COMP 551 MINIPROJECT I : MACHINE LEARNING 101

OWEN LEWIS - 260706614,
BEATRICE LOPEZ - 260654565, AND
MINH ANH NGUYEN - 260671180

January 31, 2019

1. Abstract

In this project, we sought to predict comment popularity on Reddit through linear regression models and to investigate the performance of these models. We partitioned our dataset into training, validation, and test sets in order to build, compare, and report on our models effectively, using the given features as well as additional text and non-text ones that we ourselves extracted. We found that the closed-form solution for linear regression is stable, always resulting in the same solution, while the results of gradient descent vary according to the hyperparameters (i.e. learning weight, initialization). For the implementation of gradient descent, we found that the learning rate should be scaled according to the size of the input data used – if the input matrix is larger, the learning rate should be smaller and have a slower decay speed. The projects best performing model was built on the top 160 word count features, and this was improved by our interaction term $is\_root \times children$ and our basis expansion term $children^2$.

2. Introduction

This modern age of online communities, likes and trolls provide much data for us to analyze and draw conclusions on the workings of not only the Internets communities as subcultures, but also of general modern-day society – for its benefit or not. Popularity prediction, for example, would be a great source of interest for our friends in marketing and consumer product development. For this project, we were given a preprocessed dataset of Reddit comments from the community r/AskReddit. We were tasked to predict comment popularity on Reddit by building and exploring linear regression models on this data. Success in this could lead to improvement in Reddits infrastructure and better user experience. While the results of our experiments were generally consistent with the material covered in lecture as well as our personal intuition, we see that the models built are relatively simplistic in light of the potential complexity of the problem of popularity prediction on Reddit comments. As well, our best model still performed worse on the test set than the validation set. As Rohlin notes in "Popularity Prediction of Reddit Texts,"[1] the prediction problem can indeed be difficult, especially within more general communities such as r/AskReddit in comparison to more niche communities. In her paper, Rohlin describes a similar project based on models built through methods other than linear regression, such as the Naive Bayes classifier and Support-vector machine, which makes for interesting comparison with this project.

In building our models, we worked with the 3 simple features given (*is_root*, *controversiality*, and *children*) as well as the word count features we were asked to implement. This involved finding the top 160 words of the training data comments and their word counts. In addition to these features, we extracted the 5 following features: (1) $is\_root \times children$, (2) $children^2$, (3) number of words per sentence, (4) number of sentences in a comment, and (5) average number of words per sentence in a comment.

The project allowed us to explore the differences between the closed-form solution of linear regression and gradient descent. We found that the closed-form solution is more stable, always resulting in the same solution. In comparison, gradient descent we found to be unstable, returning varied solutions depending on the hyperparameters of the learning rate, α, which is built on the scale η_0, and β, which controls the speed of the decay. We had to try different values for the learning rate until we found one that led to

[1] Rohlin, Tracy. "Popularity Prediction of Reddit Texts" (2016). Masters Theses. 4704.

convergence. We observed that the learning rate should be inversely proportional to the the size of the data used, i.e. if the input matrix is larger, the learning rate is smaller and have a slower decay rate. Using these implementations of linear regression, we built various models on different combinations of the features. Through our validation set, we were able to compare these models and identify the best one. The project's best performing model was that of the top 160 word count features and this was improved by our interaction term $is\_root \times children$ and our basis expansion $children^2$.

3. Data Set

The dataset comprised of 12,000 data points, which we partitioned into subsets of 10,000 data points for training the models, 1,000 for validation, and 1,000 for testing. The data points each consist of the raw text from a given comment, the number of children of the comment, Reddit's proprietary measure of the controversiality of the comment, a Boolean value indicating whether the comment is a root comment or child comment, and the target popularity score of the comment. As tasked, we extracted the top 160 words and their word counts from the training data. In addition to this, we extracted 3 new text features. The first is a simple count of the number of words in a comment. Similarly, the second text feature is a sentence count of a comment. The third is the average number of words per sentence in a comment.

Besides these text features, our team also designed two other non-text features. The first is an interaction term which multiplies the is_root binary value and the integer number of children $is\_root \times children$. The second is a basic expansion term that squares the number of children $children^2$.

A few ethical issues may arise in working with such a public social media dataset as this. The most obvious possible ethical issue would be the infringement of privacy of Reddit users. Depending on Reddit's bylaws, such work could infringe upon their policies. The ability to engineer popular comments could also be used for nefarious purposes such as advertising in communities popular among younger users.

4. Results

4.1. **Task 3.1.** The below table summarizes our comparison of the closed-form solution linear regression and gradient descent approaches. We observe that in this case the closed-form approach has a faster runtime, is stable, and performs better than gradient descent by 0.00001402499. However, the closed-form method has a slightly higher bias. Gradient descent is not stable as it gives different results as we vary the learning rate. For our training set, of size 10,000, the learning rate started at 9.99900×10^{-6}, and decayed by about 4.9965×10^{-9} per iteration, until it stopped at 1.90909×10^{-6} for convergence. For the validation set, which is 10 times smaller than the training set, we found that the learning rate was about 10 times larger and its rate of decay was about 10 times faster.

	Closed-form solution	**Gradient Descent**
Runtime	15.8 ms	2.53 s
Performance	1.0846830709157251	1.0846970959155506

Table 1. Results of linear regression models built on *is_root*, *controversiality*, and *children* features of training set

4.2. **Task 3.2.** The below table demonstrates that none of these models seem to suffer from overfitting or underfitting as each performs better on the validation set than the training set. We also see that the models improve as more features are added.

	Model 2.1 (base)	**Model 2.2**	**Model 2.3**
Features	No text features	Top 60 words	Top 160 words
MSE (training set)	1.0846830709157251	1.060429141685383	1.0477763217987115
MSE (validation set)	1.0195047390212957	0.9196270063580609	0.784336216609445

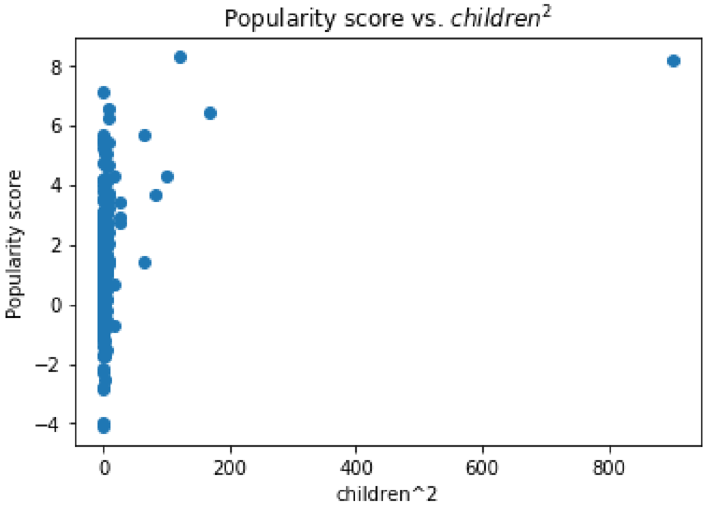
TABLE 2. Results of 3 models built through closed-form solution linear regression on training set and validation set

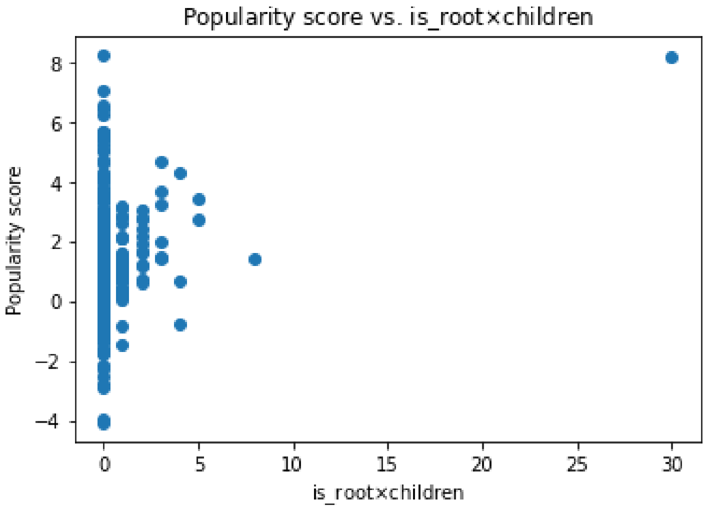
4.3. **Task 3.3.** As shown below, our additional features improve the model both individually and combined. The basis expansion feature improves performance more than the interaction term, but together the features improve the model even further. In addition to these 2 additional features, our team designed 3 other additional text features (number of words, number of sentences, and average number of words per sentence), but these features only improved the model minimally (at best, 0.001) and so the results of their models are not reported here.

	Model 3.1 (base)	**Model 3.2**
Features	Top 160 words	$is\_root \times children$
MSE (validation set)	0.7843362166094	0.7789275613176
Improvement		0.0054086552917
	Model 3.3	**Model 3.4**
Features	$children^2$	$is\_root \times children$ and $children^2$
MSE (validation set)	0.7729657390639	0.7720563430855
Improvement	0.0113704775455	0.01227987352386

TABLE 3. Improvement on best closed-form solution linear gradient descent model on validation set

4.4. **Task 3.4.** Our best model is built through closed-form linear regression and uses all of the following features: is_root, $controversiality$, $children$, top 160 word occurrences, $is\_root \times children$, and $children^2$. It returns an MSE of 1.211145521413 on the test set.





5. Discussion & Conclusion

Through this project, we confirmed our hypothesis that Reddit comments that have more children and is a root comment tend to be more popular. This intuition allowed us to improve the original models given in the project specifications. In Task 3.1, we also noticed that even though gradient descent is theoretically built to be more efficient, the closed-form solution performed faster, likely due to the fact the matrices were a reasonably small size (closed-form solution is more expensive due to the matrix operations). Finally, in exploring gradient descent, we learned that the learning rate must be inversely proportional to the size of the input data in order to ensure convergence. For further research, a larger data set, including the date and time of comments, could lead to the study of other correlations with comment popularity and stronger results. Another hypothesis we could explore is to classify the topic of discussion of a given dataset of comments.

6. Statement of Contribution

The breakdown of the workload was as follows: Lewis handled the majority of task 1, writing functions to process the text, extract the various text features we designed as a team, and return all output in the right format to be fed into the linear regression algorithms, as well as typesetting the writeup. Nguyen took the lead in implementing the closed-form solution and gradient descent with Lopez coming alongside her in building the various models, running the experiments, and summarizing the project in this write-up.