Stack Exchange Posts Analysis

for the Partial Fulfillment of the Requirements of the course

STAT 6021-001

A Project Report
Submitted by-
Chloe Yan - ly6pc
G Haritha - hg5mn
Shengua Wu - sw2hg
Andi Yu- ay6wy

Data Science Institute

August 2019

Table of content

Abstract:

Question and answer sites like Stack Exchange network allow users to contribute knowledge in a variety of topics. Of particular interest are the posts with higher scores and questions with more answers. We performed an exploratory analysis of the data to determine how to get high scores on answers and how to get your question answered. Our results show that the statistics community in Stack exchange is pretty active as the probability of questions getting answered is high. Our findings confirm the odds of questions being answered.

1. Introduction

The posts data set was sourced from Stack Exchange Data Explorer, which contains 11 variables and 50,000 observations. Since we will only explore on these variables that will affect our prediction/model, we will discard some variables related to name, ID and some irrelevant dates. Here are the variables we use:

Variables	**Description**
PostTypeId	The type of the post(1: Question, 2: Answer, etc)
CreationDate	The creation date of the post
Score	The score of the post
ViewCount	The count of the views of the post(only present for Question)
Body	The content of the post
LastEditDate	The last edit date of the post
LastActivityDate	The last activity date of the post
Tags	The topics of the post(only present for Question)
AnswerCount	The count of the answers(only present for Question)
CommentCount	The comment received for the post
FavoriteCount	The favorite received for the post

The first thing we did after got our data, we checked if the data is completed or not. Unfortunately, there is a lot of missing values.After thorough inspection of the data, we observed following interesting things about our dataset.

a. Few variables are only present for PostTypeId 1. For instance, ViewCount is only available for questions.
b. There are many Id column which we know would not affect the nature of the posts.
c. There are many variables that need to be preprocessed before we use them in modelling. Like Body CreationDate LastActivityDate and Tags

2. Deliverables

Since the data set has few variables present only for question datatype, we decided to split the data set and consider them as two different datasets.

a. Question dataset: We have a lot of missing data for the FavouriteCount. To fill in these missing data, we will make our linear model based on these posts with count of favorites and make a prediction. Then, we will explore what variables will determine if a question will be answered, and predict what kind of questions will be answered.
b. Answer dataset: We are curious about what variables will affect the score of the answer, and how can we predict the score of the answer. After this, we are able to reach what is a good answer.

3. Feature engineering

A. Data - Cleaning:

Every data analysis starts with experience. We checked the whole dataset with 22 variables, and decided to discard some of variables that will not help us to get our approach, ex. Id, AcceptedAnswerId, ParentId, DeletionDate, OwnerUserId, OwnerDisplayName, LastEditorUserId, LastEditorDisplayName, Title, ClosedDate, CommunityOwnedDate. After filtering out these variables, we got our dataset with 11 variables(mentioned in Introduction).

WEAfter that, we decided to only watch the PostTypeID = 1 or 2, which are Question and Answer. We split the whole dataset to 2 datasets based on post types, so we discard all the data not included in the Question and Answer.

B. Feature - Reconstruction:

In order to add the content of the post(body) to our model, we use lines of body(body lines) to stand for the content. Aiming to reduce the dimension of predict matrix, we want to combine two qualitative data by subtracting creation date from last activity date, to one quantitative features called duration. However, we realized that for all dates in this dataset, they are originally factors. So we have to transform all the factors to strings, then made them to date form, so we are able to subtract them. Also, there is a variable called LastEditDate. We treat it the same as the previous date and convert the values to 0 if it does not have a LastEditDate, to 1 if it has a LastEditDate. What’s more, we also transferred AnswerCount to a binary variable showing if the question got answered (0 for not answered, 1 for answered).

C. Feature - selection

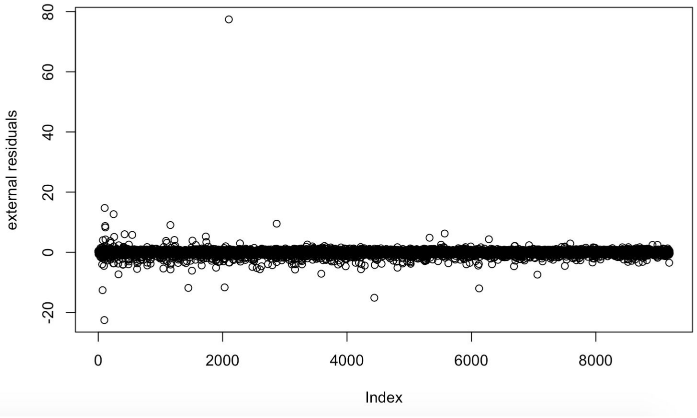
In this special case, after data washing and feature reconstruction, we think rest of data are important overall.

Variables	**Description**
PostTypeId	The type of the post with 2 values(1: Question, 2: Answer)
Score	The score of the post
ViewCount	The count of the views of the post(only present for Question)
Bodyline	The number of lines of the contents
hasEdited	If the post was edited(1: edited, 0: not edited)
Duration	The duration of the post being active
Tags	The topics of the post(only present for Question)
hasAnswer	If the question got answered (1: answered, 0: not answered)(only present for Question)
CommentCount	The comment received for the post
FavoriteCount	The favorite received for the post

4. Model Analysis(Question dataset)

Question dataset has CommentCount, Duration, BodyLine, Score, ViewCount, FavoriteCount and hasEdited as predictors and hasAnswer as the response. We are interested to see what makes a question answered and how someone must post their questions to make sure they get answered. Hence, we decided to do logistic regression and find the probability of questions being answered.

Since the FavoriteCount for each post has a lot of missing values which count about 40 percent of the variable, so we decide to build a linear model to predict our FavoriteCount based on post's score, Viewcount, CommentCount, timeDuration, bodyLine and edited or not. We first split our question dataset into two parts based on if the FavoriteCount is missing for that post. Then we build our model in the complete data set. Our linear model has overall p-value around 0, and R-square around 90.5%. One of our predictors called bodyLine is not significant, so we exclude it from our model. Our anova test can't reject out null hypothesis that the reduced model is better, we kept using this model to make predictions on FavoriteCount. After filling all missing values in the FavoriteCount for the incomplete data set, we row bind these two data set to get a full one.

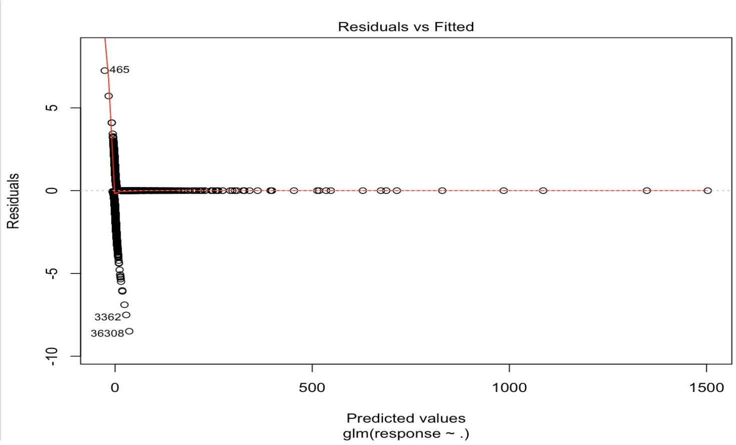


Our normality assumption is satisfied, so we are confident about our predicted FavoriteCount and decided to fill in our missing values and use them for future prediction.

After we fill the missing value and transformed our predictors into the form we think will be useful for our prediction. We built our logistic regression by using all the predictors. We did likelihood ratio test to compare it to the null model, which turns out that the model is pretty significant. Since we found a relatively high correlation between CommentCount and Score, we decided to keep one in the model and built a reduced model excluding CommentCount. However, the likelihood ratio test suggests that we should use the full model. Hence we decided to use the model with all predictors.

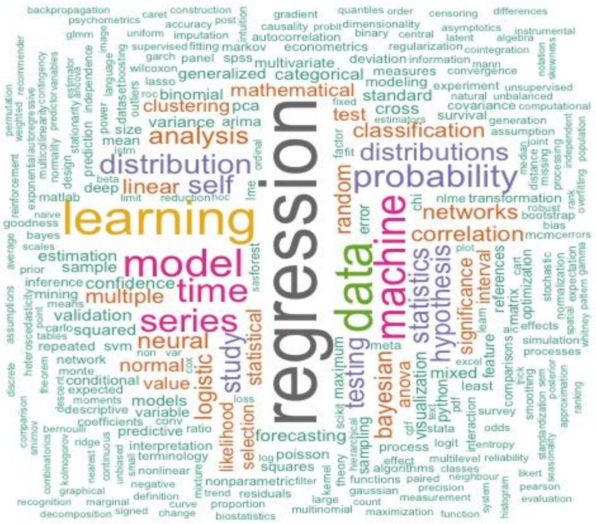
$$
\begin{aligned}
&log(\frac{p}{1-p}) = -0.5 + 1.07 \times Score + 0.001 \times ViewCount \\
&- 0.19 \times CommentCount - 0.99 \times favoriateCount \\
&- 0.0006 \times timeDuration - 0.03 \times bodyLine - 0.08 \times editedYes
\end{aligned}
$$

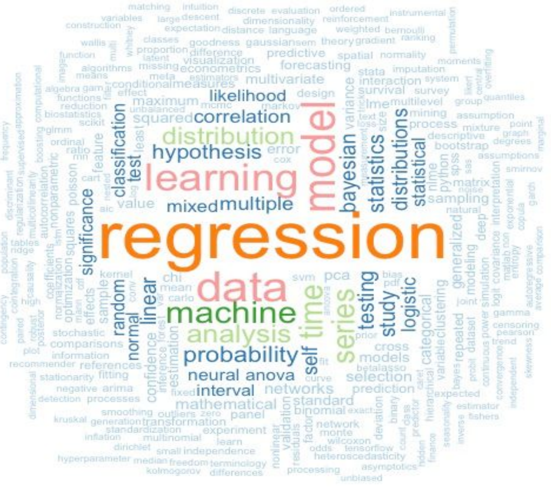
Basically, more ViewCount and higher score will make the probability the answer to be higher, commentCount, favoriteCount, timeDuration, bodyLine and if edited or not will lead to the opposite way when each of them increase by one unit. Based on the model we got, we think people might favorite the post because they want to keep track of the post they have no answer for. Also, people make comments under the post to discussion the question but can’t figure out the right answer might be another reason it does not help with getting an answer.



5. Text Mining

After getting a sense of how to get answers for posting and how to get high-quality answer, we wondered what are the top words in those not answered questions and answered questions. We used the variable called tag in our question data set using text mining to get the top keywords in those questions. On the left hand side, we have a text cloud for those answered questions, and on the right hand side, we got a cloud for those questions with no answers. We can see the most frequently appeared words are regression, model, learning and data. However, if we look down into the less frequent words, we noticed that people are more willing to answer questions about neural networks, time series who are the hottest topics in the data analysis field, and topics like hypothesis, anova are questions that people don't want to answer. It probably because those questions are too easy and tedious to answer, or it is even harder to answer those basic ideas thoroughly.



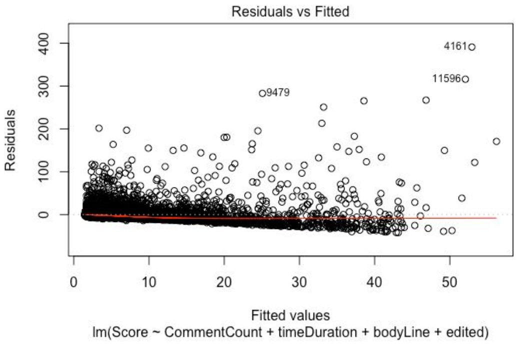


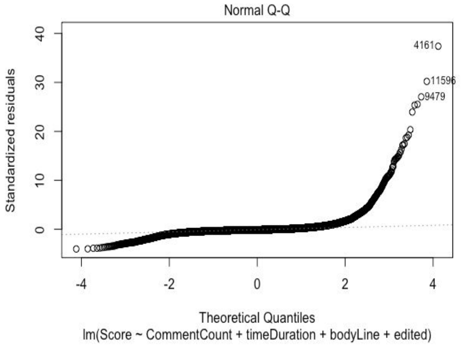
6. Model Analysis(Answer dataset)

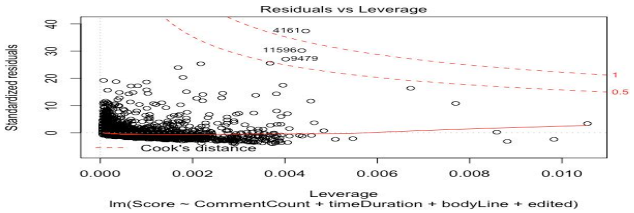
Our answer dataset has CommentCount, TimeDuration, BodyLine and edited as predictors and Score is the response. As the response and most of our variables are quantitative variables, we decided to do linear regression for answer dataset. We tried to regress score over the exact variables to understand the dataset and check for any influential points.

A. Model 1:

For our first model, we plotted Residuals versus Fitted values to see the nature of residuals. Obviously, the residuals are not randomly distributed from the diagnostic plot and there is a heavy tail in the normal Q-Q plot. Hence, we need to do transformations on variables or/and response.







From the above plot, we can see that there are few influential points who fall out the Cook's distance threshold like 4161,11596, 9479. We dropped these observations from our dataset.

B. Model 2 - Poisson Model

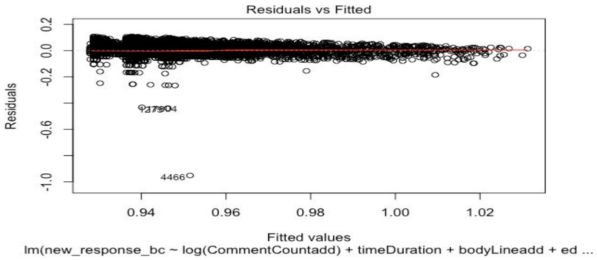
We then tried a poisson model to make prediction since we have inconsistent residuals. Again, we plotted fitted values versus residuals for this model to see if the poisson model maks any improvement from the previous model. But it turns out that our model is significant but we still did not solve the homoscedasticity, we need to try other models.

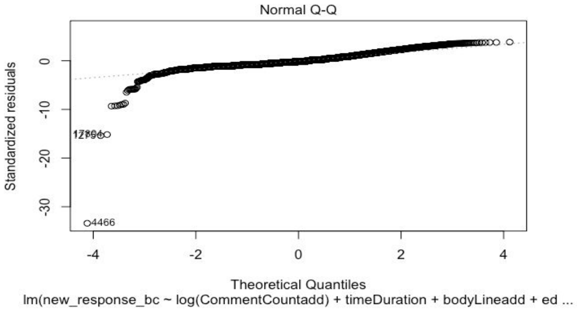
C. Box-Cox Model using poisson regression

Box cox model is used to transform our response variable after some transformations in the predictors. We got the best lambda equals -0.95 then regressed transformed response over other predictors using poisson regression.

D. Final Model.

Since poisson regression does not do a good job on modeling, we tried to do some transformation to our regressors to make a better model. As CommentCount and number of body lines have zero values in it, we added 0.0001 to all the values to be able to apply log transformation on these variables. After trying and testing various models using all the permutations and commutations like normality and homoscedasticity test, we decided to stick with our final model which has transformed response through box-cox and log transformed commentcount.





7. Results

In order to get a better sense of how our logistic regression performs in our data set, we decide to use cross-validation to split our data into 70% training and 30% testing data. We built the same logistic regression from the training dataset and predict the binary output to make comparison with the true value. Then we make a confusion matrix for our logistic regression to get its prediction accuracy.

The numbers in the diagonals are the cases we got right from logistic prediction, and the numbers off-diagonal are the missed cases. We then tried to get the accuracy by dividing the the correct prediction by total number of observations.

	Predicted: NO	Predicted: YES
Actual: NO	1450	450
Actual: YES	526	4597

a. Question Dataset:

We then tried several machine-learning algorithms like random forest, adaboost and svm. Random Forest produced the highest accuracy. It turns out that our logistic regression did a good job on prediction since all models have similar accuracy.

Model	Accuracy
Logistic Regression	86%
Random Forest	91%
AdaBoost	90.8%
SVM	86%

b. Answer Dataset:

Final Betas:

Variable	Estimated Beta
CommentCount	1
timeDuration	0.00002.264
bodyLine	0.0007.633
hasEdited	0.007.658

From the table, we can see all coefficients are positive, hence we can conclude that timeDuration, number of body lines, make edition have positive effect on getting higher answer scores. Among all significant regressors, we can also conclude that comment counts play the most important role in predicting the post score since it has the largest estimated accuracy.

8. Conclusion

Q & A sites like StackExchange bring people from diverse backgrounds together to be able to share knowledge on an unlimited number of topics. Stack Exchange fundamentally shifted how programmers problem solve and distribute knowledge. Through our analysis, we statistically explored how a user can get his/her answered, what are the major things to remember when posting a question. And how to get a good score for someone posting the answer. From our findings we can conclude that timeDuration, bodyLine, hasEdited, CommentCount have positive effect on getting high scores. Also, through text mining, we can see that the posts with tags like regression, machine learning, model, time series, are more frequent. Which gives us an idea of current trends in the market. From the betas we got from logistic regression, we can conclude that viewCount, score have positive effect on whether your question gets answered or not. Whereas, more bodyLines, CommentCount decreases the probability of the question being answered. For future work, we should gather more information about the user's background like its reputation or its active days in stack exchange to get a better prediction and even more give practical suggestions to users who want to get the good quality answers they want after they post.

9. Reference

"Database Schema Documentation for the Public Data Dump and SEDE." *Meta Stack Exchange*, 1 Jan.1960,meta.stackexchange.com/questions/2677/database-schema-documentation-for-the-public-data-dump-and-sede.