Predicting the Popularity of Online News Articles

PROJECT OVERVIEW

With the help of technological advancements, spreading online news around the world can be easily achieved. Social networks such as Twitter, Facebook created opportunities to read and share online news. Online news' popularity can be valuable for content providers, advertisers. Therefore, it is helpful to use machine learning techniques to predict the popularity of online news.

In this project, we will compare the models to find the best classification ones to accurately predict whether the news will be popular or not based on the dataset which includes 39 644 articles of the website called *Mashable*.

PROBLEM STATEMENT

This project explains how to use machine learning techniques to predict if an online news article will become popular. We will define a threshold that is used to label an article as popular or unpopular. The main problem is to extract the articles' features and find the best machine learning model to classify the target label. We can define the problem as a binary classification, therefore we will implement and compare three classification algorithms containing Logistic Regression, Adaboost, and Random Forest.

METRICS

Evaluation of machine learning algorithms is an essential part of the project. For this project, we will use three types of evaluation metrics including Classification Accuracy, F1 Score, AUC (Area Under Curve).

1. Classification Accuracy: It is the ratio of a number of correct predictions to the total number of input samples.

$$accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}$$

2. F1 Score: It is the harmonic mean precision and recall. The range of it is [0, 1]. It tells how precise the classifier is and how robust it is.

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

3. AUC (Area Under Curve): It is a widely used metric for binary classification. The AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example.

ANALYSIS

Data Exploration and Visualization

The dataset donated by the author [1]. includes 39 644 news articles from the website called *Mashable* and its acquisition date is January 8, 2015. The dataset and more descriptions can be found from this link. Each instance of the dataset covers 61 attributes including 2 non-predictive features (article's URL and days between the article publication and the dataset acquisition), a target feature (number of shares), and 58 predictive features. The dataset [2] is already preprocessed, namely, the categorical features have been transformed by the one-hot encoding scheme.

First of all, the main thing that we need to do is to determine the threshold to classify the articles as popular and unpopular. In Fig. 1 the statistics of the target attribute of the dataset are shown. From the figure, we can take 1 400 as a threshold. Using this threshold, we can convert a continuous number target attribute to boolean labels.

```
count     39644.000000
mean       3395.380184
std       11626.950749
min           1.000000
25%         946.000000
50%        1400.000000
75%        2800.000000
max      843300.000000
Name:  shares, dtype: float64
```

Figure 1. The statistics of the target attribute "shares".

We can observe other relevant features. In Fig. 2 the counts of popular and unpopular news over different days of the week are shown. From the figure, we can say that the online news published over weekends have a significant possibility to be popular. It is very likely as the majority of people spend more time surfing online news on weekends.

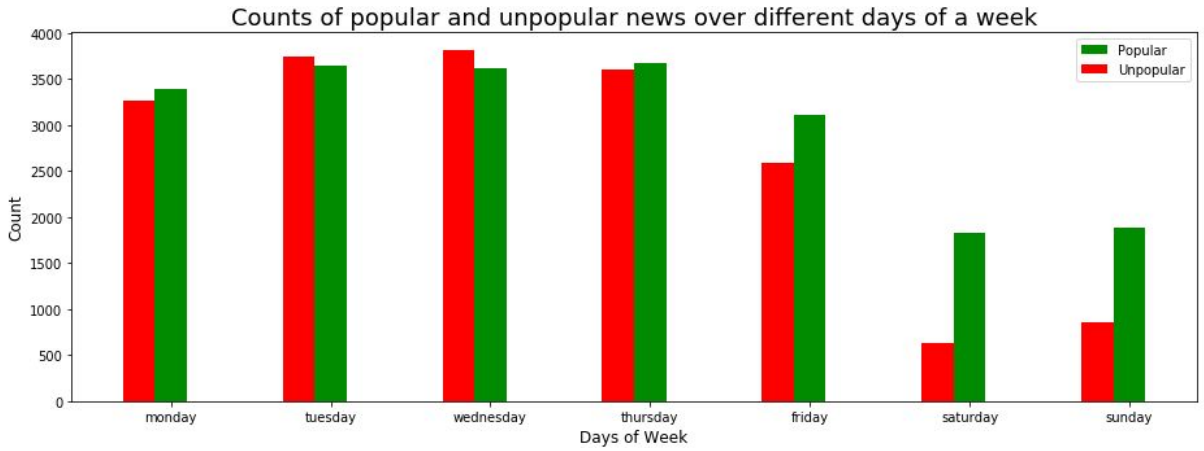


Figure 2. The counts of popular and unpopular news over days of the week.

In Fig. 3 the counts of popular and unpopular news articles over different categories are given. We can see that the categories of technology and social media are much popular than others. The categories of entertainment and world are much unpopular. This means that the users of the website *Mashable* prefer to read technology and social media news articles.

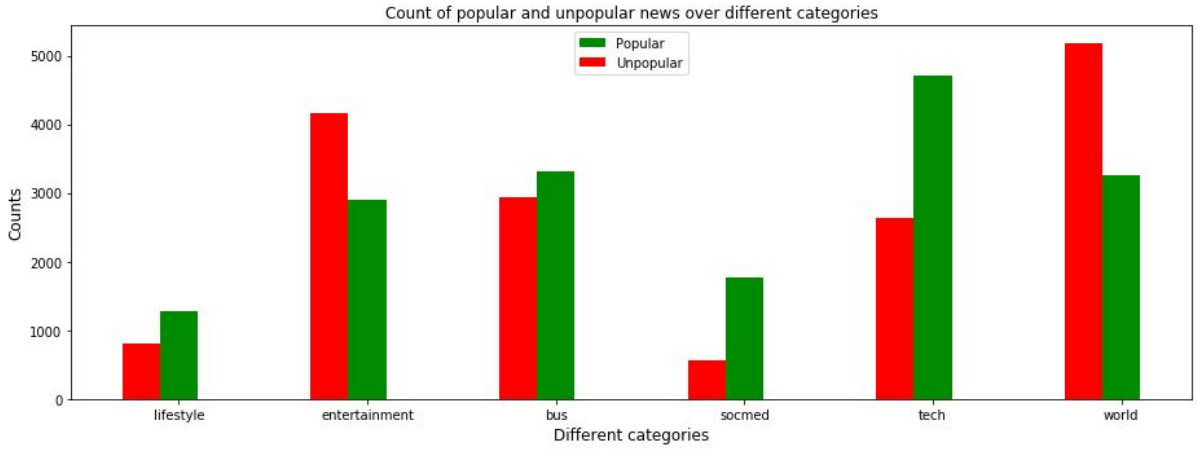
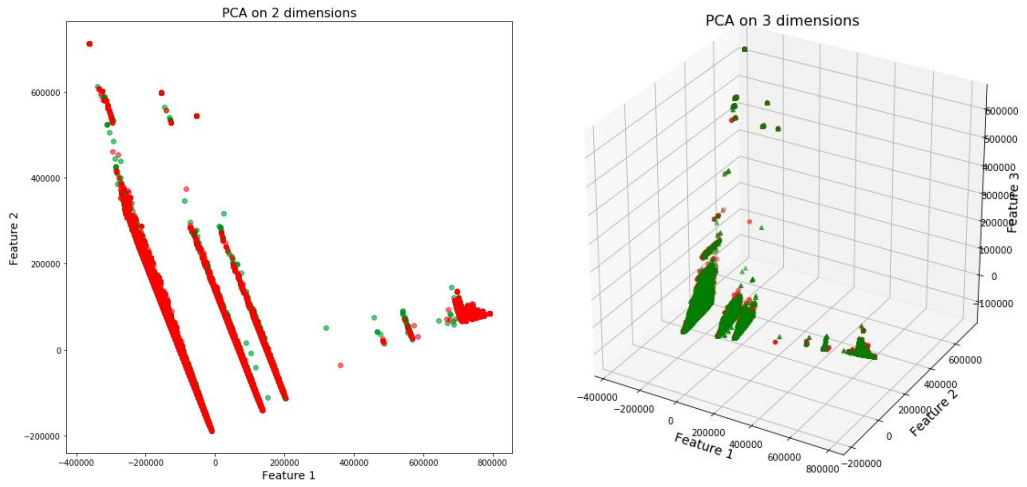


Figure 3. The counts of popular and unpopular news over categories.

We will check whether the data can be linearly separable. To do this, we will visualize the data, choose the number of components 2 and 3. In Fig. 4, it is clear that in PCA space, the data cannot be linearly separable.



(a) (b)

Figure 4. Data projection on (a) 2 components and (b) 3 components. (Green: popular, Red: unpopular)

Algorithm and Techniques

In this project, as we stated earlier, we can address the problem as a binary classification. Three classification algorithms which are Logistic Regression, Adaboost, and Random Forest will be used for this project. We will use the sklearn library to implement all the algorithms.

1. **Logistic Regression:** It is a linear model for classification. The output of the single-trial can be interpreted as a class probability which is generated using logistic regression or sigmoid function. The main advantage of it is that the training and prediction speed is very fast.
2. **Random Forest:** It is like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction, and the class with the most votes becomes our model's prediction.
3. **Adaboost:** short for "Adaptive Boosting", focuses on classification problems and aims to convert a set of weak classifiers into a strong one. It is an ensemble classifier starting fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

In this project, the algorithms will be implemented with the default hyperparameters, later to improve the performance grid search will be used for hyperparameters. For the RF (Random

Forest) algorithm, the hyperparameter “n_estimators” which is the maximum number of trees will be tuned. The higher number of trees, the better performance for the algorithm, but it may take slower. For Logistic Regression, the hyperparameter “C“ which is the inverse of regularization strength, will be tuned. The smaller the value of “C”, the lower magnitude of model parameters to prevent overfitting. For Adaboost, decision tree classifier is considered as the default base estimator. Like RF, the hyperparameter “n_estimators” means the maximum number of trees. For better performance, its value should be higher. But it will decrease the speed.

Benchmark

As a benchmark model, the work of the author [1] is taken for this project. In the work, using RF, 0.67 accuracy score, 0.69 of F1 score, and 0.73 of AUC score were achieved by tuning the hyperparameters. We will build the model that can achieve high accuracy as the benchmark model.

METHODOLOGY

Data Preprocessing

As we mentioned in “Data Exploration and Visualization” section, the dataset is already preprocessed. The categorical features are transformed by one-hot encoding. What we need to do is to normalize the numerical features to the interval [0, 1]. Additionally, as a threshold the median value of target attribute is selected to generate a boolean label.

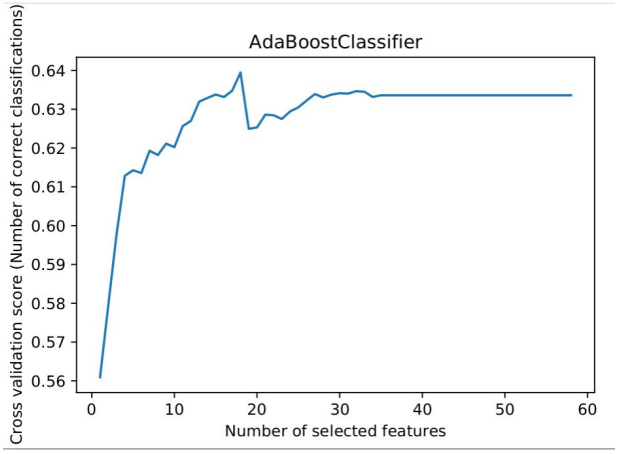


Figure 5. The cross validation score versus the number of features selected for Adaboost.

As there are many features in the dataset, conducting a feature selection process to reduce the data noise and increase the speed should be made. One of the effective ways of doing this process is using RFECV (recursive feature elimination with cross validation) to select

the most important features for a certain classifier. To do this, sklearn provides a function called RFECV.

In Fig. 5, it is shown that the cross validation score versus the number of features for Adaboost algorithm. From the graph, we can say that the number of features to be selected is 18. The selected features are listed in Table 1.

n_non_stop_unique_tokens	num_self_hrefs	num_imgs	num_videos
data_channel_is_entertainment	num_keywords	data_channel_is_socmed	kw_min_min
self_reference_min_shares	kw_min_max	kw_max_max	kw_min_avg
self_reference_max_shares	kw_avg_avg	kw_avg_min	kw_max_avg
global_subjectivity	global_sentiment_polarity		

Table 1. 18 features selected using RFECV with Adaboost estimator.

In Fig. 6, the cross validation score versus the number of features selected for RF estimator is shown. RFECV selects 38 features for RF.

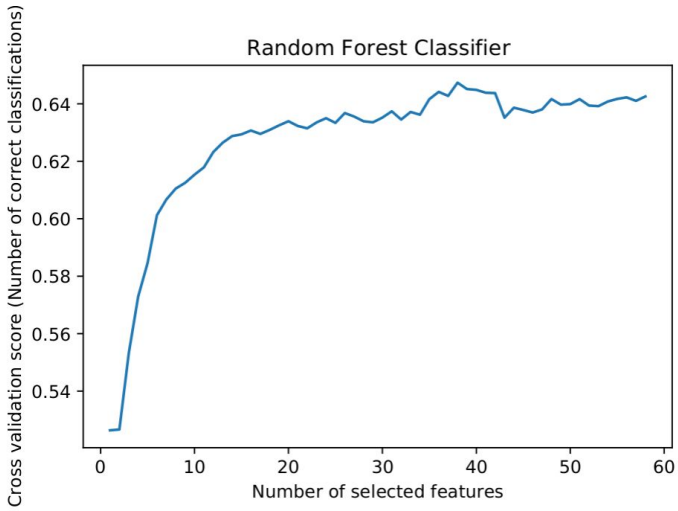


Figure 6. The cross validation score versus the number of features selected for RF

Lastly, the cross validation score versus the number of features selected for Logistic Regression is shown in Fig. 7. From the graph, you can see that the number of features selected is 8.

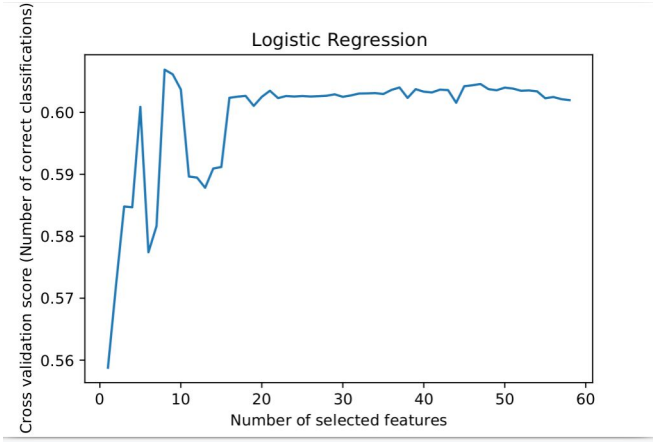


Figure 7. The cross validation score versus the number of features selected for LR

Implementation

The dataset is splitted with its own selected features into a training set (90%) and testing set (10%) for each algorithm. All algorithms are implemented with the help of sklearn. For the first process, the default hyperparameters are used. In Fig. 8, we can see three classification algorithms and their performance.

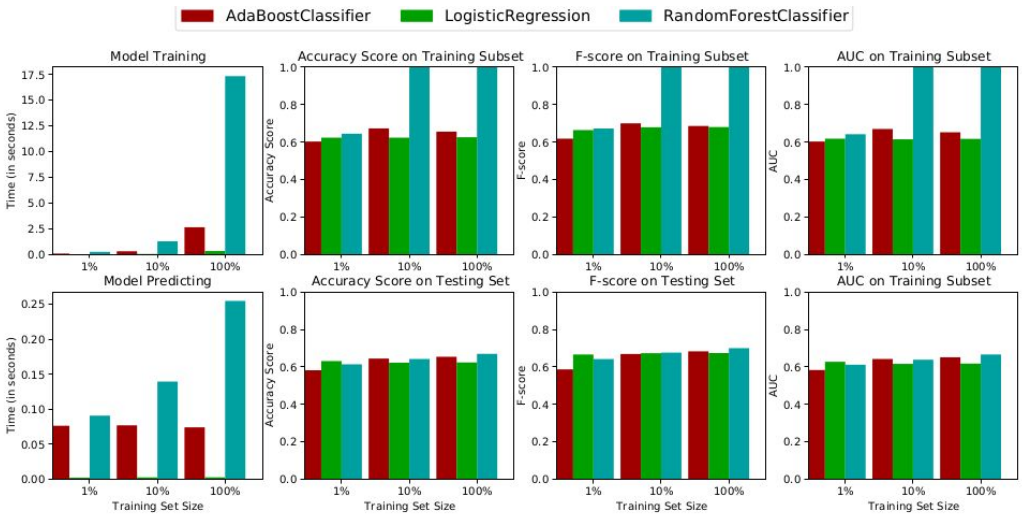


Figure 8. Performance of three classifiers with default hyperparameters.

The three metrics (Accuracy score, F1-score, AUC) are summarized in Table 2. Using default parameters. From the table, we can see that RF performs better in all three metrics. In the second place, Adaboost performs better.

Classifier	Accuracy	F1-score	AUC
Random Forest	0.66986	0.69997	0.66648
Logistic Regression	0.62396	0.67410	0.61775
Adaboost	0.65422	0.68315	0.65127

Table 2. Metrics score for three classifiers with default parameters

Refinement

In this section, to refine hyperparameters, the grid search method will be used for the classifiers. The method searches all possible combinations of model parameters, cross validates the model and finds which set of model parameters gives the best performance. Using the method, the learning algorithm can be optimized. The sklearn library provides the method for grid search. For **logistic regression**: "C" - [0.1, 0.5, 1.,2.,2.5,5], for **RF**: "n_estimators" - [10, 20, 50, 100, 250, 500], for **Adaboost**: "n_estimators" - [100, 200, 300, 400], "learning_rate" - [0.1, 0.5, 1].

After using the method, the refined hyperparameters are obtained as follows:

- **RF**: "n_estimators": 500
- **Logistic Regression**: "C": 2.0
- **Adaboost**: "n_estimators": 200, "learning_rate": 0.5

The results are summarized in Table 3. As compared with the Table 2, we can see that the RF and Adaboost are slightly improved with refined hyperparameters. But the Logistic Regression does not show any significant improvement.

Classifier	Accuracy	F1-score	AUC
Random Forest	0.66936	0.70265	0.66539
Logistic Regression	0.62395	0.67410	0.61775
Adaboost	0.65725	0.68780	0.65397

Table 3. Metrics for classifiers with refined hyperparameters.

RESULTS

Model Evaluation and Validation

After the steps of initial implementation and refinement of the classifiers, the best performance is obtained by the RF classifier with 500 trees in the forest. The best results by RF

are accuracy 0.6694, F1-score 0.7027, and AUC 0.6654. The final scores are reasonable. As shown in Fig.4, the dataset is not linearly separable, but still good performance in predicting news popularity compared with a random guess.

The dataset split ratio of training/testing set is changed from 0.1 to 0.15 to test the robustness of the models. In Table 4, the results are shown of classifiers with refined hyperparameters. Compared with Table 3, the performance of the models are similar, and the RF still shows the best performance.

Classifier	Accuracy	F1-score	AUC
Random Forest	0.66605	0.69909	0.66203
Logistic Regression	0.62300	0.67554	0.61599
Adaboost	0.65142	0.68385	0.64775

Table 4. Testing models with training/testing set ratio of 0.15..

Justification

Table 5 shows the final metrics of the benchmark work of the author [1]. In the work, the best performance is given by the RF model with accuracy score 0.67, F1-score 0.73, and AUC 0.73. Although the performance of my RF model is not better than the performance of the benchmark model, it is significant enough to solve the popular news classification problem.

Classifier	Accuracy	Precision	Recall	F1-score	AUC
RF	0.67	0.67	0.71	0.69	0.73
Naive Bayes	0.62	0.68	0.49	0.57	0.65
Adaboost	0.66	0.68	0.67	0.67	0.72
SVM	0.66	0.67	0.68	0.68	0.71
KNN	0.62	0.66	0.55	0.60	0.67

Table 5. The metrics of the benchmark models.

CONCLUSION

Free-Form Visualization

In this section, we will show the performance visualization of the classifiers with refined parameters. As shown in the visualization of Fig. 9, the training/testing time of RF is increased considerably as the number of estimators became 500, however it improved the performance of the model in terms of accuracy, F1-score, and AUC.

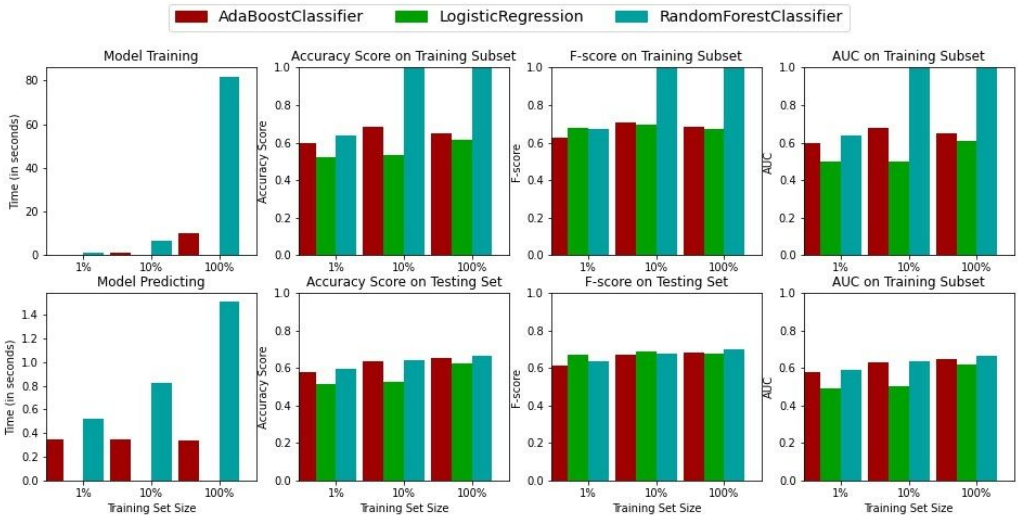


Figure 9. Performance of three classifiers with refined hyperparameters.

Reflection

To conclude, the project covered following steps:

1. **Data Collection**: The dataset that contains 39 644 news articles is downloaded from UCI Machine Learning Repository.
2. **Data preprocessing**: As the dataset is already preprocessed, normalizing numerical features is done to treat equally. As well as, the median value of the target attribute is selected as a threshold to generate the target boolean label.
3. **Data exploration and visualization**: Certain features are explored to check for relevance using visualization. PCA is also used to visualize data distribution.
4. **Feature selection**: To select the most relevant feature among 58 features, RFECV method is used for each classifier.
5. **Classifier implementation and refinement**: Sklearn library is used to implement algorithms of Logistic Regression, RF, Adaboost. Their hyperparameters are tuned by grid search method.
6. **Model evaluation and validation**: Three metrics (accuracy, F1-score, AUC) are used to evaluate the performance of the models. The performance is also compared with the benchmark model.

The difficult part of the project is how to define a problem, collect the relevant features and choose appropriate learning algorithms and refine hyperparameters.

Improvement

To improve the performance, I think there are several ways:

1. Increasing the size of the dataset to improve the performance of RF models.
2. Adding more relevant features to the dataset.
3. Using more advanced methods for validation.

REFERENCES

1. "A Proactive Intelligent Decision Support System for Predicting ...." https://www.semanticscholar.org/paper/A-Proactive-Intelligent-Decision-Support-System-for-Fernandes-Vinagre/ad7f3da7a5d6a1e18cc5a176f18f52687b912fea.
2. http://archive.ics.uci.edu/ml/datasets/Online+News+Popularity