Introduction:

Tons of news, stories and articles are published on website everyday. The author or editor would like their articles get shared and referred around the world as many times as it can be. But even the most skillful journalists can't be completely sure that their news directly hit people's tastes, no matter how well organized or gorgeous it might be. There certainly exists a large amount of features contributing to an impressive online news or article. If one can know what kind of news people mostly like prior to the publication, creating an amazing article is just a matter of time and proper modification. Our project aims to develop an effective learning algorithm to predict how many shares an online article (especially news or short stories) would get before its publication by analyzing several statistic characteristics extracted from it. Measurement of popularity is the number of shares an article gets. We use real-world dataset from UCI Machine Learning Repository. Instead of inspecting each single word or phases of the contents, we used some derivations. The input to our algorithm is a large list of features of articles which were published in Mashable: popularity of referenced articles; natural language features (e.g. global subjectivity and polarity); popularity of articles used the same keyword; number of digital media (e.g. images and videos) and published time (e.g. day of the week). We use these features to predict the popularity an article would be prior to its publication.

Previous Works:

Our working idea in this issue is actually derived from a research work of three brilliant students of Stanford University named Xuandong Lei, Xiaoti Hu and Hongsheng Fang. The title of there work was "Is Your Story Going to Spread Like a Virus? Machine Learning Methods for News Popularity Prediction." They used over 39000 articles from Mashable in their research. The dataset they used had 2 non-predictive features, 58 predictive features and 1 output feature. They predicted the exact shares using different regression models (e.g. Regression, GAM, Lasso). The root meen square error of their algorithm was 0.7649.

Our Work:

We have used the same dataset of Mashable used by the previously mentioned research. We have tried to implement their algorithm to predict the exact shares of the article before the article being published. We have tried linear regression which resulted in a root mean squared error of 5077.95615699. As the result is very high comparing with the standard result we assumed by the previous research, we assumed that they may have used some parameter in the linear regression function-we didn't. After that, we have tried a different method name support vector regressor. It worded comparatively better then previous one. But the result is far away from the standard result we assumed. The root mean squared error of this method is now 2186.74038554. Finally we have tried a neural netword based machine learning algorithm named multilayer perceptron regressor. It resulted in the smallest root mean squared error than all the others. First of all, we scaled the dataset by MinMaxScaller and then applied MLPRegressor. We used double layered neural network having unit 10 in each layer. The activation function was 'relu', solver of the weight update was 'adam', and learning_rate_init was 0.05. We have used scikit-learn as the library to solve our problem. The output of this method is given:

Iteration 1, loss = 0.00286731
Iteration 2, loss = 0.00017107

Iteration 3, loss = 0.00015351
Iteration 4, loss = 0.00015146
Iteration 5, loss = 0.00014669
Training loss did not improve more than tol=0.000100 for two consecutive epochs. Stopping.
RMSE: 8.17335709764

Here we represent all the RMSE of the methods we have used for the betterment of the observation.

Methods	RMSE	RMSE of Online News Predictor
Linear Regression	5077.95615699	
Support Vector Regression	2186.74038554	0.7649
MLPRegressor	8.17335709764	