Machine Learning 101: Linear Regression

Victoria Chatron-Michaud, Humayun Khan, Mahad Khan

Abstract

In this project, we assess the performance of Linear Regression models, namely closed-form and gradient descent approach, for predicting the popularity of comments on Reddit. We used the training dataset for a sanity check and to stabilize our model and we then experimented against the validation dataset. We initially started training our model with three features: is-root, controversiality, and children. To lower the mean squared error (MSE), we added 160 features to each comment. These features were the 160 most frequently occuring words in the dataset and the value assigned to each of these features was the number of occurrences of each of these words in the comments. This drastically reduced our MSE. We then added three more features: the word count, the use of hyperlinks and the use of foul words in each comment. Each one of these features further reduced our MSE. As a result of our observations, we found that the closed-form approach has a lower runtime and higher stability compared to the Gradient Descent method, which relied heavily on the choice of hyperparameters and the initial weight vector. Furthermore, the MSE from the closed-form approach was always lower than that from the gradient descent approach and therefore we used the close-form approach to experiment using the validation dataset.

1. Introduction

We investigated the performance of Linear Regression models for predicting the popularity of comments on Reddit, a website where users submit links or write posts about various topics. There are thousands of niche sub-communities (subreddits) that a user can subscribe to and each post in the community can get points when other users vote up (upvote) or vote down (downvote) a post.

While the website has been around for over 10 years, there has been minimal research on predicting popularity of comments on Reddit. In order to fill this gap, we explored methods to predict the popularity of Reddit comments. We would like to recognize Popularity Prediction of Reddit Texts (Rohlin, 2016) as a more in depth version of our experiments, mostly through the use of Latent Dirichlet Allocation. This study adds features such as subreddit specificity and the topics or keywords of the comments in relation to their subreddit, as well as exploring other models.

The ability to predict popularity of a comment is beneficial in various ways. Online influencers can formulate more effective strategies to expand their reach, for example by using predicted popularity to produce content that is more likely to be popular with their

audience. In addition, web services can achieve higher efficiency and provide a better user experience based on prediction. Comments predicted to be more popular can be made more readily accessible to users: Reddit does this by featuring more popular comments at the top of their webpage layout.

We trained and tested for our data set using a linear regression model. This was implemented in two ways, the closed-form solution method and the gradient descent method. For this, we used a preprocessed dataset of Reddit comments from the community r/AskReddit, one of the largest, general question-answering forums on Reddit. The data then underwent feature selection and extraction. We added three additional features that will be described in the dataset portion of the report.

Subsequently, we applied machine learning regression models for the training dataset to train the model. This trained model was then tested on the validation dataset to choose the best performing model and lastly the test set to report our findings.

2. Dataset

The dataset provided was made of 12,000 comments in a dictionary form, with three given features being is-root, controversiality, and children. Preprocessing included splitting the strings into lowercase words separated by white spaces, including punctuation marks. See table 1 for the dataset variables and their descriptions.

We first partitioned the data into sections, for the training set (10000 datapoints), validation set (1000), and test set (1000). Our task was to find the 160 most common words in all of the comments of that dataset and then create a vector, Xcounts, indicating how many times each word appeared in that comment. This was extracted by using the Common class in the collections Python package.

We added three additional features to improve performance, or lower the MSE, of the model. The first was word count, in which we counted the number of words in the comment itself. Our personal experiences on the web indicate that posts or comments with a larger word count tend to be less popular. Microsofts study (2015) suggests that the average attention span of humans has fallen since the start of the century supports this hypothesis.

The second was hyperlinks, giving a value of 0 or 1 if there was a link in the comment. This feature stems from our analysis of the data, in which we observed higher average popularity of comments with hyperlinks, compared to comments with no hyperlinks in them.

The last feature we added was existence of foul language in the comments. We flagged comments with a value of 0 or 1 if they contained any swear words from a list we created observable in the submitted code. Reddit has moderators who police subreddits and there are bots that downvote comments with vulgar language. Users also tend to downvote such comments, so we predict this feature would enhance our model accuracy.

We were aware that ethical concerns might arise when working with a public social media dataset of this variety. The public accessibility of social media data is used to justify its use for research purposes, however, as with other forms of data collection, this poses

ethical concerns. These issues often boil down to consent. For example, Reddit members consent to the use of their public data for research purposes when they sign up, but it is questionable whether informed consent exists given that social media users commonly report that they do not read terms and conditions. We took these concerns into account and used a dataset that did not include any usernames for the posted comments.

3. Results

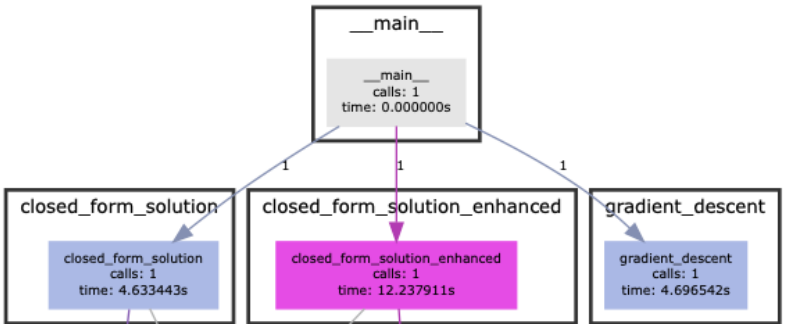

Figure 1: Call graph and time spent in each function

To implement gradient descent, an integral part of creating the model is deciding on the hyperparameters used: being the initial learning rate, initial weights, beta (to control the speed of decay), and precision. The algorithm provided in the description of the assignment was followed exactly, and values for beta and precision were recommended by the teaching assistant and professor to be very small. As initial weights can be either random or all zeroes, we attempted both to find which was best performing. The learning rate which allowed for decreasing MSE with every iteration of gradient descent was 1e-6, which we adjusted via trial and error to get the best value.

	LR = 1e-6	**LR = 7.85e-6**
W initialized as 0	1.05343123	1.04788336
W initialized as random [0,1]	1.10118345	1.04901717

Table 1: Choosing Hyperparameters for GD (Training Set), with 160-word text features

Next was a comparison between performance and stability of the two methods of linear regression, with observations on runtime as well. The closed form method both has a lower MSE and runtime than gradient descent, and was chosen to be used throughout the rest of the experiments for this reason. We also added comments on the stability of the two models, all of which can be seen in figures 4 and 5.

Another feature to be explored was the amount of popular words to be considered in the model. We adjusted the length of Xcounts for two different sizes (60 and 160) and its removal to observe that 160 words is likely the best fitting model, as 60 words or no

text features is underfitting. This confirmed our use of closed form linear regression with 160-word text features. Figures 3 and 4 express the performance of each of the text feature options with the different regression methods and datasets.

We then tested our additional three features to confirm that they improved the performance of the model. These were added on top of the 160-word text features as it was the best performing thus far. The observed details and their improvements to MSE are outlined below.

	Validation Set MSE	**Change in MSE**
160-word text features	0.77926736	0.24023738
Word count	0.77845555	8.1181e-4
Hyperlinks	1.08468307	6.6494e-4
Foul language	1.08468307	3.79627e-3
All additional features	0.77399434	

Table 2: Testing additional features - Closed Form

4. Discussion and Conclusion

An extension we would like to explore is the date and time of the comments made, as the amount of views a comment receives and therefore its popularity can be heavily influenced by the amount of traffic on the site.

As a result of our findings, we can conclude that Linear Regression is a powerful statistical methodology modeling the relationship between a scalar dependent variable y or more explanatory variables denoted X. However, the accuracy of prediction using Linear Regression is actually not satisfactory since it is not a complete description of relationships among variables. It only provides the functionality to investigate on the mean of the dependent variable and the independent variable.

5. Statement of Contributions

Victoria completed all of the text preprocessing and coding for matrix X and vector y creation, as well as implementing the closed form linear regression. She also collaborated with the group to fix the gradient descent model and come up with additional text features. Mahad worked on the gradient descent model and optimized it further. Humayun worked on the closed-form solution and introduced extra features in the model. The entire group collaborated on the report and findings of our model.

6. References

1. T. Rohlin, Predicting the Popularity of Reddit Texts. SJSU ScholarWorks, 2016.
2. Microsoft Attention Spans, Microsoft Corp. Spring 2015

7. Appendix

	Training Set MSE	Validation Set MSE
160-word text features	1.04777632	0.77926736
60-word text features	1.06042914	0.92032371
No text features	1.08468307	1.01950474

Table 3: Text feature inclusion (Closed Form)

	Training Set MSE	Validation Set MSE
160-word text features	1.04788482	0.80385558
60-word text features	1.06044485	0.92782789
No text features	1.08469826	1.02733639

Table 4: Text feature inclusion (Gradient Descent)

	MSE (no text features)	MSE (160-word features)
Closed Form	1.01950474	0.77926736

Table 5: Stability - Closed form consistently outputs the same solution solely based on the values in the matrix X and vector y.

	MSE (no text features)	MSE (160-word features)
Gradient Descent	1.04422636	0.8135488

Table 6: Stability - Gradient descent will provide different results for the output weights w given the choice of hyperparameters, as well as the possible randomization of the initial weights w0. Because of this, we may observe a different MSEs with every iteration of gradient descent.

Training Set MSE	Validation Set MSE	Test Set MSE
1.04653672	0.77399434	1.05063218

Table 7: Best performing model