Predicting and Evaluating the Popularity of Online News

ARCHANA SINGH 668528470
CHAITRA SRIRAMA 674121942
VIHARIKA BHARTI 655974244

ABSTRACT—Social Media has greatly transformed the ways in which individuals consume news. According to a recent survey conducted by Pew Research Center, 89% of Americans choose to read at least some extent of the news online. Furthermore, democratization of the content creation process allows more individuals to become creators and distributors of online news. However, the ease of generating and sharing news online produces copious amount of information which consecutively intensifies the competition to gain enough user attention. Hence it is very essential to understand what makes one news article more popular than other. Popularity can typically be indicated by the number of shares. Such prediction before publication can help the publishers to increase revenue by optimizing the articles further. Thus, we chose to work on this project which aims at predicting the popularity of an online news article. In this paper, we have implemented different machine learning algorithms to forecast the number of shares associated with an online news article. The performance of these algorithms is then compared to select the best model for prediction. Our work can help the publishers to anticipate popularity before publication and accordingly take measures to maximize user engagement.

KEYWORDS—Online News Popularity, Popularity Prediction, Machine Learning, Classification, Feature Selection, Model Selection

I. INTRODUCTION

Hundreds of articles are published daily on social media. Traditional methods such as search engine optimization with relevant keywords, images, videos, and exciting content has been used to increase the popularity of online news articles. However, we wish to achieve a deeper understanding as to why some articles are more popular than others by trying to analyze a variety of factors such as category, day of publication, internal content such as title, keywords, images etc., and certain natural language processing parameters. Using this we aim to predict the popularity of online news articles even before they are published. Popularity prediction is beneficial across different sectors like business, marketing, online advertising, and recommendation systems as online visitors prefer reading and sharing the most popular articles which can possibly impact the general interests and opinions of people.

This paper has been arranged in the following structure. Section II presents the existing work of online news popularity prediction while Section III describes our dataset alongside presenting the feature selection process. Section IV lists the prediction methodologies that have been used to forecast the popularity of online news articles. The analysis and comparison of the implemented machine learning algorithms has been summarized in Section V. Finally, the conclusion is presented in Section VI and Future Scope is presented in Section VII.

II. RELATED WORK

Several researchers have explored this engaging topic to analyze the popularity of articles well before its publication. But one of the most famous experimentation has been carried out by Kelwin Fernandes, Paulo Cortez and Pedro Vinagre [1]. These researchers implemented a unique approach called the Intelligent Decision Support System (IDSS). This system first analyzes the article to predict its popularity and then uses techniques to identify a set of article features that can be optimized

further to maximize the popularity. After successful implementation of the project, the researchers donated the gathered data to the UCI Machine Learning Repository.

Additionally, Tatar, Alexandru, Panayotis Antoniadis, Marcelo Dias De Amorim, and Serge Fdida [3] explored the subject of predicting the popularity of online news articles by analyzing the comments from the users. They collected the data from a French online news platform to precisely rank articles based on their predicted popularity. With respect to the ranking performance, simple linear regression proved out to be the best machine learning algorithm.

Arapakis, Ioannis, B. Barla Cambazoglu, and Mounia Lalmas [4] conducted a similar study on predicting the popularity of cold start news. They acquired the articles from Yahoo news and measured the popularity based on metrics like tweet counts and page views.

III. DATASET & FEATURE SELECTION

A. DATA ACQUISITION

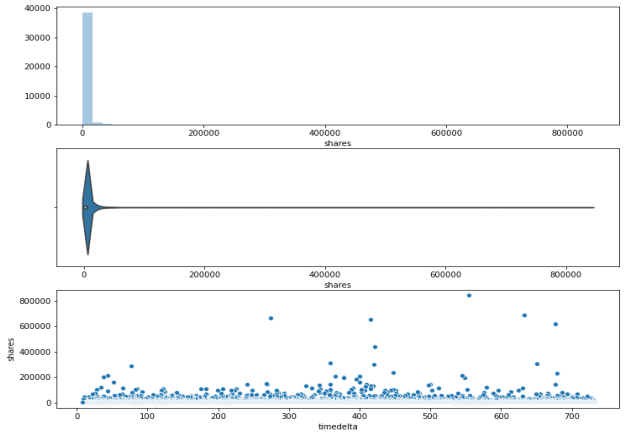
This dataset has been acquired from UCI Machine Learning Repository and originally belongs to Mashable Inc which is one of the largest news websites. The data was collected for a two-year period from 2013 to 2015. It has a total of 39644 rows (news articles) and 61 attributes (59 as numerical values) which describe the diverse set of aspects for each article and are perhaps considered relevant to impact the number of shares. The complete feature set has been presented in a table in the Appendix.

B. DATA EXPLORATION

Before beginning with any data preprocessing, the most common step is to explore the data to learn more about any existing trends in the data. Such exploration will also help us to detect any missing values and outliers in the data. This analysis will thus help us prepare the data accurately for modeling.

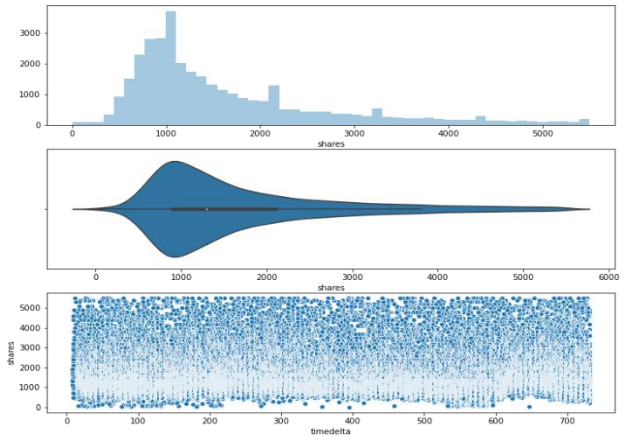
1) Understanding the Distribution of the Target Variable 'Shares' to Convert the Problem into a Binary Classification Problem

The summary statistics of the target variable show that the distribution is highly skewed. We have found the outliers using the 1.5*IQR rule. All the instances that fall over the 1.5*IQR third quartile have been considered as outliers and thus have been dropped from the dataset. The improvement in distribution after dropping the outliers can be seen in the graphs below.

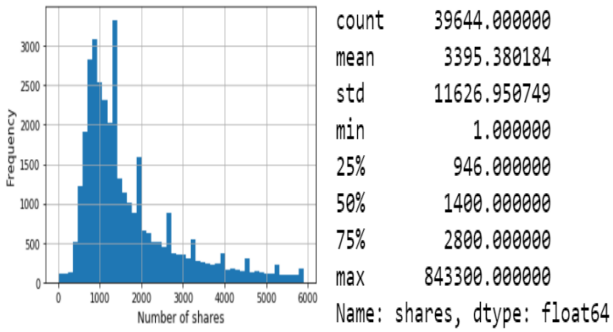


Graph 1: Target Variable Distribution with Outliers

After removing the outliers, the size of our dataset is (35103, 62). Since this is still a skewed distribution, we consider the median value which is 1400 as the threshold to decide whether an article is popular or not popular, thereby converting the problem into one with binary classification.

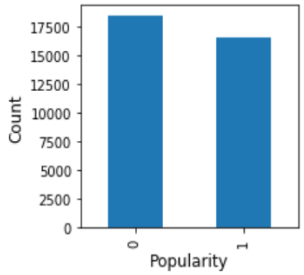


Graph 2: Target Variable Distribution after Removing Outliers



Graph 3: Frequency Distribution Graph of Number of Shares

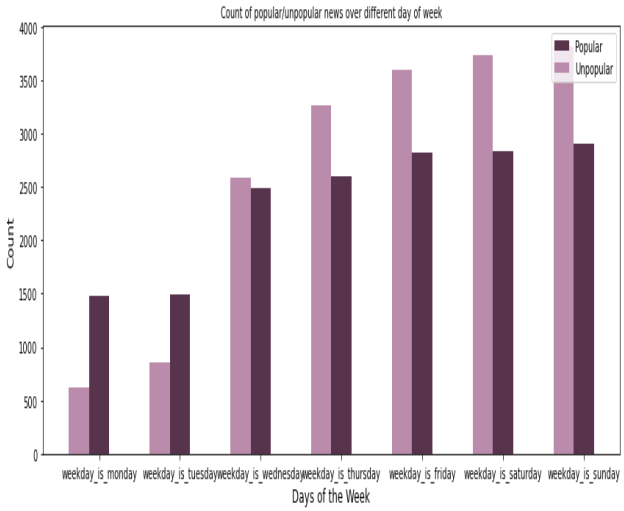
With this threshold of 1400, we see the distribution of articles to be approximately balanced with 16613 popular articles and 18490 not popular articles.



Graph 4: Distribution of Popular/Unpopular Articles

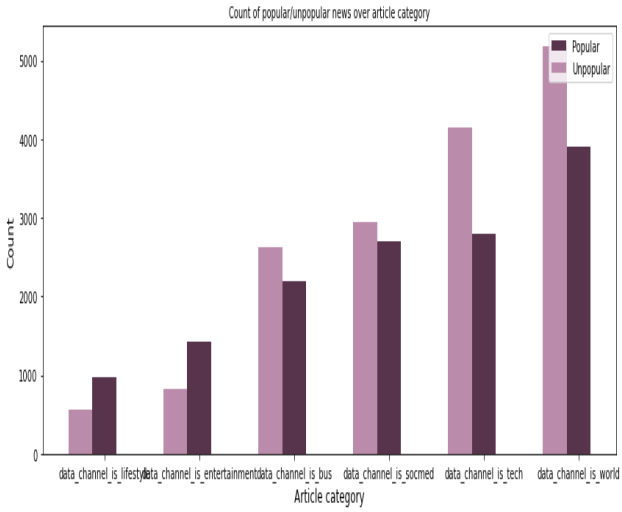
2) Popularity based on Days of the Week and Article Category

The most obvious features to look at are the article categories and the days of the week when the article was published.



Graph 5: Distribution of Popular/Unpopular Articles over Days of Week

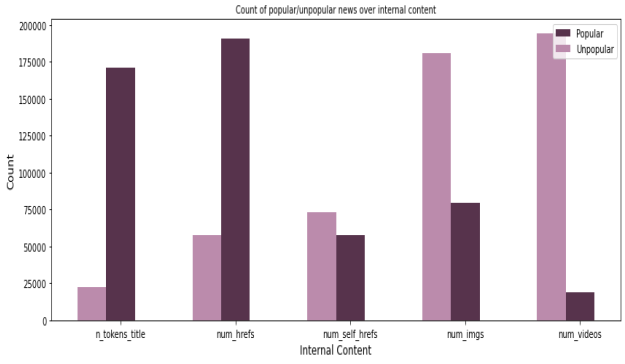
We see that the articles published over the weekends have more potential to be popular. This is probably due to the fact that the people are more likely to spend more time online browsing the news over the weekends than on weekdays, however there are more number of articles published on the weekdays since they are the working days of the week and thus there is not much differentiation between the popularity of these articles.



Graph 6: Distribution of Popular/Unpopular Articles over Article Categories

With respect to the category of the article, lifestyle and entertainment have a larger proportion of popular news than unpopular ones, and it is vice versa in the case of other categories. This reflects that the readers of Mashable prefer lifestyle and entertainment channels more over other categories.

3) Popularity based on Content of the Article

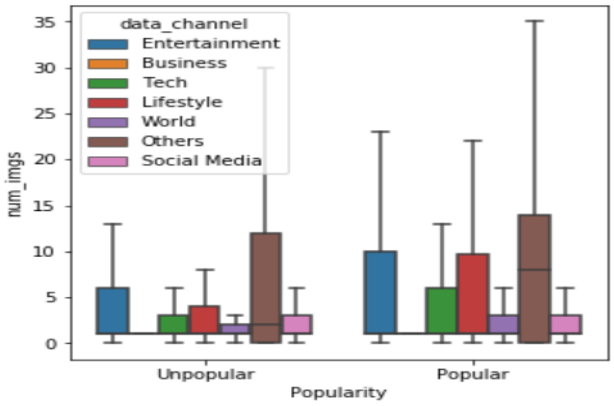


Graph 7: Distribution of Popular/Unpopular Articles over Article Content

Features of the article such as the number of tokens in the title, presence of images and videos, external links etc. also influence a reader's ability to read the article. We observe that the articles with large numbers of tokens and external links are more popular. This is because the title leaves the first impression on the article. On the other hand, we see that links to Mashable articles are neutral. We also notice that as the number of images and videos increases, the article is more likely to be unpopular.

4) Popularity based on Number of Images and Data Channel

We have compared the impact of number of images on different data channels for article popularity. It has been observed that popular articles tend to have more images than less popular articles. Good visuals make an article more interesting and easier to understand. But in the Business channel, the number of images does not have any impact on popularity. Other channels like entertainment, lifestyle and tech tends to be more popular with high image content in the articles.



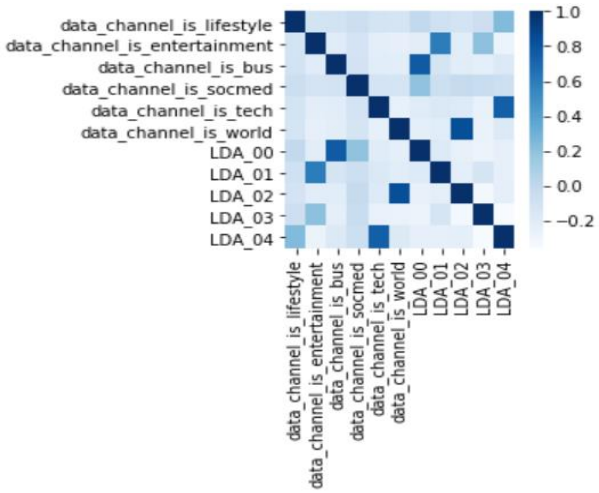
Graph 8: Distribution of Popular/Unpopular Article Images over Data Channels

5) Correlations between Variables to Develop Relationships between Relevant Features

When we analyzed the correlations between all the variables in the dataset, we observed that there were no high correlations between the features and the target variable and hence the popularity of the article is not highly dependent on a single variable in the dataset.

Some of the other obvious correlations observed were between the LDA topics and article category content which gives us some intuition about the LDA topics being related as follows:

LDA_00 Topic	Business
LDA_01 Topic	Entertainment
LDA_02 Topic	World
LDA_03 Topic	No High Correlations with Article Categories
LDA_04 Topic	Technology



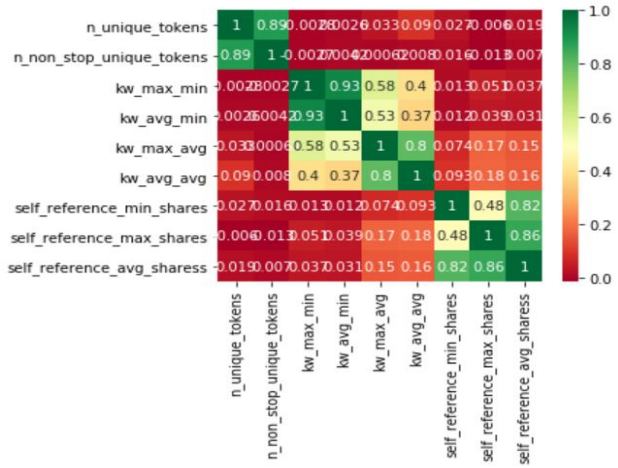
Graph 9: Correlation amongst Feature Set

C. DATA PRE-PROCESSING

Data preprocessing is a very essential as in this step we transform the data to bring it in a state that the machine can easily parse. Alternatively, it can also be stated as a step wherein we make the features of the dataset easily interpretable by the machine learning algorithm.

1) Data Cleaning

We began with the data cleaning process by dropping the non-predictive columns like ‘url’ and ‘timedelta’ as they do not contribute to the predictive ability of the models. Similarly, the variable ‘n_non_stop_words’ has also been dropped as it does not provide us with much intuition to predict the target variable. Additionally, we also dropped rows wherein the value of ‘n_tokens_content’ column was equal to zero. The reason behind dropping these rows is that the articles described through these records do not contain any words.



Graph 10: Correlation Matrix

Having performed a check for any NA or null values, we observed that the dataset was already acquired in a cleaned and processed state in this aspect. While analyzing the correlation between features, we considered a threshold of 0.8 to remove only those variables that are highly correlated. Below mentioned is a list of variables that have been eliminated due to high correlation.

'n_non_stop_unique_tokens'
'kw_avg_min'
'kw_max_avg'
'self_reference_min_shares'
'self_reference_max_shares'

2) Data Scaling

The numerical features of the dataset were skewed individually, and each numerical feature had a different range of values. Hence, we have used Robust Scaler to remove the outliers and MinMaxScaler to rescale the dataset such that all the feature values are in the range [0,1].

3) Dimensionality Reduction (Principle Component Analysis)

Dimensionality reduction involves reducing the number of features to achieve efficient modelling. It can be categorized into feature selection and feature extraction. Feature selection involves selecting important features for modelling whereas feature extraction is a process that transforms high dimensional data into fewer dimensions by deriving new features from the original ones to remove redundant and irrelevant attributes. Therefore, dimensionality reduction can be used to train the models faster and increase their accuracies by reducing over-fitting.

Principal components analysis (PCA) is an unsupervised linear feature extraction technique designed to reduce the size of the data by extracting information into new features known as Principal Components. These principal components are treated as new attributes and used to develop models. Generally, the principal components have better explaining power than the individual attributes. The variance ratio gives a measure of how much information is retained in the principal components.

After performing PCA on different group of variables to decrease the dimensions of the features, we observed that PCA was most effective for keyword and NLP related variables.

Variables	PCA Output Analysis	PCA Outcome
LDA_ [00-04]	By selecting 4 principal components out of 5, we can preserve 100% of the information of the total variance of the LDA variables	Since a significant number of features has not been reduced, PCA is not an effective method for this set of variables.
Keyword related Variables (kw_min_min , kw_max_min, kw_min_max, kw_avg_max, kw_min_avg, kw_avg_avg)	By selecting 5 principal components out of 6, we can preserve 97% of the information of the total variance of the keyword variables.	We have used the 5 Principal components to replace the original set of variables in this case.

Variables	PCA Output Analysis	PCA Outcome
NLP set of variables related to the subjectivity, sentiment polarity, presence of positive and negative words in the article and the title of the article	By selecting 11 principal components out of 16, we can preserve 99% of the information of the total variance of the NLP set of variables.	Since there is a significant decrease in the features, PCA is effective for this set of variables.

IV. PREDICTION METHODOLOGIES

This section elaborates on the different machine learning models that have been implemented in this project. This section will give us a better understanding of the functionality of the models before we could go ahead and comprehend their performance in the next section.

A. NAÏVE BAYES MODEL

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship, given class variable y and dependent feature vector x_1 through x_n

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n \mid y)}{P(x_1, \ldots, x_n)}$$

Using the naive conditional independence assumption that

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y),$$

for all i, this relationship is simplified to

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

Since P (x_1, …, x_n) is constant given the input, we can use the following classification rule:

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

B. K NEAREST NEIGHBORS (KNN) MODEL

KNN is a non-parametric and a lazy algorithm. It uses the dataset to predict and classify the new sample points. It is considered non-parametric because it does not make any assumption about the underlying data. This is especially useful is classifying the "real world" data as most of the data does not follow typical theoretical assumptions. The method of k-nearest neighbors makes very mild structural assumptions: its predictions are often accurate but can be unstable.

Nearest-neighbor methods use those observations in the training set which is the closest in input space to x. Specifically, the k-nearest neighbor fit is defined as follows:

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} y_i,$$

KNN performs classification of the data points (where output is a class membership) by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors.

C. LOGISTIC REGRESSION MODEL

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.

In order to map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

$$f(x) = \frac{1}{1+e^{-(x)}}$$

The cost function represents optimization objective i.e., we create a cost function and minimize it so that we can develop an accurate model with minimum error.

$$J(\theta) = -\frac{1}{m}\sum\left[y^{(i)}\log(h\theta(x(i))) + \left(1-y^{(i)}\right)\log(1-h\theta(x(i)))\right]$$

Ridge regression uses L2 regularization which adds the following penalty term to the OLS equation.

$$+\lambda\sum_{j=0}^{p} w_j^0$$

Lasso regression uses the L1 penalty term and stands for Least Absolute Shrinkage and Selection Operator.

$$+\lambda\sum_{j=0}^{p} |w_j|$$

Elastic Net - A third commonly used model of regression is the Elastic Net which incorporates penalties from both L1 and L2 regularization:

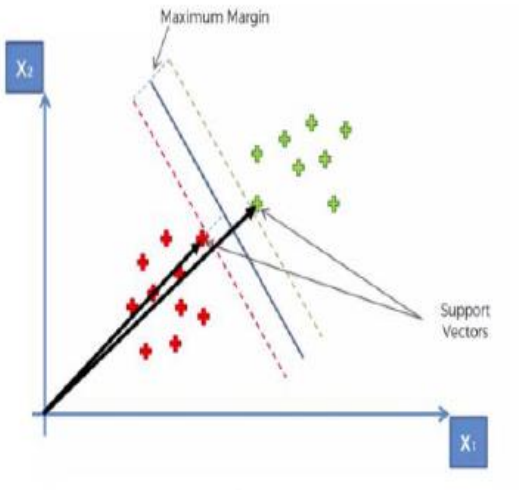
$$\frac{\sum_{i=1}^{n}(y_i - x_i^J\hat{\beta})^2}{2n} + \lambda\left(\frac{1-\alpha}{2}\sum_{j=1}^{m}\hat{\beta}_j^2 + \alpha\sum_{j=1}^{m}|\hat{\beta}_j|\right)$$

Elastic net regularization - In addition to setting and choosing a lambda value elastic net also allows us to tune the alpha parameter where $\boldsymbol{\alpha}$ = 0 corresponds to ridge and $\boldsymbol{\alpha}$ = 1 to lasso. Simply put, if you plug in 0 for alpha, the penalty function reduces to the L1 (ridge) term and if we set alpha to 1, we get the L2 (lasso) term. Therefore, we can choose an alpha value between 0 and 1 to optimize the elastic net. Effectively this will shrink some coefficients and set some to 0 for sparse selection.

D. SUPPORT VECTOR MACHINE (SVM) MODEL

The main objective of support vector machines is to determine an optimal separating hyperplane, which correctly classifies the data points of different classes. The dimensionality of the hyperplane is equal to the (number of input features -1). One can chose many possible separating hyperplanes, but the objective is to find a plane that has the maximum margin. The data points closest to the separating hyperplanes are the support vectors.

The input to SVM is a set of (input, output) training pair samples. The features are denoted as x1, x2…xn, and the output result y. There can be high number of input features xi. The output is given as a set of weights wi for each feature, whose combination predicts the value of y.



The optimization function for SVM is as given below

$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, m \end{aligned}$$

The given optimization problem has concave quadratic objective function and only linear constraints. We can solve this problem by Lagrange duality.

$$L = \frac{1}{2}\|\vec{w}\|^2 - \sum_i^n \alpha_i \left[y_i(\vec{w}\cdot\vec{x}+b) - 1\right]$$

$$\frac{\partial L}{\partial w} = \vec{w} - \sum_i^n \alpha_i y_i x_i = 0$$

$$\boxed{\vec{w} = \sum_i^n \alpha_i y_i x_i}$$

$$\frac{\partial L}{\partial b} = -\sum_i^n \alpha_i y_i = 0$$

$$\boxed{\sum_i^n \alpha_i y_i = 0}$$

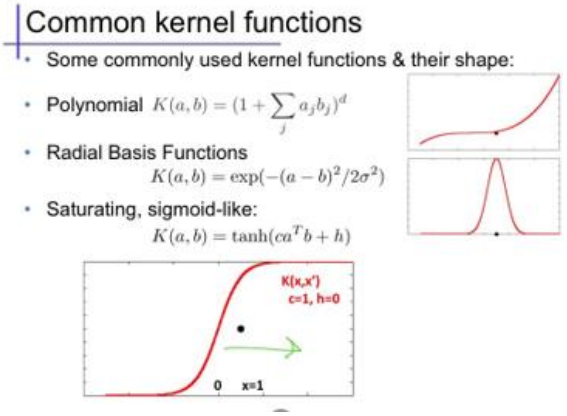
$$L = \sum_i^n \alpha_i - \frac{1}{2}\sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

The optimization of maximizing the margin is done to reduce the weights to a few, which correspond to the important features. These features are important in deciding the hyperplane and they correspond to the support vectors because they "support" the hyperplane.

There are two types of classification SVM algorithms - Hard Margin, which finds the separating hyperplane without any tolerance to any form of misclassification and Soft Margin, which allows misclassification of data points and permits the functional margin to be less that 1 (1-epsilon).

Kernels: Kernels are used to solve non-linear problems by using a linear classifier. It takes a low dimensional input space and transforms it into a higher-dimensional space. A kernel function is applied to each data point to map the non-linear observation to higher dimension where it can be separated linearly. Types of Kernels are listed below:

Gaussian Radial Basis Function: This kernel is a general-purpose kernel and is used when there is no prior knowledge about the data. Depending on σ, this kernel can either provide a good fit or an overfit. If the values of σ is larger than distance between the classes, it can give an overly flat discriminant surface. On contrary, if σ is smaller, this will over-fit the samples. A good value for σ will be comparable to the distance between the closest members of the two classes.



Polynomial: The Polynomial kernel is defined by equation shown below. Here d is the order of the kernel and 'r' is a constant that allows to trade off the influence of the higher order and lower order terms. Higher order kernels tend to overfit the training data and do not generalize well.

Linear: This is used when the data is Linearly separable, that is, it can be separated using a single line. It is one of the most common kernels to be used. It is mostly used when there are many Features in the dataset. Training a linear SVM model is generally faster than any other kernel.

$$k(x, y) = x^T y + c$$

Sigmoid: This kernel uses tanh function, and it is generally used as a proxy for neural networks. Alpha and constant c(intercept) are the two adjustable parameters in sigmoid. A common value for alpha is 1/N, where N is the dimension of the dataset.

Cost C (Regularization): C is a penalty parameter, and it represents how much misclassification is bearable. Through C, one can control the trade-off between decision boundary and misclassification.

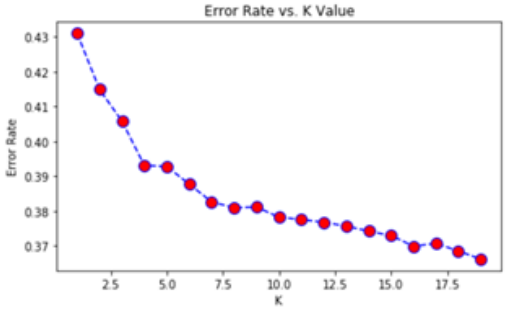
Gamma: Gamma defines how far the influence of single training example reaches. This value in set for different Kernels - 'rbf', 'poly' and 'sigmoid'.

V. RESULTS

The purpose of the prediction methodologies listed above is to classify the data that is in the form of articles in order to predict its popularity. After processing all the data and converting it into a form that can be easily interpreted by the machine learning algorithms, we separated the data into training and test sets in a ratio of 70:30. It is after this step that the model learns on the training data and is evaluated on the testing data. There are several model evaluation parameters like accuracy, precision, recall, F1 score and AUC value, that have been used by us to compare different models.

A. K NEAREST NEIGHBOR (KNN) BASELINE MODEL

We have treated KNN as the baseline model as it is easy to implement and tune. Moreover, it also does not make any assumptions with regards to the input data distribution.



Graph 11: Elbow method for optimal K value

We performed grid search and ran the KNN model for k values ranging from 1 to 20. As shown in the graph above, we then used the elbow method to discover that the optimal number of neighbors is at K=4.

```
accuracy_score on train dataset :  0.78
accuracy_score on test dataset :  0.59
confusion matrix on test data
[[3446 1974]
 [2181 2639]]
              precision    recall  f1-score   support

           0       0.61      0.64      0.62      5420
           1       0.57      0.55      0.56      4820

    accuracy                           0.59     10240
   macro avg       0.59      0.59      0.59     10240
weighted avg       0.59      0.59      0.59     10240
```

Result of KNN classifier at K=4

After having determined the optimal number of neighbors we trained the model again with the optimal value to achieve a baseline accuracy of 59% with precision of 61%.

B. NAÏVE BAYES MODEL

After the baseline KNN model, we developed the probabilistic naïve bayes model. It has the property of class conditional independence. Moreover, as we have also removed all the correlated variables, we went ahead and applied this model on our dataset. As compared to the baseline KNN model this model showed a further improvement in performance with an accuracy of 62% and precision of 61%.

```
accuracy_score on train dataset :  0.62
accuracy_score on test dataset :  0.62
confusion matrix on test data
[[4354 1074]
 [2827 1985]]
              precision    recall  f1-score   support

           0       0.61      0.80      0.69      5428
           1       0.65      0.41      0.50      4812

    accuracy                           0.62     10240
   macro avg       0.63      0.61      0.60     10240
weighted avg       0.63      0.62      0.60     10240
```

Best Results for the Naïve Bayes Classifier

C. LOGISTIC REGRESSION MODEL

With the aim to improve the accuracy further, we ran the Logistic Regression model. In Logistic Regression, we developed models on different regularizations like Ridge and Lasso. In order to get a good idea of the generalization ability of the model, we performed 10-fold cross validation technique. Moreover, we also experimented with the cost values by varying them from 0.001 to 10000.

Penalty	Cost	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
L1	1	64.9	64	64	64	64
	0.001	52.6	53	27	50	35
L2	0.001	63.1	62.8	63	62	62
	0.01	64.5	63.8	64	63	64
	0.1	64.8	63.9	64	64	64
	1	64.9	64	64	64	64
	10	64.9	64.2	64	64	64
	100	64.9	64.2	64	64	64
	1000	64.9	64.2	64	64	64
	10000	64.9	64.2	64	64	64

After performing several trials and tuning the parameter through grid search, we observed that Lasso turned out to be the best Logistic Regression model with an accuracy of 64.2%.

D. SUPPORT VECTOR MACHINE (SVM) MODEL

Support vector machine contains many hyperparameters that can be tuned. Thus, we performed a grid search in order to discover the optimal set of hyperparameters for the SVM models. The grid search was performed on hyper parameters like kernel, cost, and gamma values.

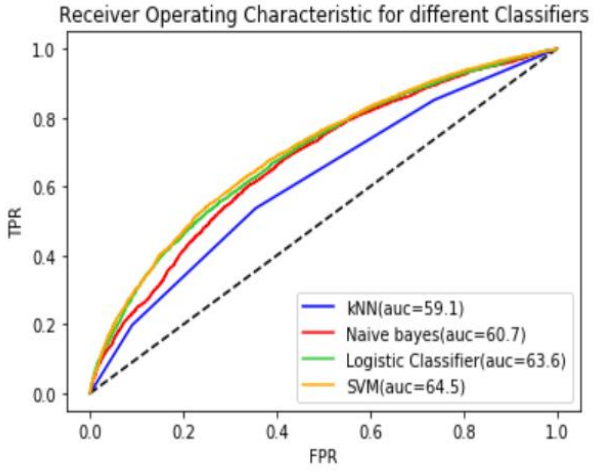
We varied the gamma values from 0.0001 to 100 while the cost values were varied from 0.01 to 1000. Unique combinations of the gamma and cost values were then experimented by picking different kernels like linear, sigmoid, radial, and polynomial. In case of the polynomial kernel, we experimented with the different gamma and cost values for degree 3. Some of the best results that we captured during the grid search have been tabulated below.

HyperParameters			Best Model's Evaluation Parameters				
Kernel	Gamma	Cost	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
Linear	0.0001	1	64.8	64.9	64	64	64
	0.01	1	64.8	64.1	64	64	64
	1	1	64.8	64.1	64	64	64
	10	1	64.8	64.1	64	64	64
	100	1	64.8	64.1	64	64	64
Sigmoid	0.0001	10	64.8	64	64	64	64
	0.01	0.1	64.8	64	64	64	64
	1	10	52.3	52.2	52	52	52
	10	0.01	51.4	52	51	50	48
	100	0.01	52.2	52.4	51	51	46
RBF	0.0001	100	64.2	63.8	64	63	63
	0.1	10	70.0	65.1	65	65	65
	1	0.1	67.4	63.7	64	64	64
	10	10	100	54.5	56	52	54
	100	0.01	52.6	53	27	50	35
Degree = 3							
Poly	0.0001	0.001	52.6	53	27	50	35
	0.01	100	63.4	62.8	64	62	61
	1	0.001	65.9	64.2	64	64	64
	10	0.0001	76	63.4	63	63	63
	100	0.001	62	63.2	63	63	63

The best accuracies reported by the usage of different kernels lie in the range of 64-65% with sigmoid kernel giving the least accuracy and RBF kernel giving the maximum accuracy of 65.1%. The best accuracy in the case of RBF kernel was achieved with a cost of value 10 and gamma being 0.1.

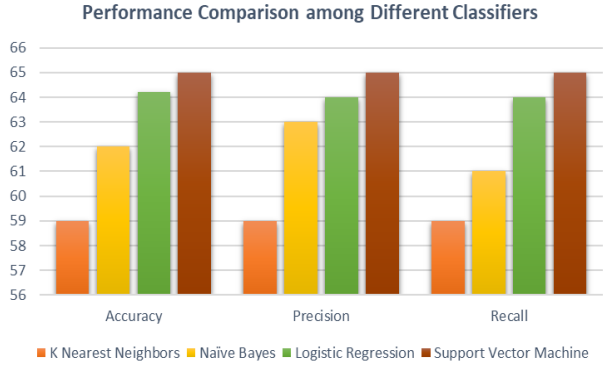
E. PERFORMANCE COMPARISON AMONG DIFFERENT CLASSIFIERS

AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve while AUC represents the degree or measure of separability. Higher the AUC, better is the model at predicting 0s as 0s and 1s as 1s. As we can observe from the graph below, all the classifiers except the baseline model have AUC > 60% with SVM having the maximum AUC of 64.5%.



Graph 12: ROC Curves for Different Classifiers

In general, the value of all the evaluation parameters i.e., Accuracy, Precision, Recall and AUC showed a gradual increase from the baseline model to the best model. Accuracy increased from 59% to 65% while AUC increased from 59.1 to 64.5 as we transition from the baseline KNN model to the best SVM radial model.



Performance Comparison among Different Classifiers

	Accuracy	Precision	Recall
K Nearest Neighbors	59	59	59
Naïve Bayes	62	63	61
Logistic Regression	64.2	64	64
Support Vector Machine	65	65	65

To summarize, the comparative analysis of the results from all the FOUR models clearly favors SVM with RBF kernel to be the best model.

VI. CONCLUSION

The prediction of news popularity has become a hot topic nowadays due to the huge expansion of social media over the years. This has made the online news as the main source of information due to the ease of accessibility. Such expansion thus demands for an advanced analytical algorithm that could predict the popularity of an article well before it is published in order to gain enough user attention. This prediction can help the publishers to maximize the user engagement and thus increase the revenue earned.

This paper focused at creating and evaluating different classification models that aimed at predicting whether an article will be popular or not based on the number of shares where we set a threshold of 1400. The performance of these classification models was then evaluated using several common evaluation metrics like accuracy, precision, recall and AUC values.

To summarize, PCA helped us in reducing the number of features while grid search helped us in arriving at the best hyperparameters. Moreover, the use of cross validation helped us to get an idea of the generalization ability of the different models. Hence, for this dataset, considering accuracy, precision, recall and AUC values of all the models, we have selected SVM with RBF to be our best model for prediction.

VII. FUTURE SCOPE

As it can be seen from the results, no classification algorithm could reach an accuracy of 70% and thus there is a scope of improvement in the classification accuracy. Existing model can be improved in terms of accuracy by performing more in-depth analysis and implementing new mechanisms like neural networks on this dataset.

Moreover, popularity can also be predicted by experimenting with other target variables like number of likes, number of comments or tweets, number of views etc. This would help us decide which amongst them is a better predictor of popularity.

Lastly, this problem can also be tackled as a multiclass classification problem wherein instead of having only two classes which are “popular” and “not popular”, we can have 5 classes namely “obscure”, “mediocre”, “popular”, “super-popular” and “viral”.

REFERENCES

[1] Kelwin Fernandes, Pedro Vinagre, and Paulo Cortez, “A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News” EPIA 2015, pp. 535–546, 2015.

[2] Ren, He, and Quan Yang. "Predicting and Evaluating the Popularity of Online News." Standford University Machine Learning Report (2015).

[3] Tatar, Alexandru, Panayotis Antoniadis, Marcelo Dias De Amorim, and Serge Fdida. "Ranking news articles based on popularity prediction." In 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 106-110. IEEE, 2012.

[4] Arapakis, Ioannis, B. Barla Cambazoglu, and Mounia Lalmas. "On the feasibility of predicting news popularity at cold start." In International Conference on Social Informatics, pp. 290-299. Springer, Cham, 2014.

[5] Elena Hensinger, Ilias Flaounas, Nello Cristianini, “Modelling and predicting news popularity” Springer, Pattern Anal Applic, 2013, pp. 623-635

APPENDIX

Variable	Description
url	URL of the article and acts as a unique reference for each article
timedelta	Days between the article publication and the dataset acquisition.
n_tokens_title, n_tokens_content,	Set of variables related to total number of words in the title, content, unique words in content and average length of the tokens.
n_unique_tokens,	
average_token_length	
n_non_stop_words,	Set of variables related to rate of non-stop words in the content and rate of unique non-stop words
n_non_stop_unique_tokens	
num_hrefs,	Set of variables related to number of external links in the content and links to other articles published by Mashable in the content and number of images and videos in the content.
num_self_hrefs	
num_imgs,	
num_videos	
num_keywords	Number of keywords in the metadata
data_channel_is_lifestyle	Binary variables indicating the type of content of the article
data_channel_is_entertainment	
data_channel_is_bus	
data_channel_is_socmed	
data_channel_is_tech	
data_channel_is_world	
weekday_is_monday,	Binary variables indicating the day the week the article was published.
weekday_is_tuesday	
weekday_is_wednesday	
weekday_is_thursday	
weekday_is_friday	
weekday_is_saturday	
weekday_is_sunday	
is_weekend	
LDA_00, LDA_01, LDA_02	Set of variables indicating closeness to LDA topics 0,1,2,3 and 4.
LDA_03, LDA_04	
kw_min_min	Set of variables related to the worst, best and average keywords based on the minimum, maximum and average shares of the article
kw_max_min	
kw_avg_min	
kw_min_max	
kw_max_max	
kw_avg_max	
kw_min_avg	
kw_max_avg	
kw_avg_avg	
self_reference_min_shares	Set of variables minimum, maximum and average shares related to self reference.
self_reference_max_shares	
self_reference_avg_sharess	
global_subjectivity	Set of numerical variables related to the subjectivity, sentiment polarity, presence of positive and negative words in the article and the title of the article
global_sentiment_polarity	
global_rate_positive_words	
global_rate_negative_words	
rate_positive_words	
rate_negative_words	
avg_positive_polarity	
min_positive_polarity	
max_positive_polarity	
avg_negative_polarity	
min_negative_polarity	
max_negative_polarity	
title_subjectivity	
title_sentiment_polarity	
abs_title_subjectivity	
abs_title_sentiment_polarity	