B.Sc. Dissertation

Evaluating the Programming Expertise of Users on StackOverflow

By
Zhong Mengjia
A0148016L

Department of Information Systems and Analytics

School of Computing

National University of Singapore

2018/2019

B.Sc. Dissertation

Evaluating the Programming Expertise of Users on StackOverflow

By

Zhong Mengjia

A0148016L

Department of Information Systems and Analytics

School of Computing

National University of Singapore

2018/2019

Project No: H175330

Advisor: Assoc Prof Hahn Jungpil

Deliverables:

Report: 1 Volume

1. Introduction

Online Q&A platforms have become important sources for knowledge sharing and knowledge creation. StackOverflow, which is one of the most popular Q&A websites among software developers, has more than two million users (Ponzanelli et al., 2014). StackOverflow is steadily growing both in the size of its community and in its influence. Not only is the knowledge exchanged among developers on the platform valuable, but the StackOverflow profiles of many users also attract a lot of attention from employers. A user's reputation, a metric calculated by StackOverflow based on user-interaction on the platform, is prominently displayed on his/her profile page and is frequently quoted in developers' résumés. Many employers take it as an indicator that reflects site familiarity, programming expertise, and peer reputation of users (Morrison & Murphy-Hill, 2013) and use it in their recruitment decision making.

It is widely accepted that reputation may be a useful metric that can encourage active participation and maintain user-loyalty. Both active participation and user-loyalty are critical success factors for a Q&A website. However, to what extent reputation can reflect the level of coding capabilities of users on StackOverflow is still in debate.

2. Motivation and Objectives

Although reputation is engineered by StackOverflow and functions well in terms of inducing user engagement, some other usages of the reputation indicator were unexpected. Reputation, as a numerical indicator that can be easily accessed, has been widely used as an indicator of developers' programming knowledge and capability in many studies conducted to better understand the Q&A platform (Morrison & Murphy-Hill, 2013). Most of the research projects were conducted based on the assumption that reputation can accurately reflect users' programming knowledge and capabilities. If this assumption is not substantiated, the conclusions derived from these studies will be tenuous.

The assumption is widely adopted among not only researchers but also many employers. With an increasing number of people including their reputation scores as an indicator of their coding skills in their résumés and LinkedIn profiles, the reputation score has become a tool that helps developers market themselves as a tech-savvy person in the job market. Moreover,

StackOverflow itself also operates as a platform aimed at connecting developers with companies. More than 20,000 companies utilized StackOverflow to hunt for talent. Different from other platforms such as LinkedIn, the employers on StackOverflow get access to not only the developers' Curriculum Vitae but also their StackOverflow profiles.

There are many posts like *"How to source developers from StackOverflow"* elaborate on the common procedures that recruiters take to source talents from StackOverflow. Tome Winter, who is an experienced hiring manager in the IT industry contends that, to take advantage of StackOverflow, the recommended hiring procedure for employers is to quickly select candidates based on the tags and the reputation shown in their profiles (Winter, 2017). Then, employers can approach and invite the candidates for a job interview or ask them to take a coding test.

Many users on StackOverflow have also captured this emerging hiring process. Therefore, many "tips" regarding how to quickly improve one's reputation score on StackOverflow, in order to leave a good impression on viewers of their profiles, have been widely discussed. The most well-known one is the "6 Simple Tips to Get StackOverflow Reputation Fast",[1] which implies that not all users with high reputation scores on StackOverflow accumulated reputation points by contributing answers mindfully.

Since referring to reputation scores is only the first step taken by hiring managers to evaluate candidates, it seems that the consequences will not be very severe even if the technical knowledge and capabilities of developers reflected by their reputation scores on StackOverflow are not very accurate. However, companies that sign up to use StackOverflow Talent, StackOverflow's hiring services, must pay premium fees. If StackOverflow cannot justify that the service that they provide is indeed premium not only in terms of helping recruiters to reach out more developers but also in terms of accurately presenting competencies of developers which is critical for recruiters to find the exact match to their vacant positions, companies may not be willing to pay for the job services provided by StackOverflow Talent in the long run. One advantage that StackOverflow has to differentiate itself from other job-posting platforms is that it can help employers identify "passive developers" who are not actively looking for jobs but may be attractive to hiring managers,

[1] https://meta.stackexchange.com/questions/17204/six-simple-tips-to-get-stack-overflow-reputation-fast

which definitely require recruiters to put more efforts into the hiring process (Gray, 2014). When hiring professionals spend more time and efforts to attract "passive developers", their expectations on those developers also increase. If they find that the candidates who they have spent a great amount of time to search, contact, and invite for interview, may not be as good as what their StackOverflow profiles suggest, StackOverflow Talent services may face a crisis of confidence. Therefore, it is essential to ensure that the profiles of developers on StackOverflow can reflect the coding capabilities of developers to a great extent.

Therefore, this research project aims at carefully evaluating to what extent the reputation score may be able to reflect the coding capabilities of users on the platform and explore more dimensions that have not been included in the current mechanism yet.

3. Literature Review

In this section, I review the relevant background and literature related to this study. I will first discuss some loopholes of the existing reputation scoring mechanism on StackOverflow that have been identified by other researchers. Then I will illustrate existing numerical metrics proposed to emphasize other aspects of users' activities on Q&A websites. Finally, since it is necessary to seek the ground truth of the users' coding expertise from GitHub, popular methodologies of cross-platform analysis that have recently been used by researchers will be carefully reviewed.

3.1 Reputation and Activeness

The formula[2] used to calculate users' reputation on StackOverflow suggests that the policies of points awarding and deduction are asymmetric. If a user's answer is voted up, the user will receive 10 points, whereas if it is voted down, only 2 points will be deducted from his total reputation score. Moreover, if a user votes down another user's answer, one point will be deducted from his total reputation score, which may discourage some users from raising objections to answers with poor quality. In addition to votes, contributing an accepted answer is another major way of accumulating reputation points. If a user's answer is accepted by the

[2] See https://meta.stackexchange.com/questions/7237/how-does-reputation-work

question asker, the user will be awarded 15 points. However, it has been shown that the faster one responds to a question, the more likely his answer will be accepted (Anderson et al., 2012). Moreover, users can also accumulate points by asking questions and accepting others' answers. Consequently, to quickly accumulate reputation, active and prompt participation seems more important than contributing high-quality answers.

Bosu et al. (2013) suggest that there are 4 simple tricks that can help users gain reputation on StackOverflow relatively quickly: 1) answering questions related to tags with lower expertise density, 2) answering questions promptly, being the first one to answer a question, 3) being active during off-peak hours, and 4) contributing to diverse areas. It is true that if users follow such guidelines to gain reputation scores, this will facilitate the development and growth of the community, since it is essentially activeness and first-mover agility that is rewarded on the platform. Although they can be praised as "good citizens" in the community, they may not fit employers' desired image of "good programmers."

3.2 Potential Factors Contributing to Expertise Evaluation

Several researchers conclud that reputation may not be a good measurement in terms of knowledge contribution in their studies. Yang et al. (2014) investigated the different behavior patterns of two groups of users – Sparrows and Owls – on StackOverflow. Sparrows refer to the users who are very active in the community and contribute to a majority of the produced content, while owls refer to users who provide useful answers but only infrequently so. Since the correlation between reputation score and user activeness for the topic of C# was quite high (r=0.68), the author argued that the reputation score is not a reliable indicator reflecting programmers' knowledge and capabilities (Yang et al., 2014). They proposed an alternative metric, called MEC (Mean Expertise Contribution), by taking into consideration two other dimensions– the *debatableness* of a question[3] and *utility of an answer*.[4]

[3] Debatableness of question was measured based on the number of answers it receives

[4] Utility of answers was measured by its relative rank in the list of answers

$$\mathtt{MEC}_{u,t} = \frac{1}{|Q_t^u|} \sum_{\forall q_i \in Q_{u,t}} \mathcal{AU}(u, q_i) * \frac{\mathcal{D}(q_i)}{\mathcal{D}_t^{avg}}$$

where:

- $\mathcal{AU}(u, q_i)$ is the **utility** of the answer provided by user u to question q_i; in our study, $\mathcal{AU}(u, q_i) = \frac{1}{Rank(a_{q_i})}$, that is the inverse of the rank of the answer provided by u for question q. The larger $\mathcal{AU}$, the higher the expertise level shown by the user in question q_i;
- $\mathcal{D}$ is the **debatableness** of the question q_i, calculated as the number of answers $|A_{q_i,t}|$ provided for question q_i;
- $\mathcal{D}_t^{avg}$ is the **average debatableness** of all the questions related to the topic t, calculated as $\frac{1}{|Q_t|} * \sum_{\forall q_j \in Q_t} |A_{q_j,t}|$.

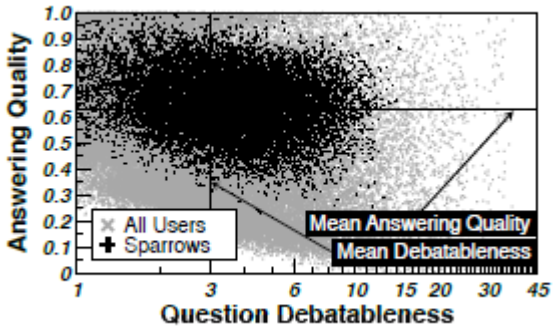


Figure 1: Distribution of users according to the avg. debatableness of questions they answer, and the avg. answer quality

The researchers observed that most users answered questions that are less debated, while only about 10% of users contributed to disputed questions. Most sparrows[5] focused on the less debated questions, which shows that activeness may not be a good measure of expertise. Moreover, Yang et al. (2014) observed that good developers identified by high MEC answer and post more difficult and popular questions.[6] However, there was no rigorous validation that MEC is a better metric that reflects the coding capabilities of users on StackOverflow more accurately compared to the reputation mechanism.

In addition to the two dimensions mentioned above, Hart and Sarma (2014) highlight that the quality of users' answers determines public perception of their expertise and explains what makes a public-accepted answer on Stack Overflow. For example, *answer length* is important

[5] Sparrows: users with |Au,t| >=10

[6] Popularity of questions was measured by number of views, and difficulty was measured by time to solution.

because longer answers tend to be more thorough. In addition, the *presence of code snippet and URL* are also essential elements of high-quality answers (Hart & Sarma, 2014).

More recently, some researchers started to apply social network analysis to Q&A websites. PageRank which was designed to measure the importance of a webpage based on links has been used to identify influential users on StackOverflow. Movshovitz et al. (2013) expected that, compared to reputation scores, PageRank might be more indicative of the quality of answers contributed by users on StackOverflow. To validate their hypothesis, they have constructed three graph models. Nodes in the three graph models are the representation of users while edges in the three graph model represent three types of interaction (i.e., answers to questions, accepted answers to questions, and upvoted answers to questions) respectively Then, the researchers plotted the correlation distribution between PageRank calculated in the three different graph models with reputation scores. After finding the similarity between the three distribution plots, they concluded that PageRank is more directed correlated with volume instead of quality (Movshovitz, 2013). However, the descriptive analysis may not be robust enough to prove that PageRank is more directly affected by the activeness of users. More importantly, the researchers did not take time discount factor into consideration. Many interactions among users represented by edges may happen a long time before the time when the graph is modelled.

3.3 StackOverflow and GitHub

GitHub is a platform mostly used for software development. Since a developer's coding activities and outputs are observable on the platform, researchers started measuring developer quality using trace data on GitHub. For instance, Wu et al. (2014) proposed to measure developer quality by calculating the proportion of non-bug-introduce commits out of the total commits contributed by the developers on GitHub.

After obtaining the ground truth of coding capabilities of developers by analyzing their activities on GitHub, then it is critical to link user identities across platforms. There have been attempts to match full names or email addresses shared by different aliases by inferring email prefixes based on combinations of name parts (Bird et al., 2006). One of the most common ways to link accounts of StackOverflow with GitHub is to match the email address across platforms which has been transformed into MD5 email hash (Vasilescu et al., 2013).

4. Overall Research Approach

The research includes three main steps. Firstly, data from StackOverflow and GitHub was collected and matched to identify how many users are active on both platforms. Secondly, a set of variables regarding users' behaviour on StackOverflow will be engineered. All features will be divided into reputation-linked features and other dimensions for future analysis. The users' behavior on GitHub will also be captured, and users will be grouped and labelled with the level of their performance on GitHub. The level of performance on GitHub will be used as the ground truth of users' coding capabilities in the next stage. Lastly, with the group labels of all users in the pool, machine learning models including Multinomial Logistic Regression, K-nearest neighbors, and Decision Trees, Neural Networks, and some ensemble learning models will be used to predict the coding expertise of users on StackOverflow. The accuracy of models and the implication of the summary of features will be discussed.

5. Methodology

5.1 Data Collection

A complete dataset containing all the actions on StackOverflow from its inception until September 5th, 2018 has been obtained from the Stack Exchange Archive site.[7] The data regarding user behavior on StackOverflow is stored in separate XML files including badges, comments, posts, users and so on. Most features needed can be obtained directly or engineered based on the XML files. The data contains 9.6 million users, 17 million questions, and 26 million answers. There are on average 11 million visits to the StackOverflow website and approximately 7000 questions are posted on a daily basis. 71% of all questions were answered.

The GitHub data was gathered from GHTorrent (Gousios & Spinellis, 2012), a service that records event streams and data from GitHub and provides that data back to the community in the form of incremental MongoDB data dumps. The GitHub dataset contains information of 9M users and their coding activities such as the number of commits, the number of lines in each commit, projects that they worked on and so.

[7] https://archive.org/details/stackexchange

In total, about 89,000 users can be accurately matched across StackOverflow and GitHub by the email address. Among the 89,000 users, 83,000 of them are active on both platforms.

5.2 Independent Variables Engineering

Firstly, all features explicitly stated in the rules of calculating reputation including the **number of both upvotes and downvotes, number of questions/answers, number of the accepted answer,** and **number of badges** will be extracted**.** In addition, the other features that may closely related to users' coding capabilities but have not been included in calculating reputation scores yet will also be explored.

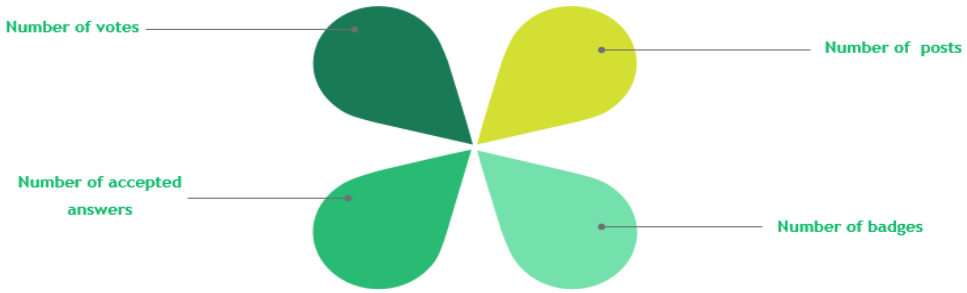


Figure 2.1 Reputation-Linked Features

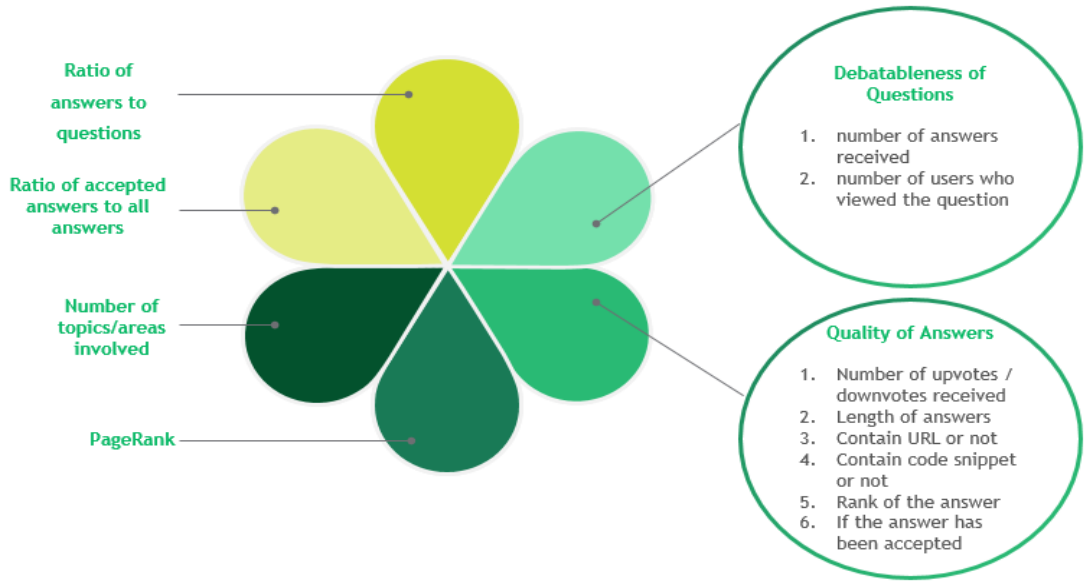


Figure 2.2 New Features with Potential

PageRank:

To analyze user interactions by the underlying graph structure of StackOverflow and to calculate **PageRank**, the StackOverflow was modelled as a network. In this case, users and interactions are represented by nodes and edges, respectively. For example, if user A answered question brought up by user B, there will be a directed edge from user B to user A. To use PageRank to identify influential users, it is important to take time factors into consideration. For instance, it is possible user A answered user B's question when user B was a novice who asked naïve questions. However, at the point in time when the graph model is built, user B may have evolved into a professional user who has helped many other users solve problems. Therefore, user B must be well connected to many other users in the modelled graph, which enables him to obtain a higher score when running PageRank. In this case, user A will also obtain a high score even if he is also very inactive in recent years and built up only one connection to user B many years ago. It is because PageRank assumes that user A who can help user B solve problems must be a developer with good expertise but ignores the time lag. It may be fallacious if user A only helped to answer a simple question raised by user B when he was a novice. Therefore, all edges will be weighted by a time discount factor which is represented by the reciprocal of time. For example, if the interaction between A and B happened 5 years ago, the time discount factor of the edge linked them will be 0.2. If the interaction happened within 1 year, the weight of the edge will be 1. In this case, the scores awarded to user A will be constrained by multiplying time factor to the scores of user B.

Debatebleness of Questions:

Both the **number of answers received** and **the number of users who viewed the question** explicitly suggest the debatebleness and popularity of questions (Yang et al. 2014).

Quality of Answers:

The utility of answers is an important dimension to measure the knowledge contribution of users. The **number of upvotes and downvotes received, length of answers, contain URL snippet or not** (Hart & Sarma, 2014), **the rank of the answer** (Yang et al. 2014), and **if the answer is accepted** are all variables have been used to assess the quality and utility of answers.

Overall Performance:

The Programmers who continue to provide answers in high quality worth more attention. Therefore, the overall performance including **the ratio of answers to questions**, **the ratio of accepted answers to total answers**, and the **number of areas** which the users were involved in will also be considered.

Within the group of variables under the quality of answers and the debatebleness of questions, principal component analysis (PCA) with one component will be used to reduce the dimensionality of the set of variables.

$$\boldsymbol{Question\ PCA} = \boldsymbol{\beta_1 number\ of\ answers\ received} + \boldsymbol{\beta_2\ number\ of\ users\ who\ viewed\ the\ question}$$

$$\boldsymbol{Answer\ PCA} = \boldsymbol{\beta_1 number\ of\ upvotes\ received} + \boldsymbol{\beta_2\ number\ of\ downvotes\ received} + \boldsymbol{\beta_3\ length\ of\ answers} + \boldsymbol{\beta_3\ if\ the\ answer\ contain\ url} + \boldsymbol{\beta_3\ if\ the\ answers\ contain\ code\ snippet} + \boldsymbol{rank\ of\ the\ answer} + \boldsymbol{\beta_3\ if\ the\ answer\ is\ accepted}$$

It is possible β may be zero, which suggests that the corresponding variable is eliminated from the group.

For every user on StackOverflow, they will have a question PCA score and an answer PCA score which are the average PCA values across all questions they have answered and answers they have posted.

5.3 Dependent Variable Engineering

Based on users' activities on GitHub, we collected the number of commits, the number of projects, the number of accepted pull request, the number of followers, and the number of watchers of each individual. To categorize all users into different groups and to rank the group, K-means has been used to stepwise divide the whole group of people into different

sub-groups that ranked by the coding performance of all developers on GitHub. Firstly, all users will be divided into two groups, the group with outstanding users will be labelled as top performers, and the left users will form another group which will be further categorized into another two groups. Similarly, the group containing users with better coding performance will be labelled as the second top group of performers, while the left group will be further divided.

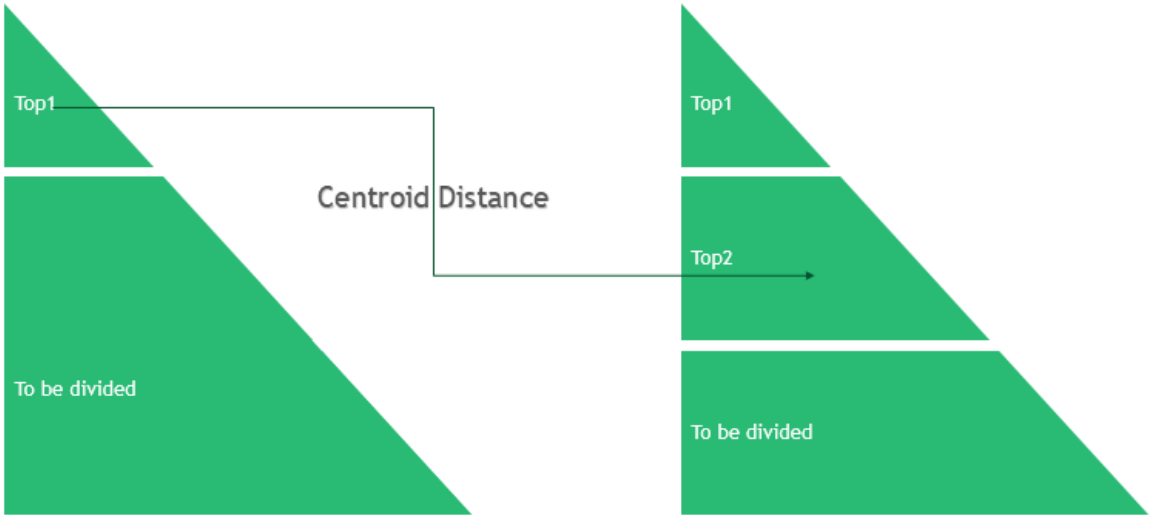


Figure 3.1 Centroid Distance between Two Adjacent Groups

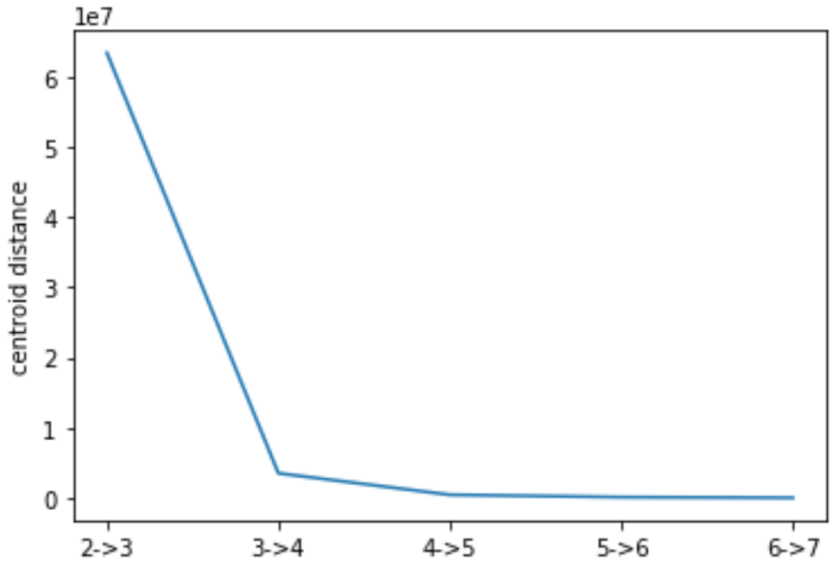


Figure 3.2 Centroid Distance between adjacent groups

The line chart (figure 3.2) shows the distance between two centroids of the two adjacent groups. For example, the first value is the sum of square difference distance between the top group and the second top group (as shown in figure 3.1). After all users being divided into four groups, the centroid distance between those groups that were further divided drop insignificantly, which suggests the similarity between two adjacent groups.

To further understand the properties of users belonging to different groups, descriptive analysis has been done to evaluate how different the adjacent groups are and if four is the appropriate number of clusters.

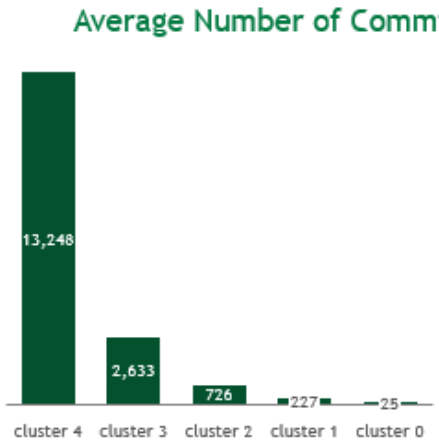


Figure 4.1

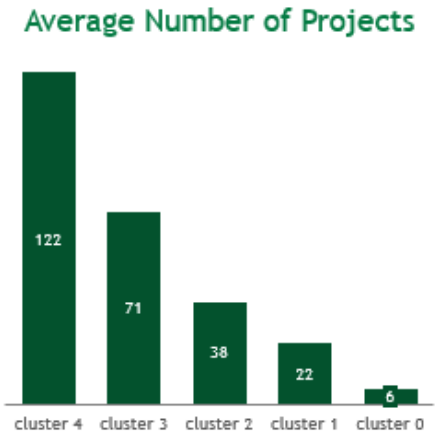


Figure 4.2

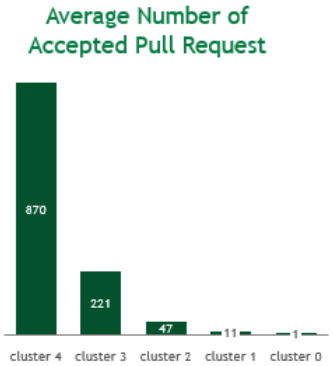


Figure 4.3

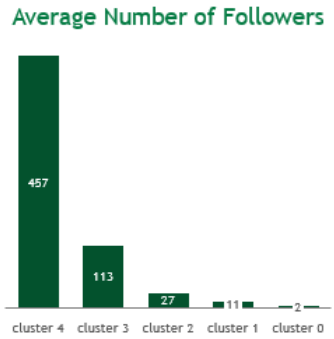


Figure 4.4

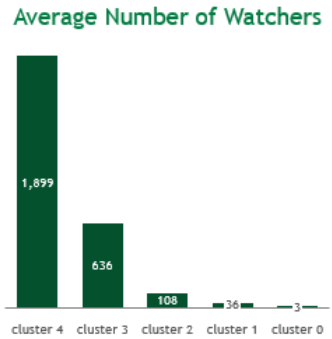


Figure 4.5

The five histograms (*figures 4.1 to 4.5*) display the average of five features used as categorizing criterion. It is apparent that all average values decreased gradually, proving the rationality of grouping and labeling mechanism being used. However, the difference among

group 0 and group 1 which contain users with the poor contribution to GitHub is not as evident as the difference between other adjacent groups. Taking the findings of *figure 3.2* into consideration, all machine learning models used to investigate users' behaviors on StackOverflow and any further analysis regarding users' coding expertise will be based on categorizing users into four groups at different expertise level. The group label starting from 3 to 0 will be attached to each user indicating which level of the expertise group they belong to. The larger the label is, the better the coding capabilities of the group of users should be.

5.4 Logistic Regression Analysis

5.4.1 Preliminary analysis

5.4.1.1 Correlation Matrix

	expertise	num_comm	num_proje	num_acc_p	num_follow	num_watch	reputation	view	upvotes	downvotes	num_posts	num_qns	num_ans	ans_qns_ra	num_badge	num_accep	accept_rati	qns_pca	ans_pca	pagerank	num_areas
expertise	1.00000	0.59897	0.52705	0.39361	0.25015	0.24813	0.09939	0.06453	0.10184	0.01727	0.10347	0.07195	0.09544	0.06199	0.14317	0.09426	0.27269	-0.01097	0.35981	0.09239	0.16941
num_commit	0.59897	1.00000	0.36664	0.41061	0.26401	0.20019	0.06201	0.04706	0.05488	0.00821	0.06689	0.03922	0.06322	0.04719	0.08322	0.06902	0.15722	-0.00873	0.19890	0.06752	0.09216
num_projects	0.52705	0.36664	1.00000	0.22805	0.21971	0.19964	0.07589	0.04643	0.09049	0.01288	0.07498	0.07396	0.06455	0.02439	0.12191	0.05769	0.15965	-0.00150	0.22197	0.05253	0.14405
num_acc_pr	0.39361	0.41061	0.22805	1.00000	0.30079	0.18224	0.05109	0.04243	0.03308	0.01198	0.05527	0.01751	0.05544	0.05153	0.06216	0.06196	0.10972	-0.00262	0.12748	0.05969	0.06925
num_followe	0.25015	0.26401	0.21971	0.30079	1.00000	0.64571	0.05455	0.07245	0.02049	0.00514	0.03775	0.00936	0.03844	0.03230	0.05784	0.04238	0.09210	0.00491	0.07752	0.03122	0.04675
num_watche	0.24813	0.20019	0.19964	0.18224	0.64571	1.00000	0.05252	0.04509	0.02578	0.00511	0.04141	0.01809	0.04052	0.03360	0.06537	0.04220	0.08388	0.00172	0.07208	0.03155	0.05710
reputation	0.09939	0.06201	0.07589	0.05109	0.05455	0.05252	1.00000	0.69007	0.56766	0.35027	0.83485	0.27231	0.83628	0.35084	0.82676	0.80075	0.17836	0.07268	0.11672	0.62921	0.53195
view	0.06453	0.04706	0.04643	0.04243	0.07245	0.04509	0.69007	1.00000	0.45509	0.51374	0.56231	0.20530	0.55797	0.18517	0.61206	0.52021	0.10930	0.05735	0.07343	0.41123	0.33646
upvotes	0.10184	0.05488	0.09049	0.03308	0.02049	0.02578	0.56766	0.45509	1.00000	0.31162	0.57997	0.34828	0.54583	0.18164	0.65431	0.49553	0.20720	0.05219	0.15981	0.42013	0.55165
downvotes	0.01727	0.00821	0.01288	0.01198	0.00514	0.00511	0.35027	0.51374	0.31162	1.00000	0.33622	0.08575	0.34114	0.15286	0.29380	0.33681	0.05550	0.02984	0.03275	0.29072	0.18985
num_posts	0.10347	0.06689	0.07498	0.05527	0.03775	0.04141	0.83485	0.56231	0.57997	0.33622	1.00000	0.42146	0.98116	0.48926	0.78310	0.94172	0.22280	0.03422	0.15023	0.87099	0.65352
num_qns	0.07195	0.03922	0.07396	0.01751	0.00936	0.01809	0.27231	0.20530	0.34828	0.08575	0.42146	1.00000	0.23872	-0.01986	0.65949	0.19032	0.19764	0.05021	0.14281	0.15876	0.54301
num_ans	0.09544	0.06322	0.06455	0.05544	0.03844	0.04052	0.83628	0.55797	0.54583	0.34114	0.98116	0.23872	1.00000	0.52873	0.69799	0.96812	0.19641	0.02595	0.13047	0.89883	0.58397
ans_qns_rati	0.06199	0.04719	0.02439	0.05153	0.03230	0.03360	0.35084	0.18517	0.18164	0.15286	0.48926	-0.01986	0.52873	1.00000	0.21919	0.57584	0.11720	-0.01065	0.07597	0.59889	0.23731
num_badges	0.14317	0.08322	0.12191	0.06216	0.05784	0.06537	0.82676	0.61206	0.65431	0.29380	0.78310	0.65949	0.69799	0.21919	1.00000	0.64018	0.30395	0.10240	0.21962	0.49803	0.74204
num_accept	0.09426	0.06902	0.05769	0.06196	0.04238	0.04220	0.80075	0.52021	0.49553	0.33681	0.94172	0.19032	0.96812	0.57584	0.64018	1.00000	0.20075	0.01945	0.10814	0.91648	0.50649
accept_ratio	0.27269	0.15722	0.15965	0.10972	0.09210	0.08388	0.17836	0.10930	0.20720	0.05550	0.22280	0.19764	0.19641	0.11720	0.30395	0.20075	1.00000	0.01278	0.42318	0.17716	0.41354
qns_pca	-0.01097	-0.00873	-0.00150	-0.00262	0.00491	0.00172	0.07268	0.05735	0.05219	0.02984	0.03422	0.05021	0.02595	-0.01065	0.10240	0.01945	0.01278	1.00000	-0.02800	0.01221	0.07221
ans_pca	0.35981	0.19890	0.22197	0.12748	0.07752	0.07208	0.11672	0.07343	0.15981	0.03275	0.15023	0.14281	0.13047	0.07597	0.21962	0.10814	0.42318	-0.02800	1.00000	0.11187	0.33772
pagerank	0.09239	0.06752	0.05253	0.05969	0.03122	0.03155	0.62921	0.41123	0.42013	0.29072	0.87099	0.15876	0.89883	0.59889	0.49803	0.91648	0.17716	0.01221	0.11187	1.00000	0.47079
num_areas	0.16941	0.09216	0.14405	0.06925	0.04675	0.05710	0.53195	0.33646	0.55165	0.18985	0.65352	0.54301	0.58397	0.23731	0.74204	0.50649	0.41354	0.07221	0.33772	0.47079	1.00000

Figure 5. Correlation Matrix

The correlation matrix shown here confirmed the conclusion drawn by many researchers that reputation is highly correlated with the number of posts including both answers and questions. However, it is evident that there is little correlation between reputation with the quality of answers and the accepted ratio which are all correlated with the coding expertise of users.

5.4.1.2 StackOverflow Features Summary

Figure 6.1	*Figure 6.2*

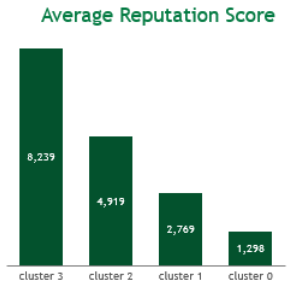
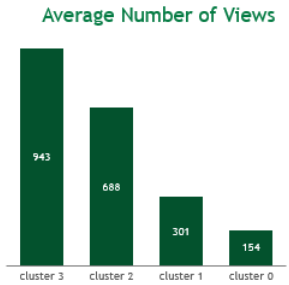

	

Figure 6.3	*Figure 6.4*

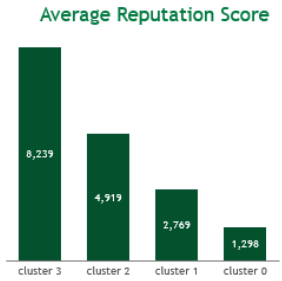
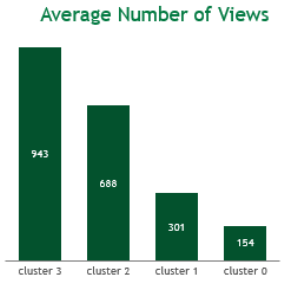

	

Figure 6.5	*Figure 6.6*

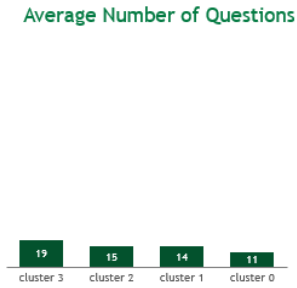
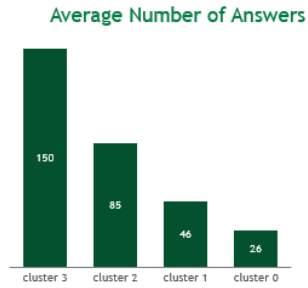

	

Figure 6.7	*Figure 6.8*

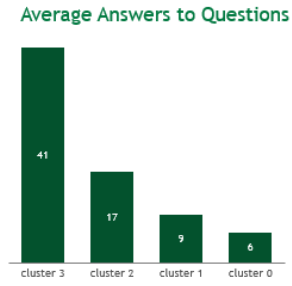
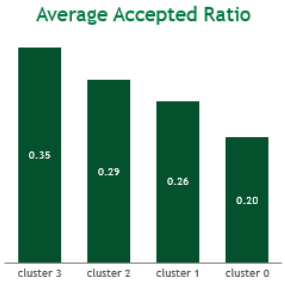

	

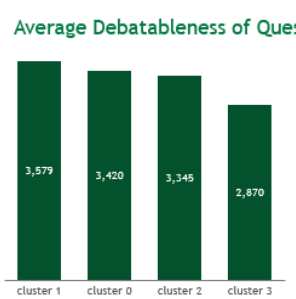
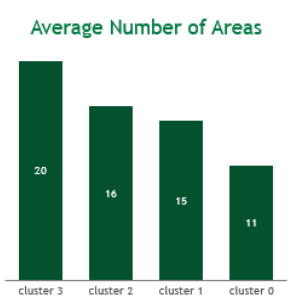
Figure 6.9	*Figure 6.10*
	

Figure 6: Summary of Features on StackOverFlow by Cluster

According to the histograms (*figures 6*), the average level of knowledge contribution made by each group on StackOverflow is consistent with their contributions on GitHub. However, the average number is just a rough indicator of the correlation between users' performance across platforms. To further evaluate the relationship of users' activities on StackOverflow with the coding expertise, linear regression, multinomial logistics regression, and some other deep learning algorithms will be constructed.

5.4.2 Linear Regression Analysis

To assess the relationship between independent variables with different dimensions of coding capabilities, six linear regression models were built. The five dependent variables including the number of commits, the number of projects, the number of the accepted pull request, the number of followers, and the number of watchers are the features used to group and rank users. The sixth dependent variable -- "coding expertise" is the label of the group that users belong to.

5.4.2.1 OLS Reputation-Linked Model

To have a better understanding of to what extent reputation and reputation-linked features can reflect the coding capabilities of users, the linear regression model was built only based on the features that are explicitly stated in the reputation calculation rules.

OLS Regression Results						
	Number of Commits			Number of Projects		
	coefficients	std error	p-value	coefficients	std error	p-value
reputation	-0.2312	0.011	0.0000	-0.2873	0.009	0.0000
number of views	-0.0064	0.011	0.5780	-0.0793	0.009	0.0000
number of upvotes	-0.0089	0.003	0.0110	-0.0039	0.003	0.1720
number of downvotes	-0.0323	0.008	0.0000	-0.0091	0.006	0.1500
number of questions	-0.0954	0.006	0.0000	-0.122	0.005	0.0000
number of answers	-0.0712	0.014	0.0000	0.0251	0.011	0.0270
number of badges	0.2712	0.008	0.0000	0.3675	0.007	0.0000
accepted answers	0.1426	0.016	0.0000	-0.0059	0.013	0.6520

Adj. R-squared	0.043			0.107		
	Number of the Accepted Pull Requests			**Number of Followers**		
	coefficients	std error	p-value	coefficients	std error	p-value
reputation	-0.2006	0.013	0.0000	-0.0922	0.009	0.0000
number of views	0.0152	0.013	0.2470	0.0989	0.009	0.0000
number of upvotes	-0.0187	0.004	0.0000	-0.017	0.003	0.0000
number of downvotes	-0.0235	0.009	0.0070	-0.0435	0.006	0.0000
number of questions	-0.0948	0.007	0.0000	-0.0555	0.005	0.0000
number of answers	-0.0754	0.016	0.0000	-0.0612	0.011	0.0000
number of badges	0.2296	0.01	0.0000	0.1235	0.006	0.0000
accepted answers	0.1551	0.018	0.0000	0.0801	0.012	0.0000
Adj. R-squared	0.022			0.017		
	Number of Watchers			**Coding Expertise**		
	coefficients	std error	p-value	coefficients	std error	p-value
reputation	-0.0847	0.008	0.0000	-0.1721	0.005	0.0000
number of views	0.0182	0.009	0.0380	-0.0349	0.005	0.0000
number of upvotes	-0.0163	0.003	0.0000	-0.0041	0.002	0.0110
number of downvotes	-0.0206	0.006	0.0000	-0.0118	0.004	0.0010
number of questions	-0.0593	0.005	0.0000	-0.0767	0.003	0.0000
number of answers	-0.0407	0.01	0.0000	-0.0089	0.006	0.1610
number of badges	0.1331	0.006	0.0000	0.2145	0.004	0.0000
accepted answers	0.0553	0.012	0.0000	0.04	0.007	0.0000
Adj. R-squared	0.015			0.114		

Table xx Summary of Pure Reputation Linear Regression Model

In most cases, the number of badges that users won on the StackOverflow and the number of accepted answers are significant predictors and positively correlated with users' performance on GitHub. To get large the number of badges and high volume of accepted answers, users need to ensure the quality of their works. The other features including the number of answers, the number of questions, and the number of votes which directly reflect the activeness of users on StackOverflow are either negatively correlated with users' performance on GitHub or are insignificant predictors.

5.4.2.2 OLS Full Model

With all independent variables in the linear regression models, the performance of the six models is listed below. The table shows the sign of the coefficient of all independent variables and whether they are significant. Please refer to *Appendix 9.1* for more details.

OLS Regression Results						
	Number of Commits			**Number of Projects**		
	coefficients	std error	p-value	coefficients	std error	p-value
reputation	0.1514	0.03	0.0000	0.0373	0.026	0.1460
number of views	0.1736	0.032	0.0000	0.1046	0.028	0.0000
number of upvotes	0.0085	0.011	0.4250	-0.0108	0.009	0.2380
number of downvotes	-0.0847	0.045	0.0620	-0.0595	0.039	0.1250
number of questions	-0.0526	0.013	0.0000	-0.0792	0.011	0.0000
number of answers	-0.6065	0.055	0.0000	-0.4806	0.047	0.0000
number of badges	0.0949	0.022	0.0000	0.1263	0.019	0.0000
accepted answers	0.494	0.051	0.0000	0.2763	0.044	0.0000
accepted ratio	-0.024	0.005	0.0000	-0.0188	0.004	0.0000
question PCA	0.0882	0.008	0.0000	0.1535	0.007	0.0000
answer PCA	0.3519	0.005	0.0000	0.2754	0.004	0.0000
PageRank	0.1099	0.029	0.0000	0.0484	0.025	0.0550
number of areas	-0.0274	0.01	0.0070	0.0658	0.009	0.0000

answer to question ratio	0.038	0.009	0.0000	0.0224	0.008	0.0040
Adj. R-squared:	0.309			0.346		
	Number of the Accepted Pull Requests			**Number of Followers**		
	coefficients	std error	p-value	coefficients	std error	p-value
reputation	0.0271	0.011	0.0150	0.0192	0.008	0.0230
number of views	0.0731	0.012	0.0000	0.1856	0.009	0.0000
number of upvotes	-0.0083	0.004	0.0370	-0.0154	0.003	0.0000
number of downvotes	0.019	0.017	0.2590	-0.0285	0.013	0.0260
number of questions	-0.0216	0.005	0.0000	-0.0369	0.004	0.0000
number of answers	-0.167	0.021	0.0000	-0.1424	0.016	0.0000
number of badges	0.0313	0.008	0.0000	0.0336	0.006	0.0000
accepted answers	0.136	0.019	0.0000	0.1008	0.014	0.0000
accepted ratio	0.0013	0.002	0.4390	0.0009	0.001	0.4720
question PCA	0.0078	0.003	0.0090	0.0172	0.002	0.0000
answer PCA	0.0591	0.002	0.0000	0.0523	0.001	0.0000
PageRank	0.0488	0.011	0.0000	-0.0036	0.008	0.6680
number of areas	-0.0105	0.004	0.0050	-0.0058	0.003	0.0440
answer to question ratio	0.012	0.003	0.0000	0.011	0.003	0.0000
Adj. R-squared:	0.097			0.145		
	Number of Watchers			**Coding Expertise**		
	coefficients	std error	p-value	coefficients	std error	p-value
reputation	0.0522	0.009	0.0000	0.0026	0.001	0.0000
number of views	0.1136	0.009	0.0000	0.0038	0.001	0.0000
number of upvotes	-0.0181	0.003	0.0000	0.0005	0	0.0080
number of downvotes	-0.0454	0.013	0.0000	-0.0005	0.001	0.5320
number of	-0.0229	0.004	0.0000	-0.0003	0	0.2880

questions						
number of answers	-0.073	0.016	0.0000	-0.0089	0.001	0.0000
number of badges	0.0287	0.006	0.0000	0.0008	0	0.0590
accepted answers	0.0837	0.015	0.0000	0.0094	0.001	0.0000
accepted ratio	-0.0017	0.001	0.1970	-0.0007	8.73E-05	0.0000
question PCA	0.0048	0.002	0.0390	0.0003	0	0.0380
answer PCA	0.0328	0.001	0.0000	0.0063	8.99E-05	0.0000
PageRank	-0.003	0.008	0.7220	0.0008	0.001	0.1340
number of areas	-0.0086	0.003	0.0030	-0.0012	0	0.0000
answer to question ratio	0.0021	0.003	0.4300	0.0003	0	0.1440
Adj. R-squared:	0.07			0.233		

Table xx Summary of Full Linear Regression Model

In most cases, reputation and number of posts including both questions and answers have negative coefficients and appear to be insignificant. The number of commits and the number of projects are two explicit indicators of users' activeness on GitHub. Therefore, it cannot be concluded that the active participation on Q&A platforms ensures the similarly active participation on other development platforms.

5.4.3 Multinomial Logistic Regression Model

In addition to taking coding abilities as continuous variables and build up linear regression to predict the numerical value that measures the expertise of developers, the coding expertise of users can be modelled as an ordinal variable ranging from 0 to 3. The larger the coding expertise is, the group with better expertise the users should belong to.

5.4.3.1 Reputation-Linked MNLogit Model

The multinomial logistic regression model with only reputation-linked feature has been trained as a benchmark.

MNLogit Regression Results			
y=1	coefficient	std error	P>|z|

reputation	0.0707	0.025	0.0050
number of views	0.2172	0.056	0.0000
number of upvotes	0.0384	0.006	0.0000
number of downvotes	-0.11	0.014	0.0000
number of badges	0.2085	0.011	0.0000
number of questions	-0.0648	0.006	0.0000
number of answers	-0.3571	0.028	0.0000
number of accepted answers	0.4465	0.038	0.0000
y=2	coefficient	std error	P>|z|
reputation	0.0159	0.025	0.5300
number of views	0.2774	0.056	0.0000
number of upvotes	0.032	0.006	0.0000
number of downvotes	-0.2061	0.021	0.0000
number of badges	0.287	0.011	0.0000
number of questions	-0.1764	0.008	0.0000
number of answers	-0.3868	0.029	0.0000
number of accepted answers	0.4879	0.039	0.0000
y=3	coefficient	std error	P>|z|
reputation	-0.2099	0.027	0.0000
number of views	0.8109	0.055	0.0000
number of upvotes	-0.0044	0.006	0.4550
number of downvotes	-1.1182	0.051	0.0000
number of badges	0.3458	0.011	0.0000
number of questions	-0.0276	0.006	0.0000
number of answers	-1.09	0.031	0.0000
number of accepted answers	1.4427	0.04	0.0000
R-squared	0.03881		

Table xx Summary of Pure Reputation Multinomial Logistic Regression Model

The R-squared of the model is only about 0.03811. If reputation score is a successful mechanism that can accurately measure the coding capabilities of users, the number of posts including both questions and answers which are highly correlated with the reputation (correlation=84%) should be positively correlated with the dependent variable, which is opposite to the results shown in the above model.

When group 0 has been set as the baseline, coefficient of the number of questions and number of answers are all negative, which suggest it is very likely that the active users on StackOverflow may not produce good content on development platforms.

Yang et al. (2014) investigated the very active users in the community versus users produced great content on the platform but are less active, and they concluded that the owl which refers to the users who provide useful answers but only infrequently so is the savvier group of users. The multinomial model shows that the sparrows who produced the majority of the content on StackOverflow may not be technology-savvy developers on development platforms.

5.4.3.2 Full MNLogit Model

MNLogit Regression Results			
y=1	coefficient	std err	P>|z|
reputation	0.4407	0.037	0.0000
number of views	0.199	0.053	0.0000
number of upvotes	0.0343	0.006	0.0000
number of downvotes	-0.1107	0.014	0.0000
number of badges	0.0633	0.018	0.0000
number of questions	-0.0304	0.008	0.0000
number of answers	-0.0512	0.037	0.1650
number of accepted answers	0.044	0.041	0.2860
answers to questions ratio	-0.2678	0.03	0.0000
the accepted ratio	0.0053	0	0.0000
answer PCA	0.2534	0.005	0.0000
question PCA	-0.439	0.015	0.0000
PageRank	0.0313	0.038	0.4070
number of areas	-0.0359	0.002	0.0000
y=2	coefficient	std err	P>|z|
reputation	0.5034	0.037	0.0000
number of views	0.2841	0.053	0.0000
number of upvotes	0.0229	0.006	0.0000
number of downvotes	-0.2198	0.022	0.0000
number of badges	0.1455	0.019	0.0000
number of questions	-0.1574	0.01	0.0000
number of answers	-0.099	0.037	0.0080
number of accepted answers	-0.2404	0.042	0.0000

answers to questions ratio	0.0244	0.027	0.3640
the accepted ratio	0.0106	0	0.0000
answer PCA	0.236	0.005	0.0000
question PCA	-0.7511	0.019	0.0000
PageRank	0.3053	0.037	0.0000
number of areas	-0.0352	0.002	0.0000
y=3	coefficient	std err	P>|z|
reputation	0.0851	0.039	0.0280
number of views	0.9791	0.052	0.0000
number of upvotes	-0.0227	0.006	0.0000
number of downvotes	-1.5535	0.062	0.0000
number of badges	0.5052	0.018	0.0000
number of questions	-0.0275	0.008	0.0000
number of answers	-0.5041	0.038	0.0000
number of accepted answers	0.3381	0.042	0.0000
answers to questions ratio	0.1393	0.027	0.0000
the accepted ratio	0.0102	0	0.0000
answer PCA	0.3437	0.005	0.0000
question PCA	-2.0778	0.028	0.0000
PageRank	0.349	0.037	0.0000
number of areas	-0.0649	0.002	0.0000
R-squared	0.08919		

Table xx Summary of Full Multinomial Logistic Regression Model

Incorporating more dimensions that better address the quality of participation and may, therefore, imply users' coding capabilities on StackOverflow in the above model, the performance of the prediction model increased significantly. In this case, the R-squared increased to 0.08771 which is more than two times of the R-Squared shown in the pure reputation-linked model.

The similar pattern discussed before shown in the summary of the reduced model as well. The users belonging to a group which is believed to have better developers may not post as frequently as the developers with poorer performance. The coefficient of the number of questions is negative when users in group 0 are compared with users belong to the other three groups, which may suggest that the people with better coding expertise posted questions less often in the Q&A community. The positive sign of the coefficient of the ratio of the number of answers to questions suggests that the users believed to be better developers post more answers and ask fewer questions on the Q&A platform.

Similar to what has been observed in the sixth linear regression model, accepted ratio and quality of answers are positively correlated with the coding expertise of users. Therefore, to measure the coding expertise of users on the Q&A platform, the researchers may want to take the quality of answers into consideration. The larger absolute value of accepted answers may not necessarily indicate the outstanding coding capabilities of users. In addition to the absolute value, the ratio of the accepted number to the total numbers seems to be a more accurate metrics that can assess the average quality of users' posts.

In both linear regression and multinomial regression models, debatableness of questions did not play a significant role and the sign of its coefficient is always negative. Different from the initial expectations -- users with higher expertise tend to answer a more debated question. However, the way of modelling the debatableness of questions may be biased. The assumption was that the questions received more answers and being viewed by more people will be considered as more debated questions. Nevertheless, there are other potential reasons may lead to questions with a large volume of answers. For example, the debugging questions at the entry level may have various solutions. Therefore, developers with different expertise may provide solutions from different perspectives.

The correlation matrix in figure 5 suggests that PageRank is highly correlated with Reputation and activeness of users on StackOverflow. However, different from reputation, in most of the above linear regression and multinomial logistic models, PageRank is positively correlated with coding expertise. PageRank taking the time discount factor into consideration can capture the interaction among all users in the community. The very inactive users with low reputation score may have a relatively high PageRank by answering questions asked by an influential user even for only one time. It is true that activeness is still an important way for users to increase their PageRank, but it is not the only way anymore. The very inactive users who did post answers with good quality and who are able to answer some difficult questions do benefit from this mechanism.

The number of areas and topics that users being involved in the community is another significant variable contributing to better predict the expertise level of users. The negative coefficient suggests that people with high expertise usually have their own focus. The users at the entry level may spend more time on exploring different topics in the community. It is

important to take note that the finding is consistent with the fourth trick mentioned by Bosu (2013) to quickly improve reputation scores – contributing to diverse areas. It is likely that there does exist, as what Bosu contend, users who earn high reputation scores by such trick which enable them to take advantage of the loophole and market themselves as expert developers.

5.4.3.3 Results Discussion

5.4.3.3.1 Reputation MNLogit Model

Multinomial Logistic Regression Test Accuracy: 0.3834

Test Prediction	**1**	**2**	**3**	**4**	**Total**	**Test Accuracy**
0	10214	462	1305	666	12647	**0.8076**
1	7884	810	2529	1567	12647	**0.0633**
2	6918	726	3256	1613	12647	**0.2602**
3	4275	673	2666	5176	12647	**0.4047**

Table 1. Prediction Accuracy of Pure Reputation-linked Model

The model built up only based on reputation-linked features performed poorly in terms of predicting the coding expertise of users. The coders with more extreme capabilities who are users belong to group 0 and group 3 are generally more easily to be captured by the model. However, the users in the middle layers cannot be identified by the pure reputation models. Especially for users belonging to group1, only 6.33% of them have been correctly identified, most of them were misclassified to other three groups, suggesting that the pure reputation model did not successfully learn patterns of users in the middle layer.

5.4.3.3.1 Full MNLogit Model

Multinomial Logistic Regression Test Accuracy: 0.4947

Test Prediction	1	2	3	4	Total	Test Accuracy
0	7596	2778	2140	133	12647	**0.6006**
1	2596	4505	3536	2153	12647	**0.3522**
2	1963	3589	4321	2640	12647	**0.3453**
3	0	1970	2142	8678	12647	**0.6785**

Table 2. Prediction Accuracy of Reputation &Other Potential Factors Model

After adding more factors into the Multinomial Logistic Regression model, the accuracy of prediction improved though the prediction accuracy of the middle two groups is still much lower than the group containing the best performers and the worst performers. Therefore, it may be concluded that the users with average performance on GitHub who are believed to be programmers with average expertise are more difficult to be identified by their activities on StackOverflow.

5.5 Deep Learning and Improvement

After adding more dimensions to the Multinomial Logistic Regression model which was constructed to predict the coding capabilities of StackOverflow user, the prediction accuracy is about 50% which can be further improved by utilizing some more sophisticated or more suitable machine learning models.

5.5.1 Support Vector Machines

With the same features in the 5.4.3.2 model, the linear SVM model used to classify data points was constructed. Given a training dataset, the SVM training algorithm constructs a model that separate categories by specifying clear gaps which are as wide as possible. Testing data points will then be mapped into space and be labelled by the group number determined by which areas they fall into.

Linear SVM Test Accuracy: 0.6144

Test Prediction	1	2	3	4	Total	Test Accuracy
0	9785	1698	1246	38	12767	**0.7664**
1	3767	4386	2896	1681	12767	**0.3435**
2	2888	2636	5094	1999	12767	**0.3990**
3	0	250	264	12112	12767	**0.9487**

Table 3. Prediction Accuracy of Linear SVM

Nonlinear SVM Test Accuracy: 0.7061

Test Prediction	1	2	3	4	Total	Test Accuracy
0	10106	1555	935	171	12767	**0.7916**
1	3581	6626	1019	1504	12767	**0.5190**
2	2684	1056	7173	1704	12767	**0.5618**
3	1	336	137	12152	12767	**0.9518**

Table 4. Prediction Accuracy of None-linear SVM

The above two models clearly show that the nonlinear SVM model which is non-parametric performed much better than the linear SVM model and Multinomial Logistic Regression models which are all parametric models. Therefore, it may be concluded that, given the nature of the current sample dataset, it may be wiser to try out more machine learning models that do not make any assumption about the sample data.

5.5.2 K-Nearest Neighbors (K-NN)

K-NN is a non-parametric method used for classification. The output which is the class membership of a testing data point is determined by the plurality vote of its neighbours. As one of the simplest machine learning model, the accuracy of K-NN reached 84% which is

much higher than some more sophisticated model. The main advantage of such non-parametric model is that there is no specific assumption that the feeding data should adhere to. Although some data transformation and feature-selection procedures have been done, some assumptions made by parametric models cannot be rigorously satisfied such as no multicollinearity among all independent variables. As a result, the lazy learning algorithm -- K-NN outperformed many other more computationally expensive parametric algorithms.

K-NN Test Accuracy: 0.8438

Test Prediction	**1**	**2**	**3**	**4**	**Total**	**Test Accuracy**
0	8557	2591	1542	77	12767	**0.6702**
1	1299	10587	730	114	12767	**0.8202**
2	599	456	11449	113	12767	**0.8968**
3	0	18	12	12616	12767	**0.9882**

Table 5. Prediction Accuracy of K-NN

5.5.3 Decision Tree Learning

Decision tree learning is a simple learner commonly used in data mining and classification. A tree can be trained by separating the original dataset into subsets based on the attributes value test. Recursive partitioning has been used to repeat the dividing process and it stops when every subset only contains nodes classified as the same group or when the performance of the model converges even if more values are being added.

Similar to K-NN, Decision tree learning is also a simple and non-parametric model. It also yields satisfying accuracy which is about 83% and outperformed parametric models.

Decision Trees Test Accuracy: 0.8264

Test Prediction	**1**	**2**	**3**	**4**	**Total**	**Test Accuracy**

0	10541	1500	725	1	12767	**0.8256**
1	1265	9134	2185	146	12767	**0.7154**
2	553	1898	10034	132	12767	**0.7859**
3	0	69	62	12495	12767	**0.9767**

Table 6. Prediction Accuracy of Decision Trees

5.5.4 Neural Network -- Multi-layer Perceptron (MLP)

MLP is an implementation of Neural Network that learns a function by training a dataset. Given a set of features and a target dependent variable, it can learn a nonlinear function for classification. Different from logistic regression, there may be multiple hidden layers between the input and the output layer. Figure xx shows the MLP with one hidden layer.

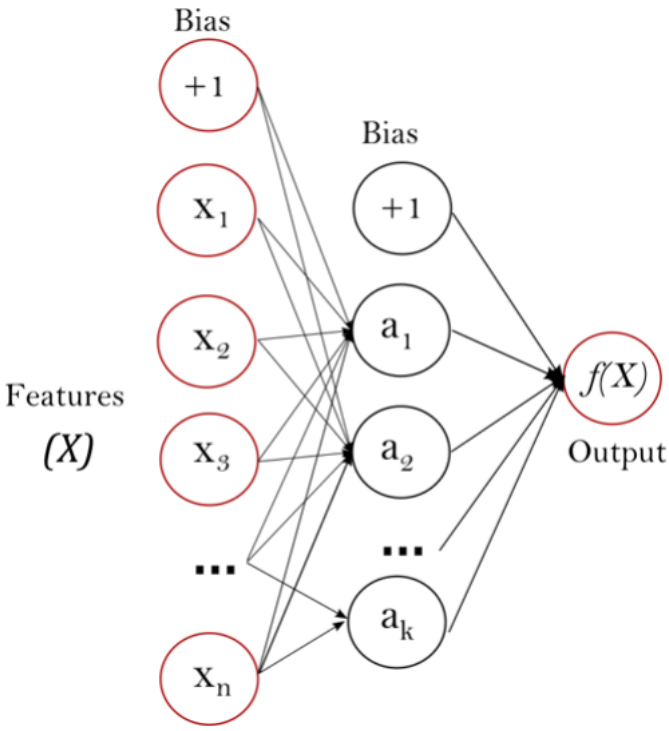


Figure 7 Multi-layer Perceptron Diagram

MLP Test Accuracy: 0.6625

Test Prediction	**1**	**2**	**3**	**4**	**Total**	**Test Accuracy**

0	10773	891	1102	1	12767	**0.8438**
1	4089	4767	3350	524	12767	**0.3734**
2	2847	2893	6238	639	12767	**0.4886**
3	0	226	346	12054	12767	**0.9442**

Table 7. Prediction Accuracy of MLP

The current accuracy of MLP is about 66% which is not as good as some weak and simple learners. By further twigging the hyperparameters including the number of layers, the number of neurons in each layer, iterations, and activation functions, the accuracy may increase. However, compared to other simple algorithms, MLP which is demanding for the computational power may not be the most suitable machine learning model in this case.

5.5.5 Random Forests

Random Forests is an ensemble learning algorithm for classification by constructing multiple decision trees at the training phase and assign the test data point to the class that is the mode of all predicted classes. Compared to decisions trees, random forests reduce the variance of the model and avoid overfitting to the training set.

Random Forests Test Accuracy: 0.8903

Test Prediction	**1**	**2**	**3**	**4**	**Total**	**Test Accuracy**
0	11674	710	383	0	12767	**0.9144**
1	1511	10229	961	29	12767	**0.8012**
2	583	1052	10945	37	12767	**0.8573**
3	0	9	1	12616	12767	**0.9882**

Table 8. Prediction Accuracy of Random Forests

5.5.6 Extra Tree Classifier

Extra Tree Classifier, also known as Extremely Randomized Trees, is a variant of random forests. Instead of computing the locally optimal split combination, extra trees select a random value for each split. Therefore, Extra Tree Classifier is able to generate more diversified trees in shorter time.

Extra Tree Classifier Test Accuracy: 0.8791

Test Prediction	**1**	**2**	**3**	**4**	**Total**	**Test Accuracy**
0	10819	1305	641	2	12767	**0.8474**
1	1589	10278	830	33	12767	**0.8050**
2	517	888	11181	31	12767	**0.8758**
3	0	6	3	12617	12767	**0.9883**

Table 9. Prediction Accuracy of Extra Tree Classifier

5.5.7 XGBoosting Classifier

Gradient Boosting and Random Forests are all ensemble learning methods and make predictions by the output of a group of models. Different from Random Forests that train each model independently, Gradient Boosting train models step by step and addressed the data points which were misclassified in the previous step.

XGBoosting stands for Extreme Gradient Boosting, which is a part of Boosting machine learning techniques. Compared to the original Gradient Boosting, XGBoost managed to fasten the training process by the implementation of parallelization of tree construction. Moreover, the performance of XGBoost also improved since it incorporates penalization mechanism, proportional shrinking of leaf nodes, and extra randomization factors.

However, compared to Random Forests, XGBoosting is more demanding for the tuning process. Tale 10 shows the testing results when the maximum depth of the trees has been set to the default value of Python package which is 3.

XGBoost Classifier (max_depth = 3) Test Accuracy: 0.6398

Test Prediction	**1**	**2**	**3**	**4**	**Total**	**Test Accuracy**
0	11368	452	947	38	12767	**0.8904**
1	3543	3741	3193	2253	12767	**0.2930**
2	2330	2302	5238	2747	12767	**0.4103**
3	0	175	123	12328	12767	**0.9656**

Table 10. Prediction Accuracy of XGBoost (max_depth=3)

It can be clearly seen that, with a group of shallow trees, XGBoosting performed poorly in prediction. Boosting works very well when the basic weak learner has a high bias but low variance, and Boosting can help to reduce the error due to variance significantly. In this case, the group of shallow trees failed to make high-quality predictions, which suggests that the error due to variance may not be very severe. It is the error due to bias that substantially lowers the prediction accuracy of the model. Therefore, by increasing the depth of trees, the performance of the model keep increasing. When the maximum depth of trees was set to 20, XGBoosting yields a prediction result of 93% accuracy level.

XGBoosting Classifier (max_depth = 20) Test Accuracy: 0.9312

Test Prediction	**1**	**2**	**3**	**4**	**Total**	**Test Accuracy**
0	12451	169	147	0	12767	**0.9752**
1	1128	10856	706	40	12767	**0.8531**
2	354	623	11595	45	12767	**0.9082**

3	0	5	1	12620	12767	**0.9885**

Table 11. Prediction Accuracy of XGBoost (max_depth=20)

5.5.8 Summary of Models

Overall, the non-parametric models which do not make any assumption about the feeding dataset outperformed the parametric models. Ensemble learning generally performed better than simple models. XGBoosting outperformed all other models and managed to predict 93% of the testing data points correctly.

6. Limitation

Due to the time constraint and the limited computational power, there is still room for the project to be further improved in the near future.

First of all, although GitHub is a platform that can be used to assess people's coding capabilities in a more practical way, there may be more dimensions can be investigated in order to build up a more sophisticated model to find out the ground truth of coding expertise of each individual. Moreover, there are many coding platforms such as HireVue may rank their users based on users' activeness as well as the quality of works. Data obtained from such a platform may be a more accurate and convincing ground truth.

To ensure the efficiency and accuracy of cross-platform analysis, only about 89,000 users on both platforms have been matched based on their email addresses. The sample size can be enlarged by closely looking at the profiles of users including the location of users, their social network accounts, and the self-created websites.

Last but not least, it is possible that the dataset can be further sampled and transformed in a more appropriate way. Thereafter, any assumption of parametric machines learning models will not be violated. With the better performance of parametric models, the coefficient of independent variables can be better interpreted and transferred into a more complete cross-platform study.

7. Conclusion

To investigate how to assess the coding expertise of users on StackOverflow, more dimensions of users' activities in addition to reputation-linked activities have been explored.

Firstly, the study shows that there is no conclusive relationship between activeness of users on Q&A platform with their performance on the developing platform. It is true that reputation scores calculated by StackOverflow managed to capture some features of users that can link to their coding capabilities such as the number of badges each individual won. However, there still exists a gap between reputation scores and coding expertise of the developers. Therefore, the reputation mechanism can be further improved by taking more potential factors into consideration. Based on the Multinomial Logistic Regression results, the accepted ratio, utility of answers, and PageRank can be included to measure the capabilities of developers. The current reputation mechanism may misjudge two groups of users. The first group consists of people who are very active on the Q&A website by posting many questions and answers with relatively low quality. However, they can still earn high reputation scores which are the reward of activeness granted by StackOverflow. From the perspectives of researchers and employers who may misinterpret the significance of reputation scores, the reputation causes the misperception of the coding expertise of such users. The second group consists of users who are able to make knowledge contribution with high quality to the community, but they may not as active as most of the users on StackOverflow. In this case, the inactive developers with good capabilities may not be well identified by the reputation mechanism.

After more dimensions of users' activities on StackOverflow have been captured and included in the model, the performance of models that predict the coding expertise of users significantly increased. The top performers and the worst performers are more easily to be identified since their behaviour patterns on the Q&A platform and the developing platform are more evident and correlated. By looking at the prediction accuracy tables, it can be found that it is the developers in the middle layer who are the average performers that are more difficult to be identified.

In summary, the coding expertise of users measured based on their activities on GitHub can be well predicted by their activities on StackOverflow when both activeness and quality of

contribution are taken into consideration. Among all models, the non-parametric and ensemble learning models generally have good performance and outperformed other parametric models that make a strong assumption about the data collected from StackOverflow. Therefore, to more accurately measure the capabilities of users instead of mostly focusing on rewarding the activeness of users, StackOverflow may want to include more dimensions into their reputation mechanisms which identify the users who make the contribution to the community by not only actively participating but also committing content with high quality.

8. References

Anderson, A., Huttenlocher, D., Kleinberg, J., & Leskovec, J. (2012). Discovering value from community activity on focused question answering sites. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 12*. doi:10.1145/2339530.2339665

Bogers, M., Zobel, A.-K., Afuah, A., Almirall, E., Brunswicker, S., Dahlander, L., Frederiksen, L., Gawer, A., Gruber, M., Haefliger, S., Hagedoorn, J., Hilgers, D., Laursen, K., Magnusson, M. G., Majchrzak, A., McCarthy, I. P., Moeslein, K. M., Nambisan, S., Piller, F. T., Radziwon, A., Rossi- Lamastra, C., Sims, J., and Ter Wal, A. L. J. 2017. The open innovation research landscape: established perspectives and emerging themes across different levels of analysis. Industry and Innovation 24(1) 8–40.

Bosu, A., Corley, C. S., Heaton, D., Chatterji, D., Carver, J. C., & Kraft, N. A. (2013). *Building reputation in StackOverflow: An empirical investigation.* 2013 10th Working Conference on Mining Software Repositories (MSR). doi:10.1109/msr.2013.6624013

C. Bird, A. Gourley, P. T. Devanbu, M. Gertz, and A. Swaminathan, *Mining email social networks*, in MSR. ACM, 2006, pp. 137–143.

Faraj, S., Kudaravalli, S., and Wasko, M. 2015. Leading Collaboration in Online Communities. MIS Quarterly 39(2) 393–412.

Gray, E. (2014, October 14). *Introducing: Stack Overflow Careers*. Retrieved from https://www.stackoverflowbusiness.com/blog/2014/10/14/introducing-stack-overflow-careers

Gousios, G., & Spinellis, D. (2012). *GHTorrent: Githubs data from a firehose*. 2012 9th IEEE Working Conference on Mining Software Repositories (MSR). doi:10.1109/msr.2012.6224294

Hart, K., & Sarma, A. (2014). *Perceptions of answer quality in an online technical question and answer forum.* Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE 2014. doi:10.1145/2593702.2593703

Morrison, P., & Murphy-Hill, E. (2013). *Is programming knowledge related to age?* An exploration of stack overflow. 2013 10th Working Conference on Mining Software Repositories (MSR). doi:10.1109/msr.2013.6624008

Movshovitz-Attias, D., Movshovitz-Attias, Y., Steenkiste, P., & Faloutsos, C. (2013). Analysis of the reputation system and user contributions on a question answering website. Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining - ASONAM 13. doi:10.1145/2492517.2500242

Ponzanelli, L., Mocci, A., Bacchelli, A., Lanza, M., & Fullerton, D. (2014). *Improving Low Quality Stack Overflow Post Detection*. 2014 IEEE International Conference on Software Maintenance and Evolution. doi:10.1109/icsme.2014.90

Safadi, H., Johnson, S. L., & Faraj, S. (2018). *Who Contributes Knowledge? Embeddedness and Marginality in Online Communities*. Academy of Management Proceedings,2018(1), 11588. doi:10.5465/ambpp.2018.35

Vasilescu, B., Filkov, V., & Serebrenik, A. (2013). *StackOverflow and GitHub: Associations between Software Development and Crowdsourced Knowledge*. 2013 International Conference on Social Computing. doi:10.1109/socialcom.2013.35

Winter, T. (2017, June 13). *How to source developers from Stack Overflow?* Retrieved from https://devskiller.com/source-developers-stack-overflow/

Wu, Y., Yang, Y., Zhao, Y., Lu, H., Zhou, Y., & Xu, B. (2014). The influence of developer quality on software fault-proneness prediction. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6895411

Yang, J., Tao, K., Bozzon, A., & Houben, G. (2014). Sparrows and Owls: Characterisation of Expert Behaviour in StackOverflow. *User Modeling, Adaptation, and Personalization Lecture Notes in Computer Science,*266-277. doi:10.1007/978-3-319-08786-3_23

9. Appendix

9.1 OLS

9.1.1 Number of Commits

9.1.1.1 Full Model

OLS Regression Results

Dep. Variable:	num_commits_x	R-squared:	0.110
Model:	OLS	Adj. R-squared:	0.109
Method:	Least Squares	F-statistic:	421.5
Date:	Mon, 01 Apr 2019	Prob (F-statistic):	0.00
Time:	21:05:07	Log-Likelihood:	-77268.
No. Observations:	47939	AIC:	1.546e+05
Df Residuals:	47925	BIC:	1.547e+05
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
reputation	-0.0032	0.012	-0.264	0.792	-0.027	0.021
view	0.0389	0.011	3.457	0.001	0.017	0.061
upvotes	-0.0026	0.003	-0.782	0.434	-0.009	0.004
downvotes	-0.0330	0.007	-4.490	0.000	-0.047	-0.019
num_qns	-0.0202	0.006	-3.418	0.001	-0.032	-0.009
num_ans	-0.1258	0.015	-8.358	0.000	-0.155	-0.096
num_badges	0.0435	0.010	4.346	0.000	0.024	0.063
num_accept	0.0974	0.018	5.465	0.000	0.062	0.132
accept_ratio	0.0050	0.000	15.835	0.000	0.004	0.006
qns_pca	-0.0020	0.004	-0.439	0.661	-0.011	0.007
ans_pca	0.1174	0.003	38.952	0.000	0.111	0.123
pagerank	0.0707	0.013	5.281	0.000	0.044	0.097
num_areas	-0.0011	0.001	-1.147	0.252	-0.003	0.001
ans_qns_ratio	0.0052	0.009	0.581	0.561	-0.012	0.023

Omnibus:	130719.902	Durbin-Watson:	0.858
Prob(Omnibus):	0.000	Jarque-Bera (JB):	8849277038.866
Skew:	33.802	Prob(JB):	0.00
Kurtosis:	2106.734	Cond. No.	103.

9.1.1.2 Model with Pure Reputation-Linked Features

OLS Regression Results

Dep. Variable:	num_commits_x	R-squared:	0.043
Model:	OLS	Adj. R-squared:	0.043
Method:	Least Squares	F-statistic:	270.0
Date:	Mon, 01 Apr 2019	Prob (F-statistic):	0.00
Time:	23:24:02	Log-Likelihood:	-78995.
No. Observations:	47939	AIC:	1.580e+05
Df Residuals:	47931	BIC:	1.581e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
reputation	-0.2312	0.011	-21.107	0.000	-0.253	-0.210
view	-0.0064	0.011	-0.556	0.578	-0.029	0.016
upvotes	-0.0089	0.003	-2.558	0.011	-0.016	-0.002
downvotes	-0.0323	0.008	-4.251	0.000	-0.047	-0.017
num_qns	-0.0954	0.006	-16.147	0.000	-0.107	-0.084
num_ans	-0.0712	0.014	-5.202	0.000	-0.098	-0.044
num_badges	0.2712	0.008	32.703	0.000	0.255	0.287
num_accept	0.1426	0.016	9.086	0.000	0.112	0.173

Omnibus:	127954.018	Durbin-Watson:	0.791
Prob(Omnibus):	0.000	Jarque-Bera (JB):	7409225638.522
Skew:	31.964	Prob(JB):	0.00
Kurtosis:	1927.900	Cond. No.	13.9

9.1.1.2 Reduced Model

OLS Regression Results

Dep. Variable:	num_commits_x	R-squared:	0.108
Model:	OLS	Adj. R-squared:	0.108
Method:	Least Squares	F-statistic:	826.9
Date:	Mon, 01 Apr 2019	Prob (F-statistic):	0.00
Time:	21:05:11	Log-Likelihood:	-77319.
No. Observations:	47939	AIC:	1.547e+05
Df Residuals:	47932	BIC:	1.547e+05
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
view	0.0398	0.010	3.854	0.000	0.020	0.060
downvotes	-0.0334	0.007	-4.596	0.000	-0.048	-0.019
num_badges	0.0006	0.004	0.142	0.887	-0.008	0.009
num_accept	0.0043	0.011	0.388	0.698	-0.017	0.026
accept_ratio	0.0053	0.000	17.647	0.000	0.005	0.006
ans_pca	0.1160	0.003	40.500	0.000	0.110	0.122
pagerank	0.0265	0.012	2.298	0.022	0.004	0.049

Omnibus:	130616.688	Durbin-Watson:	0.851
Prob(Omnibus):	0.000	Jarque-Bera (JB):	8778176964.756
Skew:	33.733	Prob(JB):	0.00
Kurtosis:	2098.262	Cond. No.	71.3

9.1.2 Number of Projects

9.1.2.1 Full Model

OLS Regression Results

Dep. Variable:	num_projects	R-squared:	0.222
Model:	OLS	Adj. R-squared:	0.221
Method:	Least Squares	F-statistic:	974.5
Date:	Sun, 31 Mar 2019	Prob (F-statistic):	0.00
Time:	16:08:10	Log-Likelihood:	-66735.
No. Observations:	47939	AIC:	1.335e+05
Df Residuals:	47925	BIC:	1.336e+05
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
reputation	-0.0134	0.010	-1.371	0.170	-0.032	0.006
view	-0.0111	0.009	-1.227	0.220	-0.029	0.007
upvotes	0.0010	0.003	0.357	0.721	-0.004	0.006
downvotes	-0.0128	0.006	-2.168	0.030	-0.024	-0.001
num_qns	-0.0350	0.005	-7.387	0.000	-0.044	-0.026
num_ans	-0.0782	0.012	-6.472	0.000	-0.102	-0.055
num_badges	0.0710	0.008	8.839	0.000	0.055	0.087
num_accept	0.0119	0.014	0.830	0.406	-0.016	0.040
accept_ratio	0.0035	0.000	13.742	0.000	0.003	0.004
qns_pca	0.0212	0.004	5.902	0.000	0.014	0.028
ans_pca	0.1312	0.002	54.221	0.000	0.126	0.136
pagerank	0.0488	0.011	4.546	0.000	0.028	0.070
num_areas	0.0081	0.001	10.445	0.000	0.007	0.010
ans_qns_ratio	-0.0051	0.007	-0.700	0.484	-0.019	0.009

Omnibus:	119879.777	Durbin-Watson:	1.558
Prob(Omnibus):	0.000	Jarque-Bera (JB):	10922672611.297
Skew:	26.575	Prob(JB):	0.00
Kurtosis:	2340.832	Cond. No.	103.

9.1.2.2 Model with Pure Reputation-Linked Features

OLS Regression Results

Dep. Variable:	num_projects	R-squared:	0.107
Model:	OLS	Adj. R-squared:	0.107
Method:	Least Squares	F-statistic:	721.2
Date:	Mon, 01 Apr 2019	Prob (F-statistic):	0.00
Time:	23:27:45	Log-Likelihood:	-70015.
No. Observations:	47939	AIC:	1.400e+05
Df Residuals:	47931	BIC:	1.401e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
reputation	-0.2873	0.009	-31.637	0.000	-0.305	-0.270
view	-0.0793	0.009	-8.359	0.000	-0.098	-0.061
upvotes	-0.0039	0.003	-1.365	0.172	-0.010	0.002
downvotes	-0.0091	0.006	-1.438	0.150	-0.021	0.003
num_qns	-0.1220	0.005	-24.907	0.000	-0.132	-0.112
num_ans	0.0251	0.011	2.208	0.027	0.003	0.047
num_badges	0.3675	0.007	53.445	0.000	0.354	0.381
num_accept	-0.0059	0.013	-0.451	0.652	-0.031	0.020

Omnibus:	116488.266	Durbin-Watson:	1.371
Prob(Omnibus):	0.000	Jarque-Bera (JB):	8830259813.410
Skew:	24.706	Prob(JB):	0.00
Kurtosis:	2104.977	Cond. No.	13.9

9.1.2.2 Reduced Model

OLS Regression Results

Dep. Variable:	num_projects	R-squared:	0.221
Model:	OLS	Adj. R-squared:	0.221
Method:	Least Squares	F-statistic:	1703.
Date:	Sun, 31 Mar 2019	Prob (F-statistic):	0.00
Time:	11:44:57	Log-Likelihood:	-66742.
No. Observations:	47939	AIC:	1.335e+05
Df Residuals:	47931	BIC:	1.336e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
num_qns	-0.0302	0.004	-7.420	0.000	-0.038	-0.022
num_ans	-0.0802	0.008	-10.119	0.000	-0.096	-0.065
num_badges	0.0598	0.005	11.419	0.000	0.050	0.070
accept_ratio	0.0036	0.000	14.234	0.000	0.003	0.004
qns_pca	0.0211	0.004	5.881	0.000	0.014	0.028
ans_pca	0.1313	0.002	54.467	0.000	0.127	0.136
pagerank	0.0541	0.009	5.884	0.000	0.036	0.072
num_areas	0.0086	0.001	12.306	0.000	0.007	0.010

Omnibus:	119864.493	Durbin-Watson:	1.558
Prob(Omnibus):	0.000	Jarque-Bera (JB):	10912201555.746
Skew:	26.566	Prob(JB):	0.00
Kurtosis:	2339.711	Cond. No.	73.6

9.1.3 Number of Accepted Pull Request

9.1.3.1 Full Model

OLS Regression Results

Dep. Variable:	num_acc_pr	R-squared:	0.048
Model:	OLS	Adj. R-squared:	0.047
Method:	Least Squares	F-statistic:	171.6
Date:	Sun, 31 Mar 2019	Prob (F-statistic):	0.00
Time:	16:08:34	Log-Likelihood:	-84906.
No. Observations:	47939	AIC:	1.698e+05
Df Residuals:	47925	BIC:	1.700e+05
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
reputation	-0.0320	0.014	-2.243	0.025	-0.060	-0.004
view	0.0506	0.013	3.836	0.000	0.025	0.076
upvotes	-0.0143	0.004	-3.636	0.000	-0.022	-0.007
downvotes	-0.0242	0.009	-2.809	0.005	-0.041	-0.007
num_qns	-0.0408	0.007	-5.890	0.000	-0.054	-0.027
num_ans	-0.1180	0.018	-6.683	0.000	-0.153	-0.083
num_badges	0.0638	0.012	5.437	0.000	0.041	0.087
num_accept	0.1116	0.021	5.339	0.000	0.071	0.153
accept_ratio	0.0039	0.000	10.445	0.000	0.003	0.005
qns_pca	-0.0029	0.005	-0.545	0.586	-0.013	0.007
ans_pca	0.0775	0.004	21.915	0.000	0.071	0.084
pagerank	0.0520	0.016	3.315	0.001	0.021	0.083
num_areas	0.0005	0.001	0.401	0.689	-0.002	0.003
ans_qns_ratio	0.0294	0.011	2.779	0.005	0.009	0.050

Omnibus:	120899.902	Durbin-Watson:	1.514
Prob(Omnibus):	0.000	Jarque-Bera (JB):	3725031896.736
Skew:	27.770	Prob(JB):	0.00
Kurtosis:	1367.478	Cond. No.	103.

9.1.3.2 Model with Pure Reputation-Linked Features

OLS Regression Results

Dep. Variable:	num_acc_pr	R-squared:	0.022
Model:	OLS	Adj. R-squared:	0.022
Method:	Least Squares	F-statistic:	134.4
Date:	Mon, 01 Apr 2019	Prob (F-statistic):	2.82e-224
Time:	21:06:42	Log-Likelihood:	-85546.
No. Observations:	47939	AIC:	1.711e+05
Df Residuals:	47931	BIC:	1.712e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
reputation	-0.2006	0.013	-15.973	0.000	-0.225	-0.176
view	0.0152	0.013	1.158	0.247	-0.011	0.041
upvotes	-0.0187	0.004	-4.703	0.000	-0.026	-0.011
downvotes	-0.0235	0.009	-2.698	0.007	-0.041	-0.006
num_qns	-0.0948	0.007	-14.004	0.000	-0.108	-0.082
num_ans	-0.0754	0.016	-4.804	0.000	-0.106	-0.045
num_badges	0.2296	0.010	24.151	0.000	0.211	0.248
num_accept	0.1551	0.018	8.621	0.000	0.120	0.190

Omnibus:	120081.733	Durbin-Watson:	1.478
Prob(Omnibus):	0.000	Jarque-Bera (JB):	3510497205.671
Skew:	27.303	Prob(JB):	0.00
Kurtosis:	1327.575	Cond. No.	13.9

9.1.3.2 Reduced Model

OLS Regression Results

Dep. Variable:	num_acc_pr	R-squared:	0.048
Model:	OLS	Adj. R-squared:	0.047
Method:	Least Squares	F-statistic:	217.8
Date:	Sun, 31 Mar 2019	Prob (F-statistic):	0.00
Time:	11:55:21	Log-Likelihood:	-84909.
No. Observations:	47939	AIC:	1.698e+05
Df Residuals:	47928	BIC:	1.699e+05
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
view	0.0421	0.013	3.338	0.001	0.017	0.067
upvotes	-0.0131	0.004	-3.357	0.001	-0.021	-0.005
downvotes	-0.0234	0.009	-2.721	0.007	-0.040	-0.007
num_qns	-0.0330	0.006	-5.376	0.000	-0.045	-0.021
num_ans	-0.1253	0.016	-7.724	0.000	-0.157	-0.093
num_badges	0.0493	0.008	5.879	0.000	0.033	0.066
num_accept	0.0997	0.020	5.043	0.000	0.061	0.138
accept_ratio	0.0041	0.000	11.351	0.000	0.003	0.005
ans_pca	0.0787	0.003	23.331	0.000	0.072	0.085
pagerank	0.0635	0.015	4.266	0.000	0.034	0.093
ans_qns_ratio	0.0301	0.011	2.852	0.004	0.009	0.051

Omnibus:	120903.077	Durbin-Watson:	1.515
Prob(Omnibus):	0.000	Jarque-Bera (JB):	3726506203.608
Skew:	27.772	Prob(JB):	0.00
Kurtosis:	1367.749	Cond. No.	92.8

9.1.4 Number of Followers

9.1.4.1 Full Model

OLS Regression Results

Dep. Variable:	num_followers	R-squared:	0.031
Model:	OLS	Adj. R-squared:	0.031
Method:	Least Squares	F-statistic:	110.0
Date:	Sun, 31 Mar 2019	Prob (F-statistic):	2.16e-315
Time:	16:08:53	Log-Likelihood:	-66925.
No. Observations:	47939	AIC:	1.339e+05
Df Residuals:	47925	BIC:	1.340e+05
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
reputation	-0.0108	0.010	-1.104	0.270	-0.030	0.008
view	0.1189	0.009	13.119	0.000	0.101	0.137
upvotes	-0.0147	0.003	-5.428	0.000	-0.020	-0.009
downvotes	-0.0439	0.006	-7.420	0.000	-0.055	-0.032
num_qns	-0.0266	0.005	-5.589	0.000	-0.036	-0.017
num_ans	-0.0684	0.012	-5.639	0.000	-0.092	-0.045
num_badges	0.0350	0.008	4.343	0.000	0.019	0.051
num_accept	0.0595	0.014	4.142	0.000	0.031	0.088
accept_ratio	0.0032	0.000	12.350	0.000	0.003	0.004
qns_pca	0.0026	0.004	0.734	0.463	-0.004	0.010
ans_pca	0.0279	0.002	11.467	0.000	0.023	0.033
pagerank	0.0038	0.011	0.350	0.726	-0.017	0.025
num_areas	0.0005	0.001	0.584	0.559	-0.001	0.002
ans_qns_ratio	0.0225	0.007	3.104	0.002	0.008	0.037

Omnibus:	150555.297	Durbin-Watson:	1.868
Prob(Omnibus):	0.000	Jarque-Bera (JB):	26970452851.964
Skew:	49.738	Prob(JB):	0.00
Kurtosis:	3676.213	Cond. No.	103.

9.1.4.2 Model with Pure Reputation-Linked Features

OLS Regression Results

Dep. Variable:	num_followers	R-squared:	0.017
Model:	OLS	Adj. R-squared:	0.017
Method:	Least Squares	F-statistic:	105.7
Date:	Mon, 01 Apr 2019	Prob (F-statistic):	1.24e-175
Time:	21:06:43	Log-Likelihood:	-67264.
No. Observations:	47939	AIC:	1.345e+05
Df Residuals:	47931	BIC:	1.346e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
reputation	-0.0922	0.009	-10.749	0.000	-0.109	-0.075
view	0.0989	0.009	11.040	0.000	0.081	0.117
upvotes	-0.0170	0.003	-6.269	0.000	-0.022	-0.012
downvotes	-0.0435	0.006	-7.304	0.000	-0.055	-0.032
num_qns	-0.0555	0.005	-12.000	0.000	-0.065	-0.046
num_ans	-0.0612	0.011	-5.708	0.000	-0.082	-0.040
num_badges	0.1235	0.006	19.013	0.000	0.111	0.136
num_accept	0.0801	0.012	6.523	0.000	0.056	0.104

Omnibus:	150261.236	Durbin-Watson:	1.850
Prob(Omnibus):	0.000	Jarque-Bera (JB):	26552729528.848
Skew:	49.464	Prob(JB):	0.00
Kurtosis:	3647.650	Cond. No.	13.9

9.1.4.2 Reduced Model

OLS Regression Results

Dep. Variable:	num_followers	R-squared:	0.031
Model:	OLS	Adj. R-squared:	0.031
Method:	Least Squares	F-statistic:	153.6
Date:	Sun, 31 Mar 2019	Prob (F-statistic):	5.89e-319
Time:	12:00:05	Log-Likelihood:	-66926.
No. Observations:	47939	AIC:	1.339e+05
Df Residuals:	47929	BIC:	1.340e+05
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
view	0.1149	0.009	13.259	0.000	0.098	0.132
upvotes	-0.0141	0.003	-5.272	0.000	-0.019	-0.009
downvotes	-0.0434	0.006	-7.347	0.000	-0.055	-0.032
num_qns	-0.0231	0.004	-5.710	0.000	-0.031	-0.015
num_ans	-0.0671	0.011	-6.334	0.000	-0.088	-0.046
num_badges	0.0299	0.005	5.644	0.000	0.020	0.040
num_accept	0.0575	0.013	4.528	0.000	0.033	0.082
accept_ratio	0.0033	0.000	13.343	0.000	0.003	0.004
ans_pca	0.0290	0.002	12.570	0.000	0.024	0.033
ans_qns_ratio	0.0241	0.007	3.387	0.001	0.010	0.038

Omnibus:	150548.758	Durbin-Watson:	1.868
Prob(Omnibus):	0.000	Jarque-Bera (JB):	26957531541.494
Skew:	49.732	Prob(JB):	0.00
Kurtosis:	3675.332	Cond. No.	92.1

9.1.5 Number of Watchers

9.1.5.1 Full Model

OLS Regression Results

Dep. Variable:	num_watchers	R-squared:	0.023
Model:	OLS	Adj. R-squared:	0.023
Method:	Least Squares	F-statistic:	81.88
Date:	Sun, 31 Mar 2019	Prob (F-statistic):	4.37e-233
Time:	16:09:03	Log-Likelihood:	-66066.
No. Observations:	47939	AIC:	1.322e+05
Df Residuals:	47925	BIC:	1.323e+05
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
reputation	-0.0252	0.010	-2.621	0.009	-0.044	-0.006
view	0.0325	0.009	3.648	0.000	0.015	0.050
upvotes	-0.0144	0.003	-5.417	0.000	-0.020	-0.009
downvotes	-0.0207	0.006	-3.555	0.000	-0.032	-0.009
num_qns	-0.0381	0.005	-8.148	0.000	-0.047	-0.029
num_ans	-0.0439	0.012	-3.686	0.000	-0.067	-0.021
num_badges	0.0706	0.008	8.912	0.000	0.055	0.086
num_accept	0.0341	0.014	2.416	0.016	0.006	0.062
accept_ratio	0.0024	0.000	9.479	0.000	0.002	0.003
qns_pca	-0.0032	0.004	-0.913	0.361	-0.010	0.004
ans_pca	0.0212	0.002	8.884	0.000	0.017	0.026
pagerank	0.0029	0.011	0.275	0.783	-0.018	0.024
num_areas	-5.842e-05	0.001	-0.076	0.939	-0.002	0.001
ans_qns_ratio	0.0255	0.007	3.573	0.000	0.012	0.039

Omnibus:	140824.771	Durbin-Watson:	1.893
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18241547321.306
Skew:	41.205	Prob(JB):	0.00
Kurtosis:	3023.859	Cond. No.	103.

9.1.5.2 Model with Pure Reputation-Linked Features

OLS Regression Results

Dep. Variable:	num_watchers	R-squared:	0.015
Model:	OLS	Adj. R-squared:	0.015
Method:	Least Squares	F-statistic:	92.69
Date:	Mon, 01 Apr 2019	Prob (F-statistic):	1.35e-153
Time:	21:06:43	Log-Likelihood:	-66265.
No. Observations:	47939	AIC:	1.325e+05
Df Residuals:	47931	BIC:	1.326e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
reputation	-0.0847	0.008	-10.085	0.000	-0.101	-0.068
view	0.0182	0.009	2.070	0.038	0.001	0.035
upvotes	-0.0163	0.003	-6.111	0.000	-0.021	-0.011
downvotes	-0.0206	0.006	-3.528	0.000	-0.032	-0.009
num_qns	-0.0593	0.005	-13.091	0.000	-0.068	-0.050
num_ans	-0.0407	0.010	-3.875	0.000	-0.061	-0.020
num_badges	0.1331	0.006	20.935	0.000	0.121	0.146
num_accept	0.0553	0.012	4.593	0.000	0.032	0.079

Omnibus:	140720.518	Durbin-Watson:	1.881
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18167305873.027
Skew:	41.121	Prob(JB):	0.00
Kurtosis:	3017.705	Cond. No.	13.9

9.1.5.2 Reduced Model

OLS Regression Results

Dep. Variable:	num_watchers	R-squared:	0.023
Model:	OLS	Adj. R-squared:	0.023
Method:	Least Squares	F-statistic:	113.7
Date:	Sun, 31 Mar 2019	Prob (F-statistic):	3.72e-235
Time:	12:02:52	Log-Likelihood:	-66071.
No. Observations:	47939	AIC:	1.322e+05
Df Residuals:	47929	BIC:	1.322e+05
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
view	0.0267	0.009	3.139	0.002	0.010	0.043
upvotes	-0.0134	0.003	-5.101	0.000	-0.019	-0.008
downvotes	-0.0202	0.006	-3.472	0.001	-0.032	-0.009
num_qns	-0.0308	0.004	-7.748	0.000	-0.039	-0.023
num_ans	-0.0475	0.010	-4.565	0.000	-0.068	-0.027
num_badges	0.0553	0.005	10.619	0.000	0.045	0.066
num_accept	0.0324	0.012	2.598	0.009	0.008	0.057
accept_ratio	0.0025	0.000	10.331	0.000	0.002	0.003
ans_pca	0.0220	0.002	9.732	0.000	0.018	0.026
ans_qns_ratio	0.0272	0.007	3.883	0.000	0.013	0.041

Omnibus:	140808.179	Durbin-Watson:	1.893
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18215721324.819
Skew:	41.192	Prob(JB):	0.00
Kurtosis:	3021.719	Cond. No.	92.1

9.1.6 Coding Capabilities

9.1.6.1 Full Model

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              expertise   R-squared:                       0.298
Model:                            OLS   Adj. R-squared:                  0.298
Method:                 Least Squares   F-statistic:                     1451.
Date:                Sun, 31 Mar 2019   Prob (F-statistic):               0.00
Time:                        16:09:13   Log-Likelihood:                -36348.
No. Observations:               47939   AIC:                         7.272e+04
Df Residuals:                   47925   BIC:                         7.285e+04
Df Model:                          14
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
reputation     0.0072      0.005      1.386      0.166      -0.003       0.017
view           0.0020      0.005      0.418      0.676      -0.007       0.011
upvotes        0.0009      0.001      0.612      0.540      -0.002       0.004
downvotes     -0.0124      0.003     -3.982      0.000      -0.019      -0.006
num_qns       -0.0163      0.003     -6.466      0.000      -0.021      -0.011
num_ans       -0.0501      0.006     -7.814      0.000      -0.063      -0.038
num_badges     0.0313      0.004      7.342      0.000       0.023       0.040
num_accept     0.0122      0.008      1.604      0.109      -0.003       0.027
accept_ratio   0.0038      0.000     28.426      0.000       0.004       0.004
qns_pca       -0.0006      0.002     -0.292      0.771      -0.004       0.003
ans_pca        0.0947      0.001     73.768      0.000       0.092       0.097
pagerank       0.0454      0.006      7.974      0.000       0.034       0.057
num_areas     -0.0007      0.000     -1.576      0.115      -0.001       0.000
ans_qns_ratio  0.0039      0.004      1.012      0.312      -0.004       0.011
==============================================================================
Omnibus:                    17119.111   Durbin-Watson:                   0.157
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            93811.571
Skew:                           1.630   Prob(JB):                         0.00
Kurtosis:                       9.028   Cond. No.                         103.
==============================================================================
```

9.1.6.2 Model with Pure Reputation-Linked Features

OLS Regression Results

Dep. Variable:	expertise	R-squared:	0.114
Model:	OLS	Adj. R-squared:	0.114
Method:	Least Squares	F-statistic:	771.7
Date:	Mon, 01 Apr 2019	Prob (F-statistic):	0.00
Time:	21:06:44	Log-Likelihood:	-41916.
No. Observations:	47939	AIC:	8.385e+04
Df Residuals:	47931	BIC:	8.392e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
reputation	-0.1721	0.005	-34.047	0.000	-0.182	-0.162
view	-0.0349	0.005	-6.615	0.000	-0.045	-0.025
upvotes	-0.0041	0.002	-2.555	0.011	-0.007	-0.001
downvotes	-0.0118	0.004	-3.368	0.001	-0.019	-0.005
num_qns	-0.0767	0.003	-28.133	0.000	-0.082	-0.071
num_ans	-0.0089	0.006	-1.402	0.161	-0.021	0.004
num_badges	0.2145	0.004	56.060	0.000	0.207	0.222
num_accept	0.0400	0.007	5.519	0.000	0.026	0.054

Omnibus:	17848.744	Durbin-Watson:	0.107
Prob(Omnibus):	0.000	Jarque-Bera (JB):	68745.385
Skew:	1.859	Prob(JB):	0.00
Kurtosis:	7.539	Cond. No.	13.9

9.1.6.2 Reduced Model

OLS Regression Results

Dep. Variable:	expertise	R-squared:	0.298
Model:	OLS	Adj. R-squared:	0.297
Method:	Least Squares	F-statistic:	2900.
Date:	Sun, 31 Mar 2019	Prob (F-statistic):	0.00
Time:	11:18:36	Log-Likelihood:	-2.0446e+05
No. Observations:	47939	AIC:	4.089e+05
Df Residuals:	47932	BIC:	4.090e+05
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
downvotes	-0.3655	0.094	-3.891	0.000	-0.550	-0.181
num_qns	-0.6611	0.071	-9.349	0.000	-0.800	-0.523
num_ans	-1.3192	0.139	-9.468	0.000	-1.592	-1.046
num_badges	1.1569	0.088	13.124	0.000	0.984	1.330
accept_ratio	0.1259	0.004	29.316	0.000	0.117	0.134
ans_pca	3.1104	0.040	77.367	0.000	3.032	3.189
pagerank	1.5995	0.162	9.861	0.000	1.282	1.917

Omnibus:	17125.108	Durbin-Watson:	0.151
Prob(Omnibus):	0.000	Jarque-Bera (JB):	90337.190
Skew:	1.645	Prob(JB):	0.00
Kurtosis:	8.865	Cond. No.	69.6