Machine learning engineer nanodegree

Capstrone project

Online news popularity

Dumpala anusha

Feb 14,2019

1.Definition:

Project overview:

The consumption of online news increases day by day due to the wide spread adoption of smartphones and the rise of social networks. For instance, using social media like Twitter and Facebook. For the online news stake holders such as content providers or advertisers, it's very valuable if the popularity of the news articles can be accurately predicted prior to the publication. This project is based on the online news article popularity. Thus, it is interesting and meaningful to use the machine learning techniques to predict the popularity of online news articles .

Various works have been done in prediction of online news popularity. In 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. the popu- larity of online articles is analyzed based on the users' comments. And in S. Petrovic, M. Osborne, and V. Lavrenko, "Rt to win! predicting message propagation in twitter." ICWSM, vol. 11, pp. 586–589, 2011. the number of retweets is predicted using both the features of the retweet content (length, words, number of hashtag, etc.) whereas my work follows calssification task(article will be popular or not) based on the shares of the news article published

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

Problem statement:

The main aim of my project is to predict the future popularity of news article prior to its publication estimating the number of shares, likes, and comments etc. (features of an article). The dataset is publicly available at https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity# . So my goal is to predict the online news article popularity. Here I am using classification models s including Logistic Regression, RF, and Adaboost. to find the accuracy of each model and select the best model with high accuracy to predict the popularity. Here the input parameters

are training data that we took and the output will be the target label (popular/unpopular) . The tasks involved in it are:

1. Download the data

2. Visualizing the data.

3. Split the data and train and test which classifier performs good on the dataset based on the evaluation metric we have chosen.

4. Choose the best classifier that gives the best accuracy scores.

Metrics:

In the data set the class labels (+, -) are very closely balanced. we are using the following metrics.

(a) Accuracy: Accuracy is direct indication of the proportion of correct classification. It considers both true positives and true negatives with equal weight and it can be computed as

accuracy = true positives + true negatives/ dataset size (1) Although the measure of accuracy might be naive when the data class distribution is highly skewed, but it is still an intuitive indication of model's performance.

The performance of a model cannot be assessed by considering only the accuracy, because there is a possibility for misleading. Therefore this experiment considers the F1 score along with the accuracy for evaluation.

(b) F1-score: F1-score is an unweighted measure for accuracy by taking harmonic mean of precision and recall, which can be computed as

F1 = 2 · precision · recall/ precision + recall (2) Precision and Recall are defined as Precision=TP/ (TP+FP), Recall=TP/ (TP+FN), where
TP=True Positive
FP=False Positive
FN=False Negative

It is a robust measurement since it is independent of data class distribution.

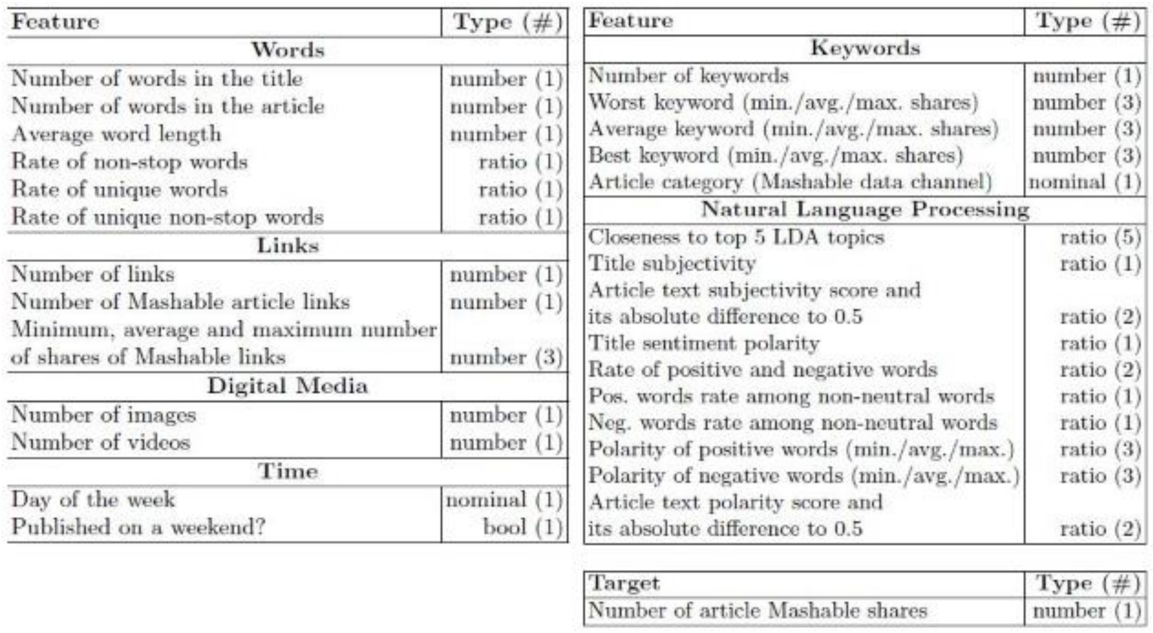
For indicating the usefulness of the classifier the following metric(AUC) is used.
(c) AUC: The AUC is the area under the ROC (Receiver Operating Characteristics) curve, which is a plot of the True Positive Rate versus the False Positive Rate. AUC value is a good measure of classifier's discrimination power and it is a more robust measure

For all above three metrics, the higher value of the metric means the better performance of model.

ll.Analysis

Data Exploration:

The dataset is downloaded from UCI Machine Learning Repository as https://archive.ics.uci.edu/ml/datasets/Online+News+ Popularity#. The dataset is consisted of 39,643 news articles from an online news website called Mashable collected over 2 years from Jan. 2013 to Jan. 2015. For each instance of the dataset : Number of Attributes: 61 (58 predictive attributes, 2 non-predictive, 1 goal field) The dataset has already been initially preprocessed. For examples, the categorical features like the published day of the week and article category have been transformed by one-hot encoding scheme, and the skewed feature like number of words in the article has been log transformed.



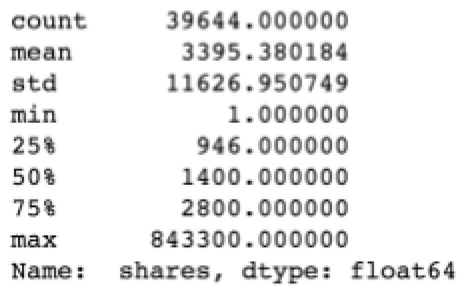
Feature	Type (#)
Words	
Number of words in the title	number (1)
Number of words in the article	number (1)
Average word length	number (1)
Rate of non-stop words	ratio (1)
Rate of unique words	ratio (1)
Rate of unique non-stop words	ratio (1)
Links	
Number of links	number (1)
Number of Mashable article 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)

Figure 1: List of predictive attributes of dataset

we show the statistics of the target attribute ”shares” to determine the appropriate threshold for number of shares to discriminate the news to be popular or unpopular. in Fig. 2, and we find the median of target attribute is 1,400, thus it is reasonable if we take 1,400 as a threshold. Then we can use this threshold to convert the continuous number target attribute into a boolean label.



```
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 2: Statistics of target attribute ”shares”.

By observing various features, I think there are several relevant features like day of the week and article category. In Fig. 3, the count of popular/unpopular news over different day of the week is plotted.

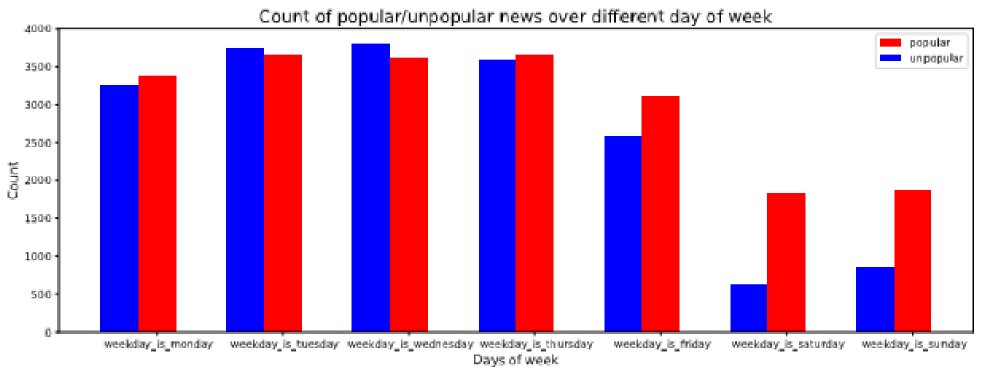


Figure 3: Count of popular/unpopular news over different day of week.

We can clearly find that the articles published over the weekends has larger potential to be popular. It makes sense because it is very likely that people will spend more time online browsing the news over the weekends

In Fig. 4, the count of popular/unpopular news over different article category is plotted.

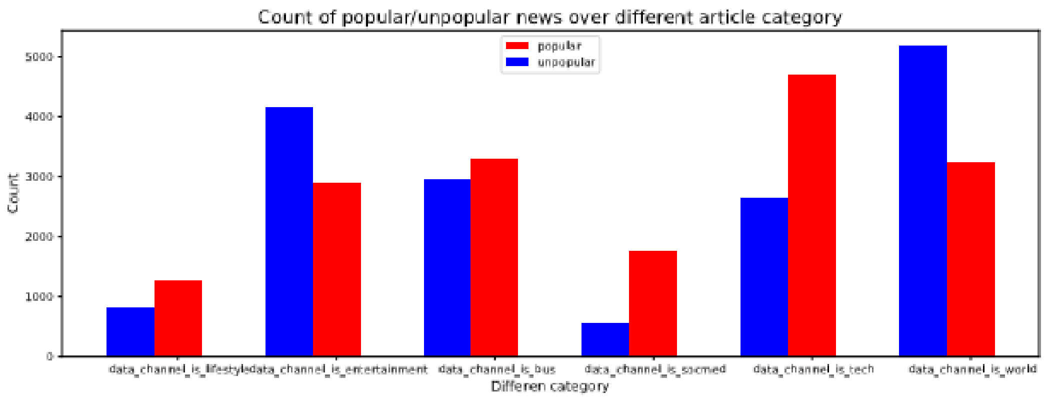
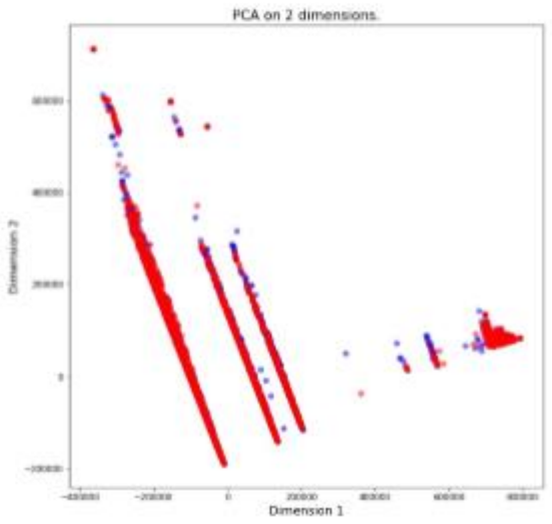


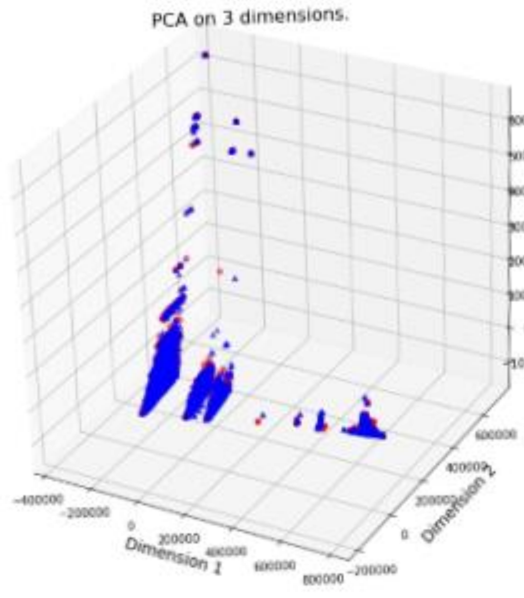
Figure 4: Count of popular/unpopular news over different article category.

We can observer that in category of technology (”data channel is tech”) and social media (”data channel is socmed”), the proportion of popular news is much larger the unpopular ones, and in category of world (”data channel is world”) and entertainment (”data channel is entertainment”), the proportion of unpopular news is larger the popular ones. This might reflects that the readers of Mashable prefer the channel of technology and social media much over the channel of world and entertainment.

Next, we do the principle component analysis (PCA) to visualize the data. As shown in Fig. 5, we project the data point onto first 2 and 3 principle components, respectively.



(a)



(b)

Figure 5: (a) Data projected on first 2 principle components (b) Data projected on first 3 principle components. Red: popular; Blue: unpopular It is clear that the dataset is not linearly separable in PCA space.

Algorithams and techniques

Since we formulate this problem as a binary classification problem three classification learning algorithms including Logistic Regression, RF,and Adaboost will be implemented

Logistic regression

Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts P(Y=1) as a function of X

Advantages:

Because of its efficient and straightforward nature, doesn't require high computation power, easy to implement, easily interpretable, used widely by data analyst and scientist. Also, it doesn't require scaling of features. Logistic regression provides a probability score for observations

Disadvantages:

Logistic regression is not able to handle a large number of categorical features /variables. It is vulnerable to over fitting. Also, can't solve the non-linear problem with the logistic regression that is why it requires a transformation of non-linear features. Logistic regression will not perform well with independent variables that are not correlated to the target variable and are very similar or correlated to each other.

Parameters:

Class sklearn.linear_model.LogisticRegression (penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='warn', max_iter=100, multi_class='warn', verbose=0, warm_start=False, n_jobs=None)

RANDOM FOREST:

Tree models are known to be high variance, low bias models. In consequence, they are prone to overfit the training data. This is catchy if we recapitulate what a tree model does if we do not prune it or introduce early stopping criteria like a minimum number of

instances per leaf node. Well, it tries to split the data along the features until the instances are pure regarding the value of the target feature, there are no data left, or there are no features left to spit the dataset on. If one of the above holds true, we grow a leaf node.The consequence is that the tree model is grown to the maximal depth and there with tries to reshape the training data as precise as possible which can easily lead to over fitting. Another drawback of classical tree models like the (ID3 or CART) is that they are relatively unstable. This instability can lead to the situation that a small change in the composition of the dataset leads to a completely different tree model.
Parameters: Class sklearn.ensemble.RandomForestClassifier (n_estimators='warn', criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1,

min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None

Advantages:

1.Reduction in over fitting: by averaging several trees, there is a significantly lower risk of over fitting.

2.Less variance: By using multiple trees, you reduce the chance of stumbling across a classifier that doesn't perform well because of the relationship between the train and test data.

Disadvantages :

1. It takes more time to train samples.

Adaboost

AdaBoost classifier is an ensemble classifier that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. sklearn provides a function called AdaBoostClassifier(). One tunable parameters is ”n estimators”, which is the maximum number of estimators at which boosting is terminated. Another is ”learning rate”, which means the contribution of each classifier shrinks by the value of ”learning rate”. There is a trade-off between ”n estimators” and ”learning rate”. The boosting can help in learning complex non-linear hypothesis which is necessary because the dataset here is not linearly separable.In this project, we will first implement all three algorithms with the default

hyperparameter settings, then we will use grid search method to refine the hyperparameters and thereby improve the model's performance. For logistic regression, we will tune the hyperparameter"C", which is the inverse of regularization strength. "C" is basically used for control overfitting. A smaller "C" means higher regularization over the model parameters, thus smaller "C" will decrease the magnitude of model parameters to prevent over-fitting. For RF, we will tune hyperparameter "n estimators", which is the maximum number of trees you can build before taking the maximum voting or averages of predictions. Higher number of trees can give better performance but makes code running slower. For Adaboost, the default base estimator is decision tree classifier. Similar to RF, hyperparameter "n estimators" represents the maximum number of trees you can build before termination. Higher numberof trees may lead to better performance but will decrease the model running speed. The trade-off between "n estimators" and "learning rate" is as follows: a larger "learning rate" typically means smaller "n estimators" in Adaboost but might lead to over-fitting, while a smaller
"learning rate" typically means larger "n estimators" and it will control over-fitting but slow down the running speed.

Benchmark

Benchmark model is a model which we will take as reference and achieve the best result than the benchmark model .In my project. Here Accuracy score will be compareD between the different classification models(Adaboost,Random Forest,Logistic regression) and select the best one.

lll.methodlogy

data pre-processing:

As mentioned in Section 2.1, some data preprocessing works have been done by the data's donator. The categorical features like the published day of the week and article category have been transformed by one-hot encoding scheme, and the skewed feature like number of words in the article has been log-transformed. I further preprocess the dataset by normalizing the numerical feature to the interval [0, 1] such that each feature is treated equally when applying supervised learning.

Since there are 58 features in the dataset, it is reasonable to conduct a feature selection to reduce the data noise and increase the algorithm running speed. One effective way is using recursive feature elimination with cross validation (RFECV) to automatically select the most significant features for certain classifier. sklearn provides a function called REFCV() that can help us.

Firstly, we run RFECV with a logistic regression estimator. The cross validation score versus the number of feature selected

versus the number of feature selected

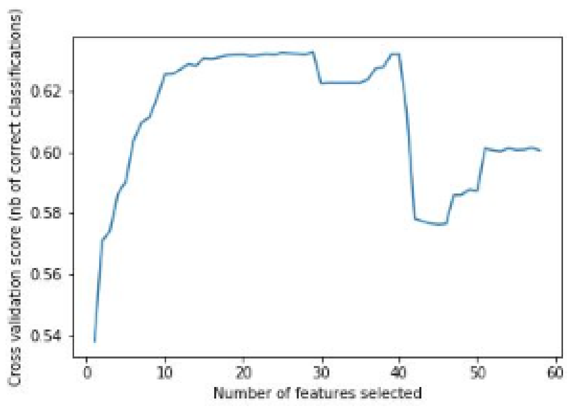
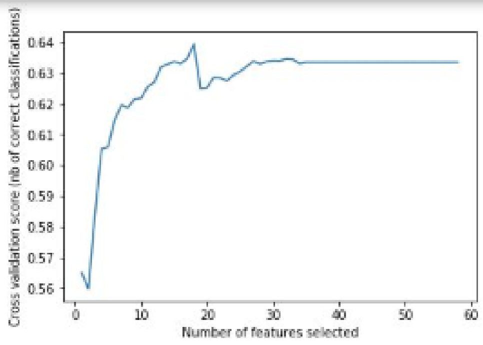


Figure 6: The cross validation score versus the number of feature selected using RFECV for logistic regression estimator.

From the figure, we can find there is drop of score when number of features is 29. Thus RFECV algorithm select 29 most relevant features from 58 original features as shown in below figure.

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			

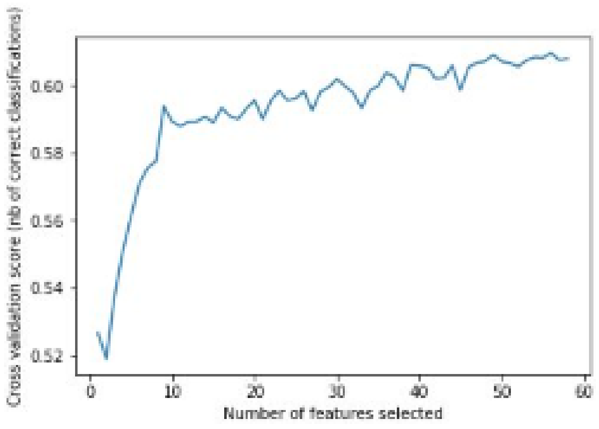
Next ,we run with Adaboost



From the figure, we can find there is a peak when number of features is 18. Thus RFECV algorithm select 18 most relevant features from 58 original features.

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

Lastly, we run with Random forest



We find that REFCV selects 56 features(almost the full features) for RF.The two features that are not selected are[n non stop words', ' data channel is lifestyle'].

Implementation

Before algorithm implementation, for each algorithm, I also randomly split dataset with its own selected features into training set (70%) and testing set (30%). Then, a method called as '"train_predict()"' is defined that takes as input the following: learner, sample_size, X_train, y_train, X_test, y_test. It returns the accuracy and F1 and AUC score on training and testing set respectively The three classifiers (logistic regression,RF and Adaboost) are sent to the 'train_predict' method, with 1%, 10% and 100% of training data respectively so that it can be seen how performance varies with sample size The logistic regression,RF and Adaboost are implemented by the sklearn function LogisticRegression(), RandomForestClassifier() and AdaBoostClassifier(), respectively. At this stage, I will first use the default setting for model hyperparameters. A method "evaluate()" is used for Visualization code to display results of various learners. I run the three classification algorithms and their performance is as

Classifier	accuracy	f1score	AUC
Adaboot	0.6432	0.67598	0.6393
Logistic regression	0.6329	0.67153	0.6280

Random forest.	0.6251	0.6282	0.627

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.RF runs fast than Adaboost

Refinement:

In this part, I will use the grid search method for each of the three classifiers to refine their hyperparameters.After running the grid the search, the refined hyper parameters are obtained.

regression - {”C”: 0.5}; RF-{”n estimators”: 500}; Adaboost-{”n estimators”: 300, ”learning rate”: 0.5}.

Now we run the three classifiers with refined parameters, the three metrics are summarized.

classsifier	acuracy	f1score	AUC
adaboost	0.6508	0.6830	0.6470
Logistic regression	0.6329	0.6714	0.628
1 Random forest.	0.6678	0.6991	0.6639

IV. Result

Model evaluation and validation After initial implementation and further refinement for the three classifiers, we find the best performance is obtained by the RF classifier with 500 trees in the forest. The best obtained metrics of RF are accuracy 0.6769, F1-score 0.7073 and AUC 0.6734.To test the robustness of the model, we change split ratio of training/testing set from 0.30 to 0.15, and then run the three classified with the same refined hyperparameters. Now the metrics are

classsifier	acuracy	f1score	AUC
adaboost	0.6494	0.6818	0.64576

Logistic regression	0.6324	0.6689	0.62825
Random forest.	0.6742	0.7052	0.67055

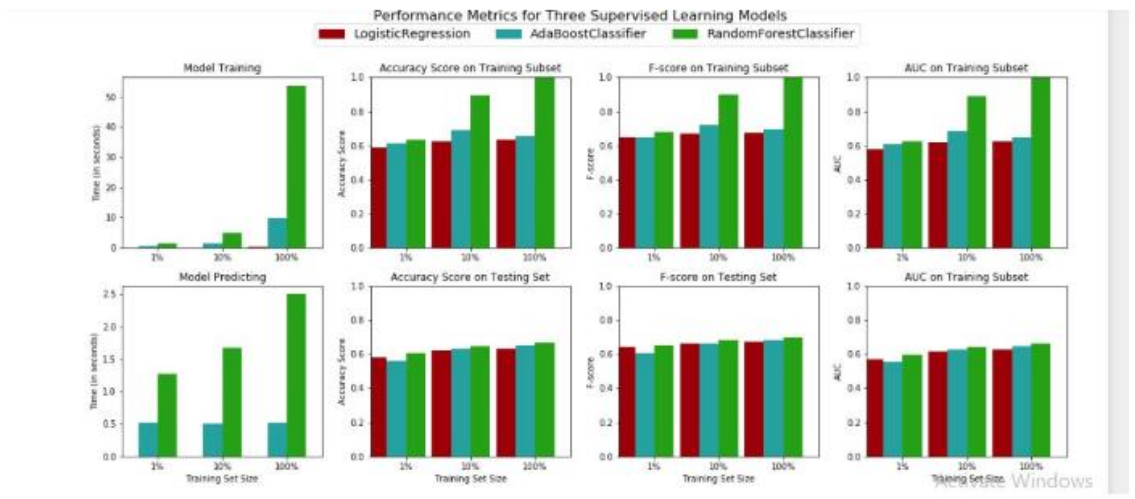
Justification:

The best performance is also given by RF model, which achieves 66% of accuracy ,69% of F1 score and AUC of 66%.So we can say the obtained model achieves a comparable performance than Logistic regression, adaboost and it is significant enough to solve the popular news classification problem.

V. Conclusion

FREE FORM VISUALIZATION:

In this part, the performance of three classifiers with refined hyperparameters is finally displayed . As shown in the visualization, the training and testing time of RF is greatly increased since 500 trees are used in the forest, but it helps RF achieve the best performance in terms ACCURACY F1 SCORE and AUC



Reflection:

To summarize, the project is conducted through following steps: (a) Data collection: The dataset of some 40,000 online news articles are downloaded from UCI Machine Learning Repository,

(b) Data preprocessing: Based on the initial data preprocessing in original dataset, I fur- ther preprocess the dataset by normalizing the numerical feature such that each feature is treated equally when applying supervised learning. I also select the median of the target attribute (number of shares) as an appropriate threshold to label all the data as either popular or unpopular.

(c) Data exploration and visualization: I explore the relevance of certain feature by visu- alization. And I also visualize the data distribution by PCA.

(d) Feature selection: To select the most relevant features among all 58 features, I use RFECV to select the most relevant features for each of the classifier.

(e) Classifier implementation and refinement: The classification algorithms including Lo- gistic Regression, RF and Adaboost are implemented using sklearn. Then the model's hyperparameters are tuned by grid search method.

(f) Model evaluation and validation: The refined models are evaluated and compared using three metrics (accuracy, F1-score, AUC). The performance is also compared with the benchmark model. The interesting aspect of this project is from the feature selection part. I assume the day of the week and article category could be the relevant features by analysis and visualization. And later in the part of feature selection by RFECV, my guess is verified, which also shows the effectiveness of RFECV. The difficult part of the project is

(1) how to define a problem, collect the corresponding data and engineer/select the relevantfeatures;

(2) how to choose appropriate learning algorithm and refine the hyperparameters. Since the obtained model achieve a fairly good performance, I think RF model with a refined hyperparameter should be used in a general setting to solve these types of problems

Improvement:

the improvement can be done as follows: 1.engineer and add more relevant features to the original dataset dataset since RF has a strong learning capability and a rich dataset might improve its prediction performance 2.we could use all the words in an article as additional features, and then try the classifier such as Naive Bayes to see if it can achieve a better performance.