE algorithm selects 29 most relevant features from 58 original features. The selected 29 features are listed below. Both Logistic Regression and Random Forest models are built by taking the below 29 most relevant features.



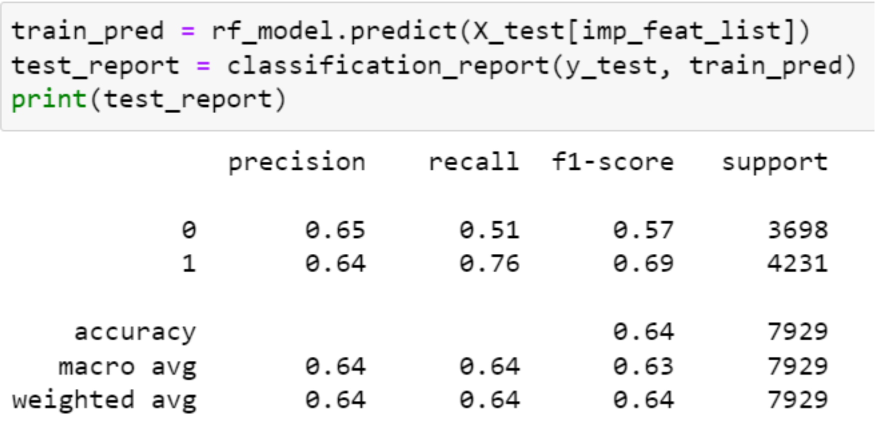
n_tokens_title	n_non_stop_words	rate_negative_words	average_token_length
data_channel_is_entertainment	title_sentiment_polarity	weekday_is_thursday	min_negative_polarity
data_channel_is_socmed	weekday_is_wednesday	LDA_00	weekday_is_monday
data_channel_is_tech	is_weekend	LDA_01	weekday_is_tuesday
data_channel_is_world	LDA_02	LDA_04	weekday_is_saturday
avg_negative_polarity	avg_positive_polarity	abs_title_subjectivity	weekday_is_Sunday
n_unique_tokens	num_keywords	kw_min_min	kw_min_avg
kw_avg_avg			

Table: Features extracted with RFE



```
y_pred = result.predict(X_test[imp_feat_list])
print(classification_report(y_test, y_pred))
              precision    recall  f1-score   support

           0       0.65      0.62      0.63      3698
           1       0.68      0.71      0.69      4231

    accuracy                           0.67      7929
   macro avg       0.66      0.66      0.66      7929
weighted avg       0.66      0.67      0.66      7929
```

Figure: Classification Report for Logistic Regression (RFE)



```
train_pred = rf_model.predict(X_test[imp_feat_list])
test_report = classification_report(y_test, train_pred)
print(test_report)
```

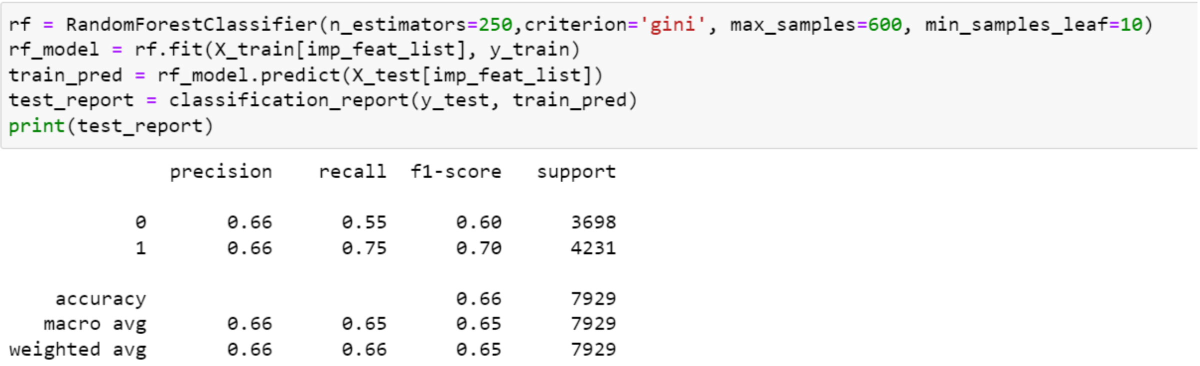
```
              precision    recall  f1-score   support

           0       0.65      0.51      0.57      3698
           1       0.64      0.76      0.69      4231

    accuracy                           0.64      7929
   macro avg       0.64      0.64      0.63      7929
weighted avg       0.64      0.64      0.64      7929
```

Figure: Classification Report for Random Forest (RFE)

3.6 HYPER PARAMETER TUNING

In this part, we implemented a grid search method for the random forest to refine their hyperparameters. The grid search method exhaustively searches through all possible combinations of model parameters, cross-validates the model, and then determines which set of model parameters gives the best performance. Since the grid search can help select an optimal model parameter, thus the learning algorithm can be optimized. We used the function GridSearchCV () from sklearn to implement hyper-parameter tuning.

After running the grid search using multiple combinations of parameters under param grid, the optimal hyperparameters obtained for RF are “n estimators”: 250, “criterion”: Gini, “max_samples”:600, “min_samples_leaf”:10. Now we run the two classifiers with refined parameters, the three metrics scores are summarized below Compared with the previous metrics, we can find the metrics of random forest are significantly improved after tuning by grid search.



```
rf = RandomForestClassifier(n_estimators=250,criterion='gini', max_samples=600, min_samples_leaf=10)
rf_model = rf.fit(X_train[imp_feat_list], y_train)
train_pred = rf_model.predict(X_test[imp_feat_list])
test_report = classification_report(y_test, train_pred)
print(test_report)
```

```
              precision    recall  f1-score   support

           0       0.66      0.55      0.60      3698
           1       0.66      0.75      0.70      4231

    accuracy                           0.66      7929
   macro avg       0.66      0.65      0.65      7929
weighted avg       0.66      0.66      0.65      7929
```

Figure: Classification Report for Random Forest (RFE) with tuned parameters.

CHAPTER 4

4.1 BENCHMARK

Benchmark for this project, we can take the dataset's donator's work as a benchmark model. By carefully tuning the hyperparameters, use the RF model to achieve a 0.67 accuracy score, 0.69 F1 score, and 0.73 AUC score.

4.2 MODEL EVALUATION & SELECTION

After initial implementation and further refinement of these three classifiers, we find the best performance is obtained by the RF classifier with 600 trees in the forest. The best-obtained metrics of RF are accuracy of 0.6769, F1-score 0.7073, and AUC of 0.71. The final scores are not exceptional, which is sort of within the expected range, because the dataset is not linearly separable as shown in the PCA in Chapter 2. But it still achieved a reasonable performance in news popularity prediction compared with a random guess.

To test the **robustness of the model,** we implemented 10-Fold Cross-Validation. Now the metrics shown below are almost similar to the previous table and the best performance is still given by RF.

	Precision (Class1)	**Recall (Class1)**	**F1-Score (Class1)**	**Accuracy**	**AUC**	**Train Score**	**Test Score**
Logistic Regression	67	70	68	65	66	66	65
Random Forest	66	**75**	**70**	**67**	**71**	**67**	**67**
Adaboost	**68**	71	69	**67**	67	**67**	66

Figure: Final Metrics score of three classifier models after Tuning

4.3 METHOD OVERVIEW

A stepwise process has been followed in the implementation of the supervised learner.

1. Data collection: The dataset of some 40,000 online news articles is downloaded from the UCI Machine Learning Repository, which is originally collected and donated by the author of [4].
2. Once the data is ready, it has been imported as a panda's data frame and the structure of the data is noted.
3. Data pre-processing: Based on the initial data pre-processing in the original dataset, A recursive approach has been followed while checking the data for missing values, variable data types, and outliers.
4. Exploratory data analysis has been performed including the univariate and bivariate analyses for all the variables as far as possible.
5. Data has been visualized using different charts including distance plots, histograms, bar charts, and box plots. Five-point summary statistics and correlation of data are also noted.
6. Further processed the dataset by normalizing the numerical feature such that each feature is treated equally when applying supervised learning.
7. The target variable has been completely analysed. Initially, our target variable is discrete but we transformed it into a categorical feature of 2 classes. We considered the median of the target variable (number of shares) as an appropriate threshold value to label all the data as either popular or unpopular. Post that, we observed no class imbalance present in our data.
8. Data exploration and visualization: Explored the relevance of a certain feature by visualization and also visualized the data distribution by PCA.
9. Feature selection: To select the most relevant features among all 58 features, we implemented RFE.
10. Classifier implementation and refinement: Multiple classification models including Logistic Regression, Decision Trees, Random Forest, AdaBoost Classifier, etc. are implemented recursively in the process to compare the results of each. Then the best performing model's hyperparameters are tuned by the grid search method.
11. Model evaluation and validation: The refined models are evaluated and compared using three metrics (accuracy, Precision, Recall, F1-score, AUC) and models have been cross-validated using K-fold Cross-Validation to learn a highly sensitive classifier that maps or classifies the data to 'popular' or 'unpopular' with high accuracy.

4.4 LIMITATIONS

Although the classifier gives a highly accurate result, there are certain limitations to the project:

- In data science, prediction models can never be 100% accurate. For example, in our dataset, it could be because the unidentified features (e.g. by using Feature extraction, thorough implementation of Feature selection techniques) which we fail to gather or observe may also contribute in enhancing true classification of the popularity of articles.
- Our model uses data such as LDA topics, subjectivity, polarity, and sentiment polarity which are obtained from NLP modeling. A brief understanding of these topics is necessary to interpret the results.
- Despite being highly relevant data since most of our data is a set of binary features, thus we see less scope for better visualization and bivariate analysis in terms of categorical variables.
- Statistical measures such as Pearson's correlation coefficient matrix to find out multicollinearity among variables were not applicable in the current dataset as they can only be used for continuous variables and not categorical.

4.5 BUSINESS FINDINGS

- In our dataset we have 58 independent features out of all these features below list shows the important features which contribute to the popularity of an article using RFE feature selection technique.

num_hrefs	kw_min_max
num_self_hrefs	kw_max_max
num_imgs	kw_max_avg
num_videos	kw_avg_avg
num_keywords	self_reference_min_shares
data_channel_is_entertainment	self_reference_max_shares
data_channel_is_socmed	self_reference_avg_sharess
data_channel_is_tech	weekday_is_friday
kw_min_min	weekday_is_saturday
kw_max_min	weekday_is_sunday
kw_avg_min	is_weekend

Table: List of features that contributes to the Popularity of article

Note: Please also be noted that other Feature selection techniques like VIF, SFS have implemented but didn't result in encouraging model performance results.

- The below graph shows how weekdays play a key role in the popularity of articles. Most of the articles are released on weekdays, hence the count of popular and unpopular articles is the same. As fewer articles are being released on weekends, the chance of the articles being popular is more.

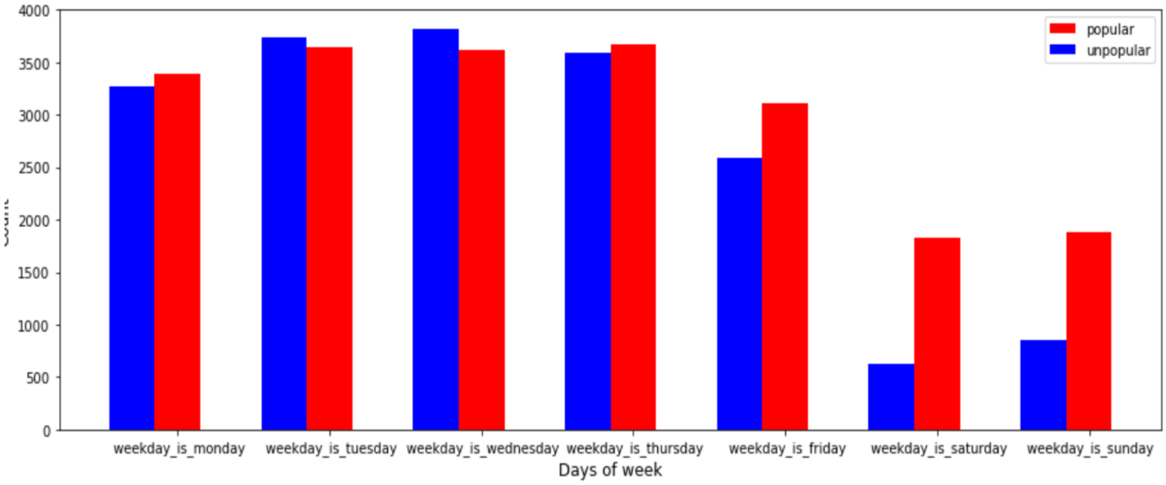


Figure: popular/unpopular articles over different days of the week

4.6 KEY RECOMMENDATIONS

- Our best model gave us a better understanding of which variables were most and least significant for shared articles. Based on variable significance (chapter 5 – Business Findings), we believe that businesses should create the fewer world, business, and entertainment genre-based articles. We also recommend the business create more articles with genres under social media, tech, lifestyle, and similar categories would help increase their shares.
- The results of our Random Forest model also focus on keyword strength. Based on the feature significance, we believe that having popular keywords is crucial to having a popular article.
- Another important factor of successfully shared articles results also focuses on the popularity of shared articles on a weekend based on the relevant percentage of published articles.

- The above recommendations to businesses are made based on assuming the goal of maximizing shares on social media.

We also recognized that making more social media articles may not translate to more total views or profit.

4.7 SCOPE OF IMPROVEMENT

To further improve model performance, we believe there are few possible ways:

- Increase the size of the dataset since RF has a strong learning capability and a rich dataset might improve its prediction performance
- Although it might increase the training time, more advanced cross-validation methods can be used.
- Implement feature engineering and feature extraction to add only relevant features from our original dataset. For instance, we could use all the words in an article as additional features using NLP techniques, and then try the classification models such as Naive Bayes to see if they can help in achieving better performance.

5. BIBLIOGRAPHY

1. A. Tatar, J. Leguay, P. Antoniadis, A. Limbourg, M. D. de Amorim, and S. Fdida, “Predicting the popularity of online articles based on user comments,” in Proceedings of the International Conference on Web Intelligence, Mining and Semantics. ACM, 2011, p. 67.
2. E. Hensinger, I. Flaounas, and N. Cristianini, “Modelling and predicting news popularity,” Pattern Analysis and Applications, vol. 16, no. 4, pp. 623–635, 2013.
3. S. Petrovic, M. Osborne, and V. Lavrenko, “Rt to win! predicting message propagation in Twitter.” ICWSM, vol. 11, pp. 586–589, 2011.
4. 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.
5. Bandari, R.; Asur, S.; and Huberman, B. A. The pulse of news in social media: Forecasting popularity. CoRRabs/1202.0332, 2012.
6. Tsagkias, M.; Weerkamp, W.; and De Rijke, M.Predicting the volume of comments on online news stories. In Proceedings of the CIKM’09, 1765–1768, 2009.

Articles:

- Arnott, D., Pervan, G.: Eight key issues for the decision support systems discipline. Decision Support Systems 44(3), 657–672 (2008)CrossRefGoogle Scholar
- Michalewicz, Z., Schmidt, M., Michalewicz, M., Chiriac, C.: Adaptive business intelligence. Springer (2006)Google Scholar
- Ahmed, M., Spagna, S., Huici, F., Niccolini, S.: A peek into the future: predicting the evolution of popularity in user-generated content. In: Proceedings of the sixth ACM international conference on Web search and data mining, pp. 607–616. ACM (2013)Google Scholar
- Bandari, R., Asur, S., Huberman, B.A.: The pulse of news in social media: forecasting popularity. In: ICWSM (2012)Google Scholar
- Szabo, G., Huberman, B.A.: Predicting the popularity of online content. Communications of the ACM 53(8), 80–88 (2010)CrossRefGoogle Scholar