Douglas Hummel-Price

Massive Data Fundamentals

Final Project

April 29, 2020

Executive Summary

Reddit provides a forum for internet users to discuss nearly anything. A key function of the website, Reddit allows users to “upvote” and “downvote” posts, which allows better, or at least more popular, ideas and posts to rise to the top of the page, quite literally. Reddit also allows users to make edits to their posts after they have done so. This project examines the question “Does the popularity of a post, whether positive or negative, impact the likelihood that a user edits a post after it receives a certain amount of attention. Ultimately, the log of the absolute value of a score has some predictive power in a logistic regression, but still results in a fairly low AUC in this data set (Top model AUC = 0.627).

Hypothesis:

The target for the modeling is a dichotomous variable indicating whether a post has been edited. The goal is to determine what kind of impact, if any, the “scoring” variable has on the likelihood that a post is edited. The hypothesis is that higher scores are more likely to prompt edits, so a logistical model with score alone should perform only marginally worse than score + controls.

Tools:

Due to the size of the data (473 million posts), the analysis will be done using Spark. Specifically, the modelling is done using Pyspark and SparkML on an AWS EMR cluster, with some visualization done via simple pandas and matplotlib plotting libraries on my local machine. In both the data cleaning stage and the modeling stage, I started with the smallest reasonable sample of the data that I could, ultimately scaling up to the full data set. Much of these earlier trials have been eliminated from the code to clean up the notebooks. As I scaled up the size of the data and increased the sophistication of my analysis, I scaled my AWS instances up from starting instances of 1 master + 2 or 4 cores (all m5.xlarge). The final run of everything used 1 master and 30 cores, which dropped the runtime of the entire notebook to around 35 minutes. Since this scaling is costly, most of the work was demoed in the smaller instances, and all of the cores except the master were done with spot-pricing.

Data Preparation:

Our target, the variable “edited” shows “false” if the post has not been edited, or the date of the most recent edit in UTC format. To convert it to a binary categorization problem, I converted this to a binary variable where 1 = Has been edited and 0 = has not been edited.

Our feature of interest, “score” is a continuous integer ranging from -22280 to 90192. We will use the following variants of the variable:

* Model 1: No Adjustments + controls
* Model 2: Absolute value of the score + controls
* Model 3: Logged Absolute value of the score + controls
* Model 4: No adjustments, no controls
* Model 5: Absolute Value, no controls
* Model 6: Logged Absolute Value, no controls
* Model 7: Controls only

Fifteen of the columns contained missing values. I don’t feel comfortable enough with the dataset to impute the missing values, so I dropped them. They are:

"archived", "author\_cakeday", "author\_created\_utc", "author\_flair\_background\_color", "author\_flair\_css\_class", "author\_flair\_richtext", "author\_flair\_template\_id", "author\_flair\_text", "author\_flair\_text\_color", "author\_flair\_type", "author\_fullname", "author\_patreon\_flair", "collapsed\_reason", "distinguished", and "removal\_reason".

The following variables have high cardinality, and so will impose substantial overhead. The behavior I am examining should be subreddit,user, and post agnostic, so I am dropped the following:

"author", "body", "subreddit", "subreddit\_id", "subreddit\_name\_prefixed", "id","link\_id", "parent\_id", "permalink", "gildings".

Since the posts are all from the same 4-month period from two years ago, I do not think the following variables will enough useful information to warrant inclusion:

"created\_utc", "retrieved\_on"

The variable “can\_mod\_post” is entirely 0s so there is no explantory power from it, so it is dropped.

The following variables are Boolean, and are converted to 1 = True and 0 = False by casting to IntegerType: "can\_gild", "can\_mod\_post", "collapsed", "is\_submitter", "no\_follow", "send\_replies", and "stickied".

The variable “controversiality” is either a 0 or 1, where 1 indicates a post that has similar numbers of upvotes and downvotes, presumably beyond some threshold of total votes to distinguish between a post with 0 votes and a post with 1,000 upvotes and 1,000 downvotes, though this threshold is not documented publically anywhere.

The variable “gilded” is a continuous integer ranging from 0 to 66 to represent how many times a post has been gilded.

The variable “subreddit\_type” is categorical and so will require being put through the StringIndexer and then the OneHotEncoderEstimator.

EDA:

The remaining columns have descriptive statistics, provided by the .describe() method, copied to notepad, and readjusted to save as a csv file delimited by the | character. The description table is found as the file “Pretty\_Descriptives.csv”, and it’s also displayed in the primary modeling notebook (dbh52\_Final\_Project\_Cleaning\_and\_modeling.ipynb).

The descriptives statistics, as well as visual representations of the distributions for each variable, is contained in the “Visualization Notebook.ipynb” file.

A few comments:

1. “Edited” has a fairly unbalanced distribution, with over 97% of the posts never having been edited.

2. As seen in the graph showing the distribution of the Score variable, nearly all scores hover right around 0. This makes sense, given how Reddit’s system of voting works.

3. Logging the score variable provides a slightly better distribution.

4. There are only 4 categories of subreddit type, and the vast majority are public.

Modeling:

To examine the impact of score on editing, I ran a total of 7 logistic regressions, all of which have “edited” as the target. The combinations are found near the bottom of the primary modeling notebook. The primary model contains:

["can\_gild", "collapsed", "is\_submitter","send\_replies","stickied", "no\_follow", "controversiality", "gilded","score","subreddit\_type\_O"]

From there, the next two models swap out score for the absolute value of the score (model 2) and the log of the aboslute value of the score (model 3).

The next three models examine just the impact of the three score variables, and the final model only examines the non-score variables. To evaluate the efficacy of the models, I evaluated both the AUC and the Accuracy scores. These are provided in the csv “AUC\_and\_Accuracy.csv” and are also included in the both the modeling notebook and the visualization notebook.

Results:

The model with the highest AUC was the logged score plus controls, followed by the absolute value plus controls, then the ordinary score plus controls. This is not surprising – the logged score is more evenly distributed than the others, so the model has more to “chew on”. Surprisingly, the controls-only model performed almost as well as the previous three models. All four of these models had an AUC of around 0.61 or 0.62. The three models with just the scores had AUCs in the .53-.57 range. These lower scores combined with the higher score of just controls suggests that the controls are doing the bulk of the work in these models. Score has a marginal impact, but ultimately, the constellation of controls predicts better.

All 7 models had accuracy scores around 97.2%. While this initially seems promising, closer examination shows that there are approximately 97.2% observations that have not been edited. Therefore, the accuracy scores might just result from calling all of the observations not edited.

Conclusion:

The score for a Reddit post has a marginal impact on whether someone decides to edit their post. However, the impact is not large enough to meaningful increase predictive power.