Predicting Popularity of Online News Article

Fatema-E- Jannat
Dept. of Electrical Engineering
University of South Florida
Tampa, Florida, USA
fjannat1@mail.usf.edu

***Abstract*—Because of the easy access of the internet, the online news has become very popular among the people of recent days. It is now a very common trend to share online news if anyone finds it interesting or valuable. Based on this sharing number we can judge which news is now trending or popular among people. This number of sharing depends on several features. Using a huge dataset, provided by UCI Machine Learning Repository, with about 39,797 articles and total 61 attributes, an analysis was done to find the best model and features to predict the popularity of any news article even before being published. Random Forest Regression Model gave promising performance for this dataset and for the classification model, Adaboost showed a significant performance of 67% accuracy.**

Keywords—regression, classification, prediction, pca, data-mining, feature-reduction, random-forest, ada-boosta, knn

I. Introduction

The online news website like Mashable, New York Times, BBC News etc publish a huge number of articles of several categories every day. The popularity of any article depends on several features. The aim of this project is to predict the popularity of any article before being published which will help the publishers and editors to maximize the popularity of the article and to increase the profit.

Regression will be applied to predict how many shares can any article get after getting published. And then Classification will be applied to predict if that article will be popular or not after getting published.

For marking any article popular or unpopular a threshold has been set which is the median of the shares, 1400. So any article which has over 1400 shares will be considered as a popular one, and any article which has shares below 1400 will be considered as an unpopular article.

This project will cover both the regression and classification formulation. In section II, the development plan will be described briefly. Section III will give an overview of the data acquisition and data information. Section IV will provide a comprehensive detail in data preprocessing. In section V, the model implementation will be described. Finally in section VI, the overall performance will be discussed.

II. Development Plan

Step 1: The first step will be to read and load the dataset

Step 2: For further processing it is important to preprocess the raw data and turn them into an understandable format. So after loading the dataset, the next step will be the data preprocessing.

Step 3: The I will implement different learning techniques such as Logistic Regression, Random Forest Regression, Linear Regression. For classification I will implement KNN, Neural Network, Random Forest, AdaBoost

Step 4: The I will make a comparison among those techniques to find the best model and features to predict the popularity of certain articles which can be used for different online news companies to estimate the value of their articles before publishing.

Fig. 1. Flow diagram describing the overall development plan

III. Data Acquisition

The dataset that I have chosen is mainly provided by UCI Machine Learning Repository. This dataset has total 61 attributes, among those 58 predictive attributes, 2 non-predictive, 1 goal field. This dataset is composed of 39797 instances which means it has 39797 articles which were taken from Mashable (www.mashable.com).

Data was collected on 31 May, 2015 and it contains data from January 7, 2013 to July 12, 2014. The statistical values of shares can be found in table-1.

Table1: Statistical measurement of shares

count	39644.000000
mean	3395.380184
std	11626.950749
min	1.000000
25%	946.000000
50%	1400.000000
75%	2800.000000
max	843300.000000

The overall datatype of the features can be found in table-2.

Table2: Features and the data-type

url	object
timedelta	float64
n_tokens_title	float64
n_tokens_content	float64
n_unique_tokens	float64
n_non_stop_words	float64
n_non_stop_unique_tokens	float64
num_hrefs	float64
num_self_hrefs	float64
num_imgs	float64
num_videos	float64
average_token_length	float64
num_keywords	float64
data_channel_is_lifestyle	float64
data_channel_is_entertainment	float64
data_channel_is_bus	float64
data_channel_is_socmed	float64
data_channel_is_tech	float64
data_channel_is_world	float64
kw_min_min	float64
kw_max_min	float64
kw_avg_min	float64
kw_min_max	float64
kw_max_max	float64
kw_avg_max	float64
kw_min_avg	float64
kw_max_avg	float64
kw_avg_avg	float64
self_reference_min_shares	float64
self_reference_max_shares	float64
self_reference_avg_sharess	float64
self_reference_min_shares	float64
self_reference_max_shares	float64
self_reference_avg_sharess	float64
weekday_is_monday	float64
weekday_is_tuesday	float64
weekday_is_wednesday	float64
weekday_is_thursday	float64
weekday_is_friday	float64
weekday_is_saturday	float64
weekday_is_sunday	float64
is_weekend	float64
LDA_00	float64
LDA_01	float64
LDA_02	float64
LDA_03	float64
LDA_04	float64
global_subjectivity	float64
global_sentiment_polarity	float64
global_rate_positive_words	float64
global_rate_negative_words	float64
rate_positive_words	float64
rate_negative_words	float64
avg_positive_polarity	float64
min_positive_polarity	float64
max_positive_polarity	float64
avg_negative_polarity	float64
min_negative_polarity	float64
max_negative_polarity	float64
title_subjectivity	float64
title_sentiment_polarity	float64
abs_title_subjectivity	float64
abs_title_sentiment_polarity	float64
shares	int64

IV. Data Pre-processing

A. Data Cleaning

This dataset contains two non-predictive features, so at first these two non-predictive features were removed. Then it was checked if there were any null values, fortunately the dataset was clean.

B. Removing Recent Articles

The recent articles where time delta is less than 1 month were discarded since the convergence for those articles was not reached yet.

C. Excluding Outliers

Outliers were being excluded from target column using Z-score method. In this approach, it removes the outlier points by eliminating any points that were greater than (μ+ kσ) and less than (μ-kσ), for this project k=2 was chosen.

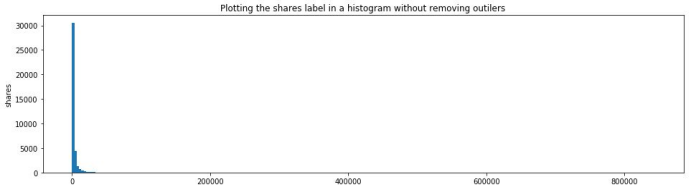

fig-2: Plotting the shares in a histogram before removing outliers

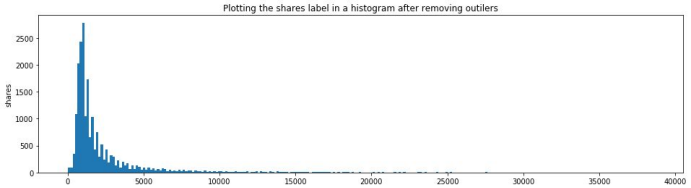

fig-3: Plotting the shares in a histogram after removing outliers

Size of dataset become (37964, 59) after removing non-predictive features, recent articles and outliers from shares column.

D. Normalizing Features

The numerical features were then normalized using Minmax normalization technique.

In this technique, the data is scaled in between 0 and 1. For minmax this equation is used,

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}}.$$

Data normalization is done to make the data more understandable and to reduce the redundancy throughout the dataset.

E. Dimensionality Reduction

This technique is used for reducing the number of features in a dataset while dealing with lot of features. PCA (Principal Component Analysis) is one of the dimension reduction technique which is used to down-sample high-dimensional datasets. It reduces the dimensionality and find a new set of features called component. These components are composed of original features but

uncorrelated with each other. First component has the highest possible variability in the dataset, 2nd component has second high variability and so on. For this project, the number of component is set to 5.

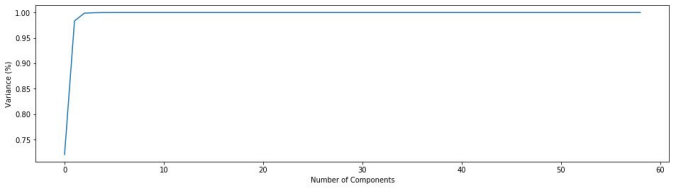

fig-4: Explained variance of the dataset

From fig-4 it is clear that if you chose 5 components then we will be able to preserve above 95% of total variance.

F. Feature Selection

Implementing 'feature_importances' technique reduces the features. This technique allows to build a list by the importance of all the features in a form of probability. So from that list we can discard the less important features and observe how it affects on the performance.

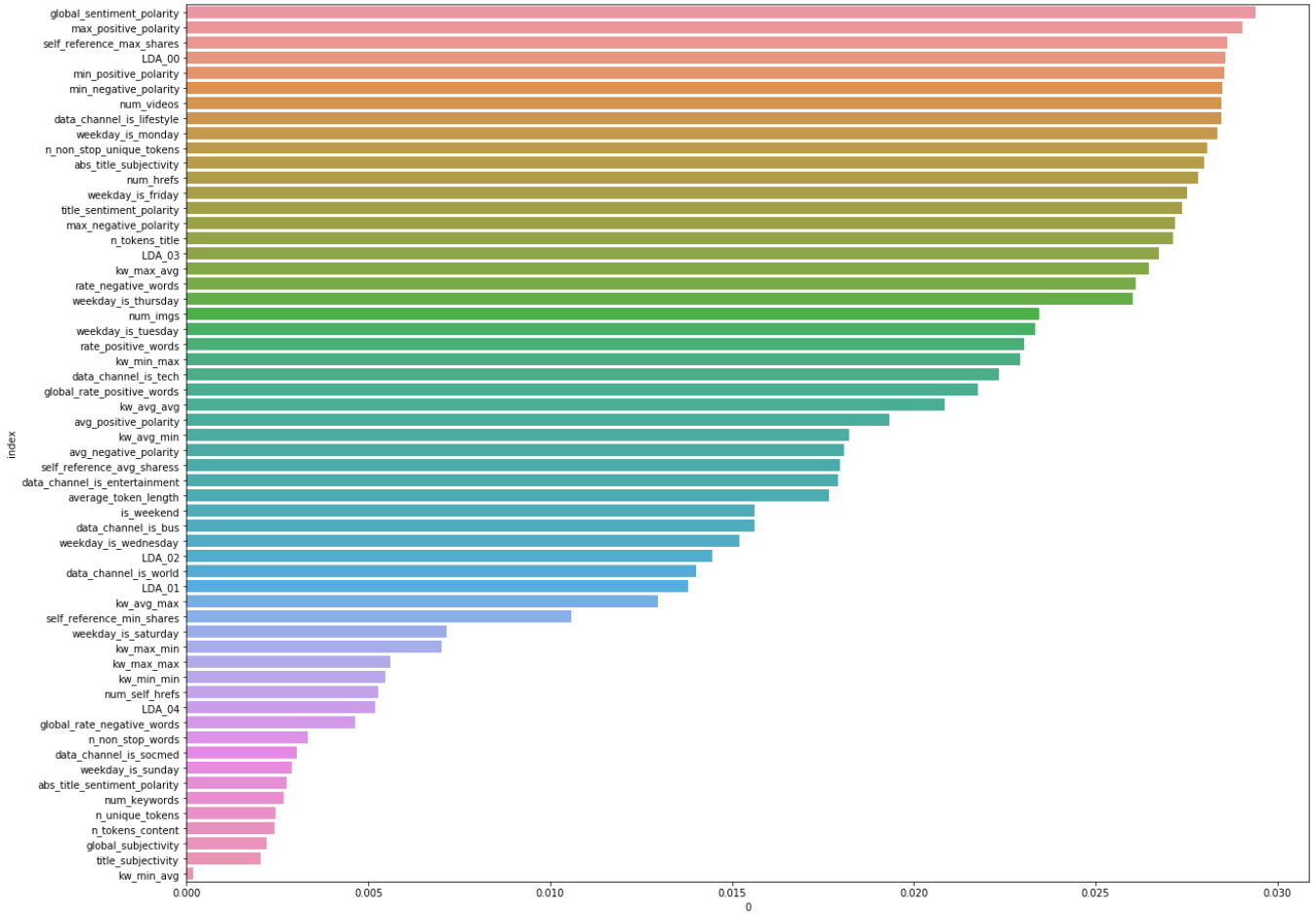

fig-5: Feature importance value for all features

In fig-5, the feature importance value can be seen for all the features. I have used Relief (feature selection) technique for my project. It computes a feature score for all the existing features from the dataset and then from them top scored features are selected.

In this process the features having lower importance value are removed. For example, the highest importance value is 0.029411 which is for "global_sentiment_polarity" feature, whereas the feature "kw_min_avg" has only 0.000185 as importance value which is quite low compared to the highest value. So this feature can be discarded. Removing lower importance valued features from the dataset reduces the noise.

In my project, while using this Relief technique I kept top 20 features. I set this number after doing several trial and error.

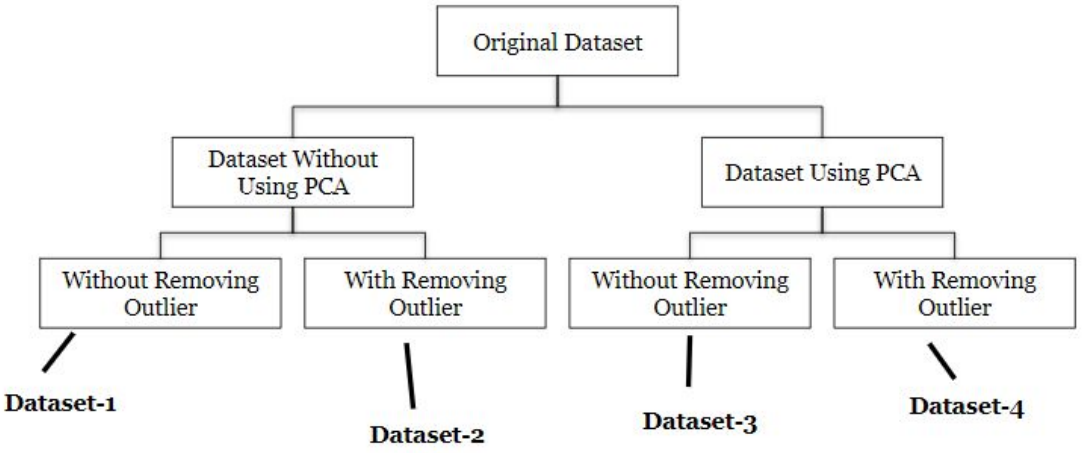

fig-6: The experimental Set-up (Subgroups of dataset)

G. *Experimental Setup*

I have divided the whole dataset into 4 subgroups. These can be seen from fig-6. Then I implemented the models into these 4 subgroups to observe how they perform, if there is any positive effect of removing outliers or applying PCA.

V. Model Implementation

In this project we employed the Scikit Learn Library to implement the models.

For the regression analysis I have implemented three models, named Random Forest, Linear Regression, Logistic Regression.

For the classification analysis I have explored Random Forest, AdaBoost, KNN, NN these 4 models.

The dataset was split into two separate parts, training and testing. 80% of the data used as training set and 20% of data used as testing set. This separation was made for each of four datasets. Then the model was implemented onto these to observe the difference of performance.

VI. Experiments and Results

A. *Regression*

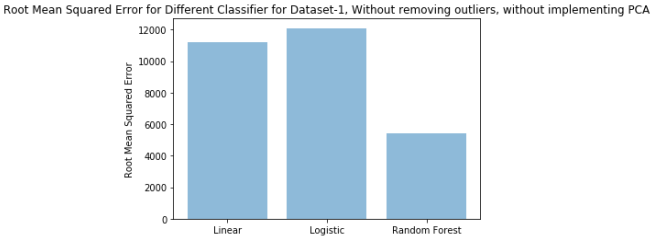

fig-7: Performance comparison on dataset-1(without removing outliers, without implementing PCA) for different Regression Model

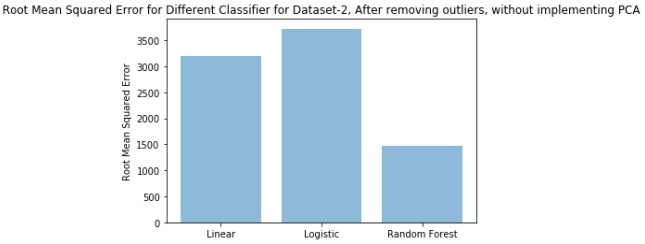

fig-8: Performance comparison on dataset-2(After removing outliers, without implementing PCA) for different Regression Model

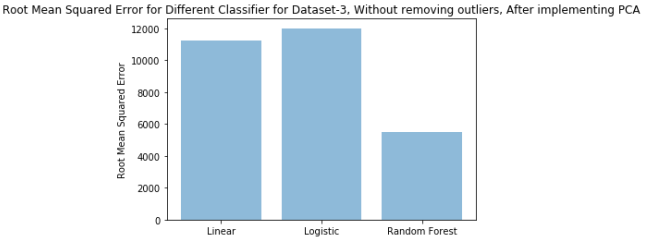

fig-9: Performance comparison on dataset-3(Without removing outliers, after implementing PCA) for different Regression Model

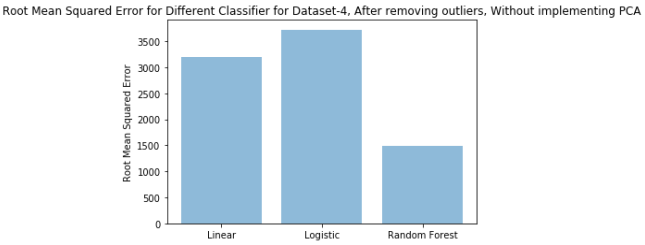

fig-10: Performance comparison on dataset-4(After removing outliers, after implementing PCA) for different Regression Model

A graphical representation of performance of different regression models can be seen from fig-7 to fig-10. It can also be observed that how their performance changes after implementing PCA or after removing outliers. The overall performance rate can be distinguished from table-3.

Table-3: Comparison of regression models in different group of dataset using Root Mean Squared Error

Name	Linear Regression	Logistic Regression	Random Forest Regression
Dataset-1	11179.90	12099.55	5406.83
Dataset-2	3199.77	3724.26	1479.85
Dataset-3	11216.12	11997.64	5468.31
Dataset-4	3199.77	3724.26	1486.02

From table-3. it is observed that for all the models dataset-2 and dataset-4 showed good performance of showing lower rmse which is a clear indication that removing outliers from dataset can improve the performance significantly. From the table-3, if we compare the performance of implementing PCA with before implementing PCA, we do not see any noticeable difference, but after implementing pca the feature set was lowered to 5 which made the computation faster. So, PCA played an important role for making the process quick.

B. Classification.

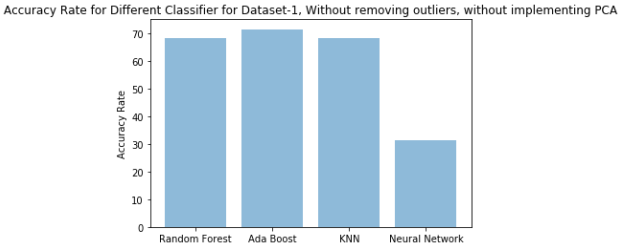

fig-11: Performance comparison on dataset-1(Before removing outliers, before implementing PCA) for different Classification Model

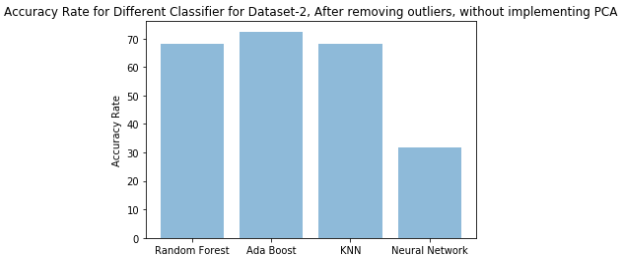

fig-12: Performance comparison on dataset-2(Before removing outliers, before implementing PCA) for different Classification Model

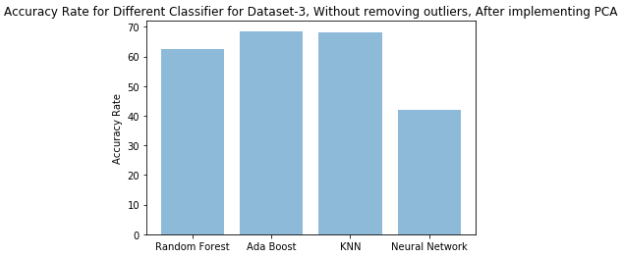

fig-13: Performance comparison on dataset-3(Before removing outliers, After implementing PCA) for different Classification Model

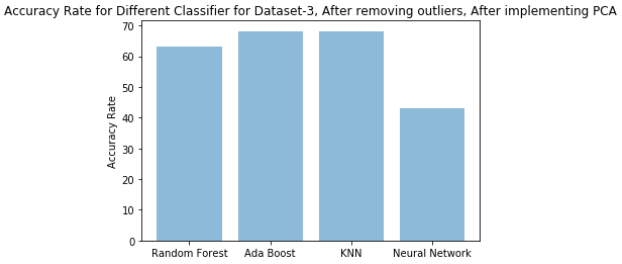

fig-14: Performance comparison on dataset-4(After removing outliers, after implementing PCA) for different Classification Model

A graphical representation of performance of different classification models can be seen from fig-11 to fig-14. It can also be observed that how their performance changes after implementing PCA or after removing outliers. The overall accuracy rate can be distinguished from table-4.

Table-4: Comparison of classification models in different group of dataset using Accuracy Rate

Name	KNN	NN	Random Forest	AdaBoost
Dataset-1	68.23	31.40	68.33	71.60
Dataset-2	68.03	31.80	68.16	72.43
Dataset-3	68.14	41.97	62.67	68.59
Dataset-4	68.10	43.07	63.28	68.18

From table-4, it is noticeable that among all the models AdaBoost showed the best performance.

After that, I have applied the feature selection technique on Adaboost model to observe how it impacts on the performance. For this I chose the dataset-4 where the outliers were removed, but PCA was not implemented since it showed better performance so far.

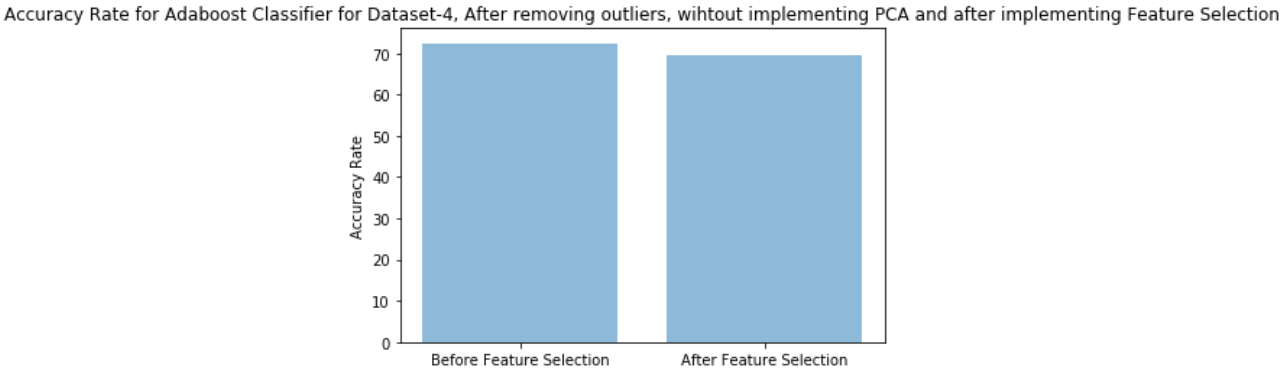

fig-15: Accuracy rate on dataset-4, on AdaBoost model after applying feature selection technique.

From fig-15, it can be perceived that the accuracy did not increase or there was not any significant changes after implementing feature selection, but the training time was reduced. So to make any system faster it can play really a vital role.

VII. Conclusions

Using a dataset of 39,000 instances with 61 features I implemented different models to predict the popularity of online news articles before being published. After implementing several data mining techniques I came to a conclusion that if the outliers can be removed from the dataset it shows a significant performance improvement. And implementing dimensionality reduction and the feature selection technique, it does not improve the accuracy rate but it does reduces the computational time noteworthy. Among several models, the AdaBoost was the one who showed the best performance.

References

[1] K. Fernandes, P. Vinagre and P. Cortez. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.
[2]https://towardsdatascience.com/an-approach-to-choosing-the-number-of-components-in-a-principal-component-analysis-pca-3b9f3d6e73fe
[3]https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba
[4]https://towardsdatascience.com/feature-selection-and-dimensionality-reduction-f488d1a035de
[5]https://medium.com/@syedsadiqalinaqvi/predicting-popularity-of-online-news-articles-a-data-scientists-report-fac298466e

VIII. Conclusions

Using a dataset of 39,000 instances with 61 features I implemented different models to predict the popularity of online news articles before being published. After implementing several data mining techniques I came to a conclusion that if the outliers can be removed from the dataset it shows a significant performance. And implementing dimensionality reduction and the feature selection technique, it does not improve the accuracy rate but it does reduces the computational time noteworthy. Among several models, the AdaBoost was the one who showed the best performance.

References

[1] K. Fernandes, P. Vinagre and P. Cortez. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.

[2] https://towardsdatascience.com/an-approach-to-choosing-the-number-of-components-in-a-principal-component-analysis-pca-3b9f3d6e73fe

[3] https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba

[4] https://towardsdatascience.com/feature-selection-and-dimensionality-reduction-f488d1a035de

[5] https://medium.com/@syedsadiqalinaqvi/predicting-popularity-of-online-news-articles-a-data-scientists-report-fac298466e