Advanced Data Analysis Final Project

Team 2:

Duanhong Gao (dg2896)

Yi Jian (yj2376)

Jingwei Li (jl4549)

Aoyuan Liao (al3468)

Hanqing Shi (hs2871)

Jia Wang (jw3315)

Xiangyu Wu (xw2423)

Yutong Yang (yy2624)

Wanyi Zhang (wz2323)

Fall 2016

**I. Introduction**

The second part we try to predict question scores based on some features. From part 1’s conclusion, we all hope that our problems can get higher score so that it can be resolved better. But how to make it happen?

We are interested in two response variables: the score of the question and whether a question gets resolved.

Then we applied exploratory data analysis to the score and time then we fitted logistic regression on score and time respectively. We also did xgboost to find out important features towards score and time.

**II. Data Description**

We have two sets of data, one contains questions and answers from Stack Overflow that are tagged with the [r tag](http://stackoverflow.com/questions/tagged/r), another one contains questions and answers from Stack Overflow that are tagged with the [python tag](http://stackoverflow.com/questions/tagged/r).

***‘R Questions from Stack Overflow’*** data set is organized as three tables:

* **Questions** contains the title, body, creation date, score, and owner ID for each R question.
* **Answers** contains the body, creation date, score, and owner ID for each of the answers to these questions. The ParentId column links back to the Questions table.
* **Tags** contains the tags on each question *besides* the R tag.

***‘Python Questions from Stack Overflow’*** data set is also organized as three tables:

* **Questions** contains the title, body, creation date, score, and owner ID for each Python question.
* **Answers** contains the body, creation date, score, and owner ID for each of the answers to these questions. The ParentId column links back to the Questions table.
* **Tags** contains the tags on each question *besides* the Python tag.

For space reasons only non-deleted and non-closed content are included in the dataset. Both datasets contain questions up to **19 October 2016** (UTC).

(data description for first two parts of this project)

In the last part of our project, we only use ***‘R Questions from Stack Overflow’*** data setto do tag recommendation, since the methodology is the same for different data sets. And we only use ‘**Questions**’ table and ‘**Tags**’ table in this data set. For ‘**Tags**’ table, we delete tags which show up less than five times during the whole time period. We randomly split ‘**Questions**’ table into training (80%) and testing (20%) sets.

**III. Methodology**

Firstly we applied exploratory data analysis to answer score and time variables to have a big picture of them.

From the histograms, we can see that the scores of questions are highly skewed. Based on that, we define that a question gets resolved when there is a corresponding answer with at least 3 scores, which is the 3rd quantile of the answer score distribution. Therefore, we can label each question as 0 or 1; 0 being a question without good asnwers and 1 being a question with good answers.

Naturally, we applied a logistic regression to predict question’s label. It shows that if the question score gets higher, the question will be more likely to get solved.

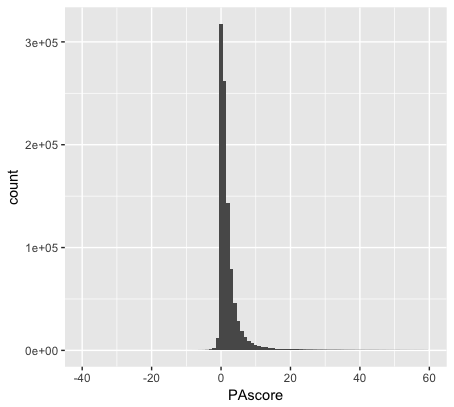
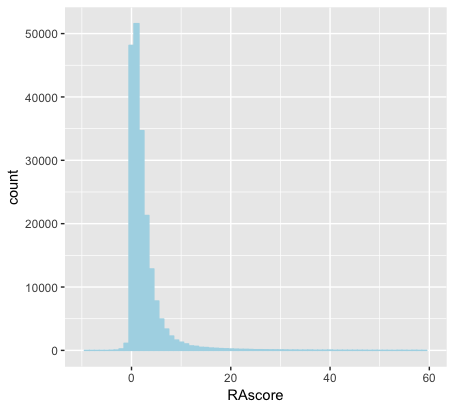
Then ,due to the departure from normality, we appeal to non-parametric method, powerful XGBoost to find out what are the important features towards score and time. And we divided them into groups according to their importance.

First, we extract some features from the raw dataset.

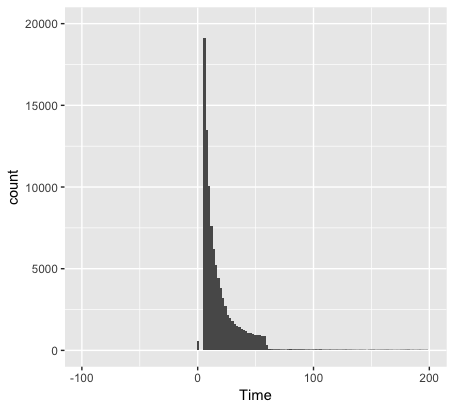
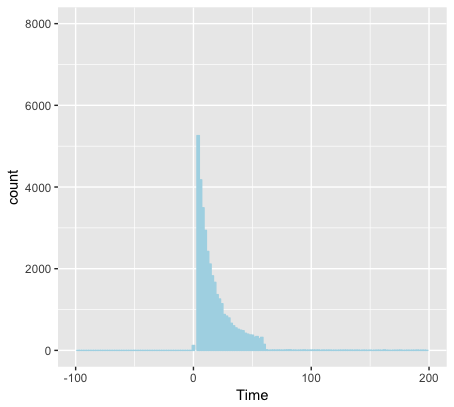
Then, we train the model on these features. After tuning parameter, we got a cross-validated MSE for the best model of R Questions as nearly 8.2 and MSE for python questions as 19.1. From the importance matrix, we can see that, for R questions, sum\_tag\_freq, max\_tag\_freq, code\_length are most relevant to the question score. And there is a gap of information gains between features in cluster 1 and features in cluster 2. As for Python questions, the most important factors change. Title length is inserted into the front row.

**IV. Empirical Results**

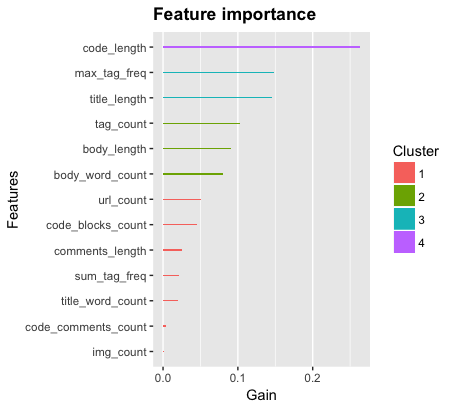
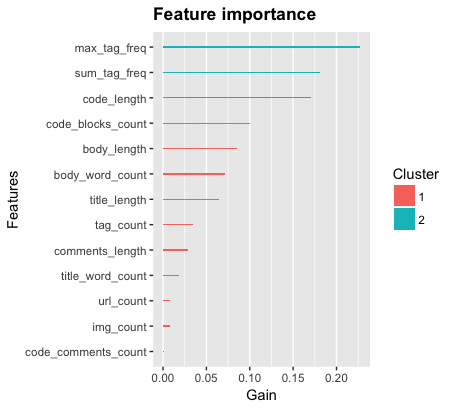
**R and python answers summary:**



**Time EDA:**



**Gradient Boosting Trees: XGBoost :**



**Logistic Regression**   
  
Label prediction: Split data into train(80%) and test(20%), do logistic regression, calculate prediction accuracy.   
  
R Questions: Prediction Accuracy = 0.7927178;  
  
Python Questions: Prediction Accuracy = 0.7923346

**V. Conclusion**

Above all, we are confident that whether a question is solved or not depends on the question’s score, and as we built a model to predict question scores, we can infer that some text features are relevant to the question’s outcome. Our suggestion is to try to tag your questions with some popular and regularized tags and control your codes’ length and title’s length short included in your question. Popular tags and brief description will make you a stack Overflow star and you don’t need to worry about your programming homework anymore.

Reference

Package ‘xgboost’—CRAN