1 Introduction

In recent times, predictive analytics has been gaining popularity. One area recently affected by improvements in predictive models is the media sector.

Using the Online News Popularity Dataset from the University of California Irvine Machine Learning Repository, we utilized both regression and binary classification models to predict the popularity of an online article–defined as the number of shares [1]. The main objective of this paper is to identify the best predictive algorithm for regression and classification.

We use the mean squared error (MSE) as the measure for regression model performance and accuracy ($1 - misclassification\ error\ rate$) as the measure for classification performance. We identify the best models as those having the lowest MSE for regression or the highest accuracy for classification.

2 Data acquisition and processing

The Online News Popularity Dataset was published in 2015 by Fernandes, Vinagre, and Cortez [5], containing 39,797 observations with 61 attributes over a two-year period. Our target feature is the number of times one article was shared. Our predictive attributes include numeric features–such as the number of words in the article, the rate of unique words, and NLP sentiment scores–as well as categorical features such as the channel of the article and the day of publication.

To improve the prediction of our models, we performed data cleaning and feature engineering before running our models. As a few examples, we removed entries whose number of words in content equal to zero (hidden missing data). We converted the channel of articles and the day of publication from numeric type to categorical type. Also, we dropped columns that provide redundant information, such as the absolute score of title subjectivity. After cleaning and feature engineering, we are left with 37,905 observations with 38 predictive features and 1 target feature (see the R code for more details on our cleaning process).

3 Regression

After processing the data, 75% of the articles were used as the training set and the remaining 25% were used as the testing set. The approaches we investigated were 1) Multiple Linear Regression, 2) Best Subset Selection, 3) Ridge Regression, 4) LASSO Regression, 5) Elastic Net, 6) Ridge Regression with features selected by LASSO, 7) Regression Trees, 8) Principal Component Regression, and 9) Partial Least Squares.

For Multiple Linear Regression, we first ran the model with all 38 predictive features. The variance inflation factors were then calculated for each feature and those with values higher than 5 were removed, since they are highly correlated with other features in the dataset. Then we ran the model on the remaining features and obtained an anova table displaying each feature's importance. To obtain our final regression model we selected features with significance levels¡ 0.001. The final model produced an MSE of 47,841,968 on our testing set.

For Best Subset Selection, we chose the most significant features using the residual sum of squares (RSS), adjusted R squares, Cp, AIC, and BIC. We then calculated the MSE by dividing the RSS by the number of observations in our testing set. The final model produced an MSE of 485,575,018.

For both Ridge and LASSO Regression, we ran the model on all 38 predictive features and used cross-validation to select the best penalty parameter λ that gives the smallest MSE. We then reran the model

with the best λ and obtained an MSE of 471,99,790 for Ridge and 47,106,652 for LASSO on our testing set. To investigate the LASSO's power of feature selection and the Ridge's power of prediction, we ran Ridge Regression again with features selected by our LASSO model obtaining an MSE of 47,204,218.

For Elastic Net, the values of α and λ in the following formula are selected using 5-fold cross-validation on our training set: $\sum_{i=0}^{n}(y_i - x_i^T\beta)^2 + \lambda[\alpha \parallel \beta \parallel_1 + (1-\alpha) \parallel \beta \parallel_2^2]$. We obtained the lowest training MSE with $\alpha = 0.95$ and $\lambda = 28.32$. The best model was run on our testing set to obtain an MSE of 47,107,300, which is very close to the LASSO.

Regression trees were used with techniques of pruning based on misclassification rate and entropy, bagging, random forests, and boosting which rendered disappointing MSEs of 68,584,173, 59,025,183, and 51,377,281 respectively. However, as we will see later, tree models perform much better for binary classification.

Lastly, Principal Component Regression and Partial Least Squares were used to try and reduce the dimensions of the data. After identifying the optimal number of components which minimised the training cross-validation error, we obtained MSEs of 47,333,013 and 47,312,545 respectively.

As we can see from table 1, LASSO gave the best prediction on our testing set. However, its performance is highly dependent on our training and testing set. Given a different split, we might have come to a different conclusion.

Table 1: Regression evaluation table ordered by MSE

Model	MSE
LASSO	**47,106,652**
Elastic Net	47,107,300
Ridge	47,199,790
Ridge with features selected by LASSO	47,204,218
Partial Least Squares	47,312,545
Principle Component Regression	47,333,013
Linear Regression	47,841,968
Boosting	51,377,281
Random Forest	59,025,183
Bagging	68,584,173
Best Subset Selection	485,575,018

4 Classification

For the purpose of balanced binary classification, we labeled an article as "popular" if its number of shares were above the median number of shares, and "unpopular" if below the median. We created a new categorical column named "pop" as our new target feature, containing two values, "popular" and "unpopular." We compared our models based primarily on accuracy. For thoroughness, we also computed other classification metrics: Precision, Recall, F1 score and the Area Under the ROC (AUC). The approaches we investigated were 1) Stepwise Logistic Regression, 2) K-Nearest Neighbors (KNN), 3) Classification Trees with pruning, 4) Bagging, 5) Random Forest, 6) Generalized Boosting Model, 7)Adaptive Boosting, and 8) Extreme Gradient Boosting with parameter tuning.

For Logistic Regression, we first fit the model on all 38 predictive features and found many insignificant predictor variables. We then built a Stepwise Logistic model to select variables which achieved 65.45% accuracy.

For KNN, we first normalized variables measured in different units to make them comparable. A grid search was used to obtain the best hyperparameter "k" (number of neighbors). When selecting 41 nearest neighbors, the model got the best performance with 63.28% accuracy.

For Classifciation Trees with pruning, we obtained a small tree with three terminal nodes, possibly due to our features having weak explanatory power for our target. Due to the small size of our tree, pruning (either on misclassification rate or entropy) did not render any improvement. Our model produced a disappointing

accuracy result of 61.29%.

For Random Forests, we first set "mtry" (the number of variables selected at each split) to be the number of all predictive features we have (bagging) with 300 trees and achieved 66.8% accuracy. We then ran the model on the default number of features (floor of the square root of 38) with 300 trees which provided an even higher accuracy of 67.39%.

Afterwards, we proceed to learn and implement several boosting techniques, which convert a set of weak learners to a strong learner. Initially, we implemented the general boosting model demonstrated on page 330 of ISLR [2]. The model gives a very high prediction accuracy of 67.31% without any tuning.

Adaptive Boosting (AdaBoost) is a popular boosting algorithm proposed by Yoav Freund and Robert Schapire in 1996. It changes the sample distribution by modifying the weights attached to each observation based on their residuals [3]. When training a new classifier, AdaBoost assigns higher weights to the training observations misclassified by previous classifiers to enhance the prediction power. It continues to add classifiers until a limit of the number of classifiers is reached. We set "mfinal" (the number of basic classifiers) to be 100 resulting in 67.48% accuracy.

Extreme Gradient Boosting (XGBoost) is one of the most popular and recent boosting algorithms for many machine learning competitions. It was developed in 2014 by Tianqi Chen, a PhD student at the University of Washington [4]. Unlike AdaBoost, XGBoost doesn't modify the sample distribution. Instead, the weak learner trains on the remaining errors of the strong learner. Using a gradient descent optimization process at each iteration, it seeks to minimize the overall error of the strong learner. XGBoost's high speed and accuracy is enabled by parallel computing, regularization and cross validation. After tuning several parameters, we obtained an accuracy of 68.25%–the highest among our classification models.

Table 2: Classification models evaluation ordered by accuracy

Model	Accuracy	Precision	Recall	F1	AUC
Extreme Gradient Boosting	**0.683**	**0.691**	**0.724**	**0.707**	**0.738**
Stochastic Gradient Boosting	0.678	0.685	0.728	0.706	0.736
Adaptive Boosting	0.675	0.681	0.725	0.702	0.729
Random Forest	0.674	0.680	0.727	0.703	0.733
Generalized Boosting	0.673	0.682	0.717	0.699	0.733
Bagging	0.668	0.679	0.707	0.692	0.727
Stepwise Logistic Regression	0.655	0.669	0.690	0.679	0.707
KNN	0.633	0.654	0.651	0.653	0.675
Classification Tree	0.613	0.619	0.701	0.657	0.634

5 Conclusion

Among our regression models, LASSO has the best predictive performance, with an MSE of 47,106,651. Among our classification models, Extreme Gradient Boosting has the highest prediction accuracy of 68.25%. With several data cleaning and parameter tuning techniques, our classification models beat the performance of models (67%) used by the original authors of the dataset[5].

Using our best classification model–Extreme Gradient Boosting model, we identified channel, mentioned keyword shares and self reference as the three top most influential predictors. This implies that online news articles with popular keywords and more self reference have a higher chance of gaining popularity. However, without meaningful variables capturing external effects and the actual contents of the articles, our prediction is limited. Nevertheless, the dataset is valuable for us to evaluate different regression and classification models.

References

[1] UCI Machine Learning Repository. *Online News Popularity Dataset*. https://archive.ics.uci.edu/ml/datasets/online+news+popularity, 2015. Last accessed: 11-12-2018.

[2] Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani. *An Introduction to Statistical Learning*. Springer, New York, 2013.

[3] Yoav Freund and Robert E. Schapire. A Short Introduction to Boosting *Journal of Japanese Society for Artificial Intelligence*,14(5):771-780, September 1999.

[4] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. *ArXiv e-prints*,2016.

[5] Kelwin Fernandes, Pedro Vinagre, and Paulo Cortez. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News.*In Progress in Artificial Intelligence*,pages 535–546. Springer, 2015.