# Supervised Learning For Online News Popularity

**Brian Neldon, Khaled Jabr**
brian.d.neldon-1, khaled.jabr-1 @ou.edu

## Abstract

The web has become our main medium for distributing and sharing news. Predicting the popularity of a news article has been a very popular topic, as news outlets are trying to reach out to more potential readers for various reasons. In this work, we present at attempt at this challenge using an dataset from mashable.com, using Ridge Regression, and Support Vector Regression (SVR). Using the two techniques, we built two predictive models to predict how many times a news article will be shared based on a set of features in that article. Our results show that our models were not suitable for this task, as the dataset proved to be complex and irregular.

## Introduction

Supervised learning techniques are best used for studying labeled datasets, and inferring a function or a best fit relationship from the examples and their features. In this project, we will be implementing and running Ridge and Support Vector Regression on an online news articles popularity dataset. We will target 15 specific features to predict the number of shares of an article. We will also use Scikit-Learn implementation of the same algorithms, and then we will compare the results of the two implementations using metrics MSE and MAE, and R-Squared.

Our work was inspired by K. Fernandes, P. Vinagre, and P. Cortez who created the dataset we are using for this project using articles from Mashable.com over a period of two years. The goal of the creating the dataset was to develop an intelligent system capable of predicting the popularity of a news article, i.e. number of shares it will get, before it is published (K. Fernandes et al. 2015). The authors produced 6 categories of features that represented different aspects to look at when studying newspaper articles popularity, totalling 58 predictive variables, and one target, the number of shares. They developed a prediction model for their target using all the predictive features and the following models: Random Forest (RF); Adaptive Boosting (AdaBoost); SVM with a Radial Basis Function (RBF) kernel; K-Nearest Neighbors (KNN) and Naive Bayes (NB). To judge their models performance, they used ROC curves. For our project, we chose a subset of those features to limit the scope of our project. The selection of the features was based on what we were interested in the most, and what fits our regression models, i.e, real valued features. In doing so, it made our project more specific to our interests, however, we will not be able to compare our results to K. Fernandes et al., since they followed a different approach when they analyzed the data.

Another related work to our project is the work done by Goldberg in his thesis, Predicting Arm Motion from Cortical Activity. In his thesis, Goldberg discusses support vector regression using least square kernel reduction. The thesis nicely explains using the moore-penrose pseudo-inverse method to solve linear systems, then it explains how to use this approach, to build least square kernel regression that preserves and expresses the nonlinear transformations in the data, while performing robustly in regards to overfitting (Goldberg 2007). For our work, we used the same approach for implementing support vector regression as presented in Goldberg′s work.

The work by Fu covers bridge regression, which is the family of regression techniques that penalize the coefficients via a regularized loss term. The difference between them is the shrinkage parameter in the loss function:

$$| B_j |^\gamma$$

where $\gamma$ = 1 for lasso, $\gamma$ = 2 for ridge, and $\gamma$ is calculated in other bridge techniques, often using the Newton-Raphson algorithm( Fu 1998). This work went into detail about how changing the shrinkage parameter influenced the geometries of the coefficients. It also introduced the ′shooting method′ for lasso regression, which can be used to update the beta coefficients.

Ng′s paper discusses how to select features in a supervised learning problem with a possibly large amount of useless parameters. He then goes on to prove that L1 regularization makes sample complexity, or the number of training examples required, grow logarithmically in the number of useless features and L2 regularization makes sample complexity grow linearly (Ng 2004).

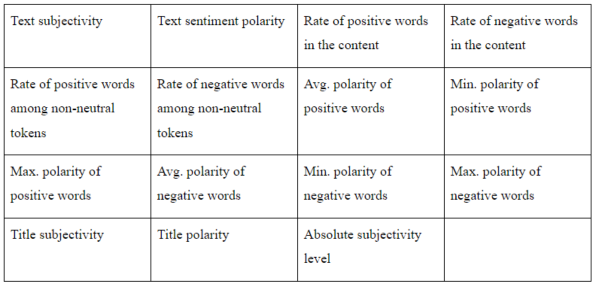
## Methodology

### Learning Model

For this project, we chose the following supervised learning techniques: Support Vector Regression, and Ridge Regression. We implemented our algorithms using the linear algebra and matrix multiplication approach. Our Ridge Regression ran smoothly; however, our Support Vector Regression ran a lot slower due to the fact that we ended up dealing with massive matrices that we had to invert.

## Approach

After we decided on the algorithms that we chose to explore and implement in this project, we shifted our attention towards the dataset. The dataset has 58 predictive predictors, and a total of 39644 examples. After studying the features, we decided to use the following ones:



| | | | |
|---|---|---|---|
| Text subjectivity | Text sentiment polarity | Rate of positive words in the content | Rate of negative words in the content |
| Rate of positive words among non-neutral tokens | Rate of negative words among non-neutral tokens | Avg. polarity of positive words | Min. polarity of positive words |
| Max. polarity of positive words | Avg. polarity of negative words | Min. polarity of negative words | Max. polarity of negative words |
| Title subjectivity | Title polarity | Absolute subjectivity level | |

Table 1: Feature Selection

For SVR, we used 80% (31716) of the examples for training and the remaining 20% (7928) for testing. For Ridge, we used 60% for training, 20% for validation, and 20% for testing.

# Experiments and Results

## Hypothesis and Experimental Setup

Our hypothesis is that we can develop a model that predicts how many times a particular article will be shared based on 15 different features. These variables are common natural language processing measures that try to quantify meaning in text. For our experiments, we used Support Vector Regression and Ridge Regression and carried out 4 experiments: two experiments building the predictive model using our implementations of the algorithms and two using Scikit-Learns. To judge the performance in our experiments, we used the following metrics and calculated them on our test dataset: Mean Square Error (MSE), Mean Absolute Error (MAE), and the coefficient of determination (R-Squared).

## Results and Analysis

The ridge regression metrics were acquired through this process: first, 20% of the data was set aside as the test dataset. Then, we employed a grid search technique where, for 100 iterations, we selected a random 20% of the data as the validation set and ran with a set of 20 different alphas between [0, 1]. For each alpha, the average MSE of the 100 runs was computed. Then the alpha that produced the lowest MSE was used to train on the data again and report the MSE, MAE, and R-Sqaured on the test data. Below in Figure 1 is the average MSE for each alpha ran.

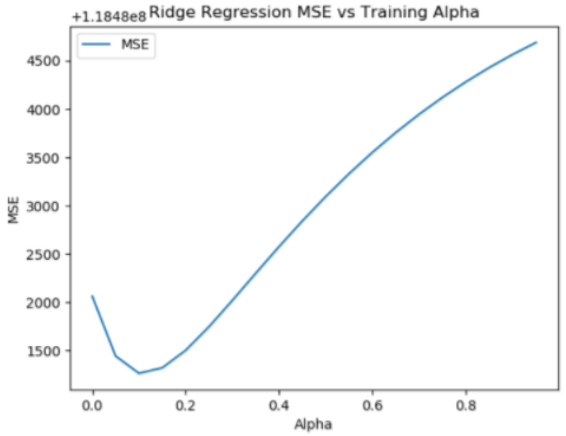
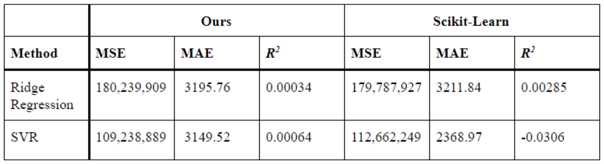


Figure 1: MSE vs. Ridge Regression Alpha

For support vector regression, we were able to run it only once due the massive computational intensity of the algorithm. The reason behind this was in the underlying math and implementation of the algorithm. For the fitting part of the algorithm, it transforms the the input into a higher dimensional space, by calculating a kernel matrix of size 39644 by 39644, where the k(i,j) entry of that matrix is the measure of how similar the ith example in our training dataset to the j example. After calculating the matrix, the algorithm does a couple of matrix multiplications, and then inverts the kernel matrix, which is also very computationally expensive. For the predicting part, the algorithm calculates another kernel matrix, measuring the similarities of the testing examples, and training examples, of size 7928 by 39644. Such computational demanding processing impeded us from our algorithm more than once.

The results of our four experiments are shown in Table 2 :



| | Ours | | | Scikit-Learn | | |
|---|---|---|---|---|---|---|
| **Method** | **MSE** | **MAE** | $R^2$ | **MSE** | **MAE** | $R^2$ |
| Ridge Regression | 180,239,909 | 3195.76 | 0.00034 | 179,787,927 | 3211.84 | 0.00285 |
| SVR | 109,238,889 | 3149.52 | 0.00064 | 112,662,249 | 2368.97 | -0.0306 |

Table 2: Results of Experiments

Looking at the results in Table 2, we can see that the MSE and MAE for our implementation and Scikit-Learns implementation of the algorithms are relatively close. Scikit-Learn still outperforms our algorithms by a small fraction. However, the errors are massive, and we cant infer much more than our predicted values different by a lot from the original target values, which means the our model is probably not the best fit for predicting the change in the original data. To examine and and explain our results more formally, and in relation to our hypothesis, we measured the R-Squared. The coefficient of determination, R-Squared, is a measure of how well our model explains the variation on predicted data in comparison to the total variation in the original data. We can see that both algorithms scored very low R-squared values, which means that none of them explain the variation in the training data. In other word, the models we created produced obsolete prediction models. To help explain why is this the case, we looked into our data and noticed that there was not linear relationship between the features, and the number of shares. Figure 2 helps to explain this.

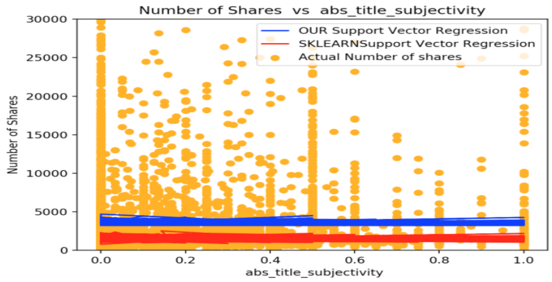


Figure 2: Data is all over the place

We can see from Figure 2 that the data, with respect to the title subjectivity feature, has no linear relationship with our target, number of shares. The is the case with the 14 other features. The date for those features do not have a linear trend with respect to the target.

## Conclusion and Future work

In this work, we implemented Ridge and Support Vector Regression, and attempted to use them to build a linear model that predicts the number of shares that an article would get based on 15 different features. We conclude that our approach for using linear regression techniques failed to develop such models as the data proved to be non-linear, had many outliers, and required much more processing than we initially planned. For further work, we would like to explore other features of the dataset, along with the features that we studied in the project. We believe that we can study the data using classification technique on an array of other features that are available in the dataset.

## References

[1] Goldberg, David Ian. *Predicting Arm Motion from Cortical Activity.* University of Oklahoma, 2007.

[2] 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.

[3] Andrew Y. Ng. 2004. *Feature selection, L1 vs. L2 regularization, and rotational invariance.* In Proceedings of the twenty-first international conference on Machine learning (ICML ’04). 78-.

[4] Fu, W. J. (1998). *Penalized regressions: The bridge versus the lasso.* Journal of Computational and Graphical Statistics, 7(3), 397-416.