Nina Kaake, Harrison Collins, and Jung Hoon Seo
24 November 2020

Out-of-Bounds: Stack Overflow Posts Analysis

Overview and Motivation:

Stack Overflow is used by programmers all over the world to ask and answer questions about a variety of concepts in software development. What makes Stack Overflows' platform unique though is its strict community guidelines. Users are expected to contribute quality questions and answers. However, not all users adhere to Stack Overflows' community guidelines. What if there was a way to predict the types of questions that will generate good quality, accepted answers on Stack Overflow? Our aim is to determine the features that contribute to a Stack Overflow post generating a high score and good quality, accepted answers. Our findings will allow users on Stack Overflow to better structure their questions to increase the likelihood of receiving better quality responses and recognition on Stack Overflow.

Related Work:

When thinking about a topic to explore for this project, we stumbled upon a Kaggle page that referred us to the BigQuery dataset that contains almost 13 million Stack Overflow posts. We thought looking into Stack Overflow may be interesting, but we weren't sure what questions to ask about this data set. As a result, we decided to do some research on what others have studied. When we came across a research article titled, "An Insight into the Unresolved Questions at Stack Overflow", we thought it would be interesting to look at what features can predict the quality of a question.

Initial Questions:

When we started our data analysis, our initial question was the following: What features created by Stack Overflow contribute to the success of an accepted answer? However, after performing a lasso regression on all of our features, our question later evolved into the following: What features created by Stack Overflow contribute to the success of an accepted answer or question? The difference now is that we are analyzing the score of a question and accepted answer. We used features such as the number of views of a post, the score of a question or answer, and the reputation of the author of an accepted answer to answer this question. However, after using models on the existing features in our dataset, we received low scores for our regression and classification models.

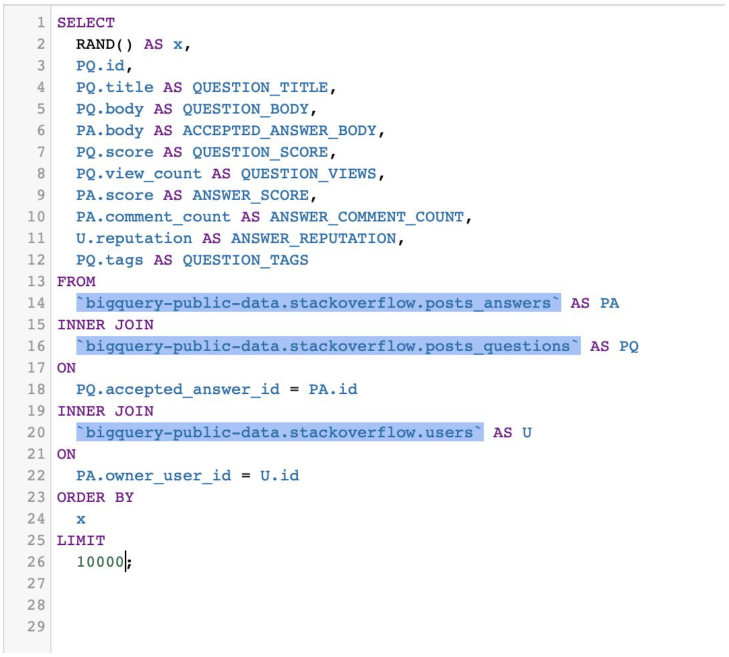
As a result, we wondered if we could make some features based on the text Therefore, our second question became: What textual features in questions get better answers and more upvotes on Stack Overflow? For example, we considered the Flesch Reading Score, the punctuation and the presence of code in posts. Some of these new variables helped us to answer our original questions. Later, we questioned if there are distinct groups of posts we should investigate so we clustered the text data to find out.

Data:

Our data source for this project was Google's BigQuery, which is described as an "enterprise data warehouse". Firstly, we needed to understand what tables and columns were available to us. Once we decided which tables and columns we wanted to use, we wrote an SQL expression to gather a random sample of 10,000 Stack Overflow posts.

Since our data was over 10,000 rows, we needed to export it to Google Cloud Storage (GCS) and then download it locally.

Once we had our data, we realized that we had some missing values and that the textual data was in HTML format. Our first order of business was to remove any HTML tags from our code and fill in missing values with either 0 or a blank string.



```sql
SELECT
  RAND() AS x,
  PQ.id,
  PQ.title AS QUESTION_TITLE,
  PQ.body AS QUESTION_BODY,
  PA.body AS ACCEPTED_ANSWER_BODY,
  PQ.score AS QUESTION_SCORE,
  PQ.view_count AS QUESTION_VIEWS,
  PA.score AS ANSWER_SCORE,
  PA.comment_count AS ANSWER_COMMENT_COUNT,
  U.reputation AS ANSWER_REPUTATION,
  PQ.tags AS QUESTION_TAGS
FROM
  `bigquery-public-data.stackoverflow.posts_answers` AS PA
INNER JOIN
  `bigquery-public-data.stackoverflow.posts_questions` AS PQ
ON
  PQ.accepted_answer_id = PA.id
INNER JOIN
  `bigquery-public-data.stackoverflow.users` AS U
ON
  PA.owner_user_id = U.id
ORDER BY
  x
LIMIT
  10000;
```

Nina Kaake, Harrison Collins, and Jung Hoon Seo
24 November 2020

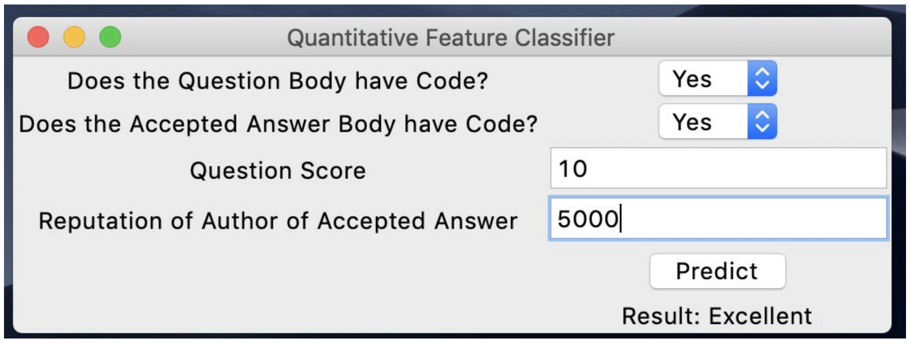
Exploratory Data Analysis:

To start exploring the quantitative data, we first used seaborn's pair plots to plot the different features against each other. Since the other plots in the pair plot were quite scattered, we decided to do a bivariable analysis of the data. For example, we created a bar chart to analyze the % of questions with a question mark at the end vs. the question quality. We used bar charts because they are a good visualization tool to use when plotting quantitative against ordinal data.

In addition, we used word clouds to see what the most frequently used words were in excellent quality and poor quality Stack Overflow questions. To do this, we used a tf-idf vector and tuned the vector to removed any English stop words and have a minimum document frequency of 0.08. Since the data is quite sparse, we were only able to identify significant words when we analyzed poor vs. excellent quality questions. The word clouds for fair, good, and very good did not show any significant word pairings. Although this could only give us an idea of the word frequencies in different kinds of posts, it was a good starting point to gain insight into our textual data.

Next, we used lasso regression to analyze any significant correlations between certain features. Since we wanted to avoid overfitting, we used a 10-fold cross-fold validation. In addition, to normalize the columns against each other, we used the StandardScaler function from sklearn. Lastly, we plotted the coefficient values using horizontal bar charts to find significant features that were produced from Lasso Regression.

From the lasso regression, we determined that the features that best predict the score of an answer are the score of a question, the presence of code in the question, the presence of code in the accepted answer, and the accepted answer's author's reputation. Using the features, we used a decision tree to predict the answer quality of a question using the features we selected from the Lasso Regression. To avoid overfitting, we used a 10-fold cross-fold validation and limited the depth of our decision tree to 13. We chose a depth of 13 since this generated a test accuracy score of 71% which is a fairly acceptable score for us.

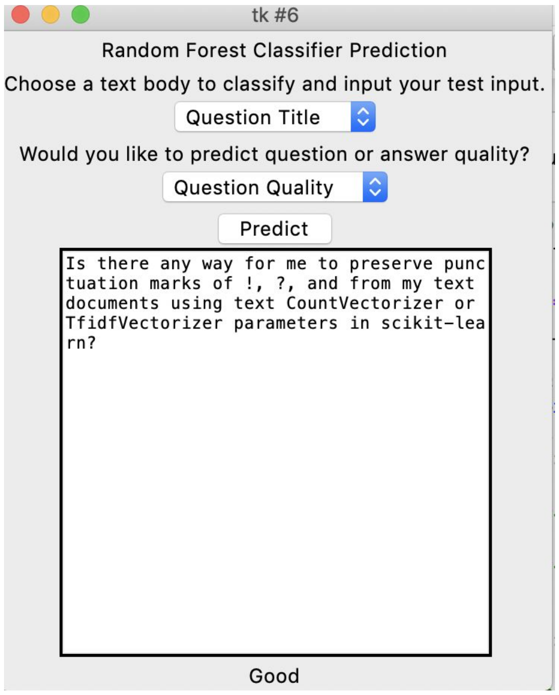

Quantitative Feature Classifier

Does the Question Body have Code?	Yes
Does the Accepted Answer Body have Code?	Yes
Question Score	10
Reputation of Author of Accepted Answer	5000

Result: Excellent

Nina Kaake, Harrison Collins, and Jung Hoon Seo
24 November 2020

Next, we used a random forest classifier to predict the question quality or answer quality of a Stack Overflow post based on the question body, question title, or accepted answer body. We used a tf-idf vector that did not use idf since the text data is very sparse. In addition, we set the max features to 2685 to use the top 1% of features (the total number of features was found to be 26850). Since we also wanted to consider the punctuation within the text body when predicting, we came up with a regular expression for the token pattern. In addition, to avoid overfitting for our random forest classifier, we used a 10-fold cross-fold validation, limited the depth of our trees to 16, and set our max features to the square root of the total number of features in our tf-idf vector. To increase the accuracy of our classifier, we decided to use 20 tree estimators, especially since we are using text data. In the end, we generated a 71% test accuracy score which is a fairly acceptable score for us. To visualize our classifier, we used a GUI so that users could better interact with our classifier and determine the quality of their Stack Overflow question or accepted answer based on their question body, title, or accepted answer body.


tk #6

Random Forest Classifier Prediction

Choose a text body to classify and input your test input.

Question Title

Would you like to predict question or answer quality?

Question Quality

Predict

```
Is there any way for me to preserve punc
tuation marks of !, ?, and from my text
documents using text CountVectorizer or
TfidfVectorizer parameters in scikit-lea
rn?
```

Good

Another question which interested us was whether there were distinct groups of text which could be viewed in clustering. To answer this, we implemented agglomerative clustering on the question title. We found that the results were best for the title rather than the body of the question or answer. PCA is used to reduce the dimensionality and view the clusters. To get an idea of each cluster, we found the most common words from each cluster.

Nina Kaake, Harrison Collins, and Jung Hoon Seo
24 November 2020

Final Analysis:

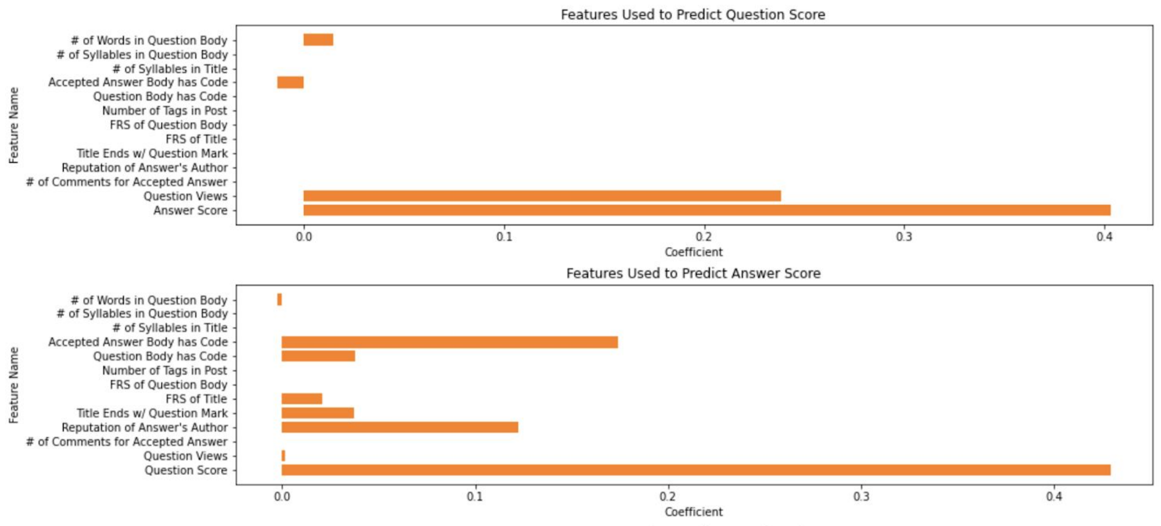
Bivariable Analysis:
Based on our bar charts from our bivariable analysis, we determined that the percentage of excellent quality questions that had a question mark at the end of a question's title was almost 3 times higher than that of poor quality questions. This shows that there is a relationship between having appropriate punctuation in a question title and the quality of a question. This makes sense since grammatically correct text is generally regarded positively. We also found that the percentage of excellent quality answers that had a question mark at the end of a question's title was almost twice that of poor quality answers. This tells us that ending a question title with a question mark may increase the likelihood of receiving a better quality answer. This makes sense since people tend to mimic the formality and style of the person they are conversing with.

Another interesting finding was that the Flesch Reading Score (FRS) of a question title and question body is inversely related to the quality of a question. In other words, higher quality questions have a low FRS score (i.e. complex readability) and low-quality questions have a high FRS score (i.e. easily understandable text). This may be because higher quality questions contain more technical language. Using more technical language in a question's title and body may incline users to upvote the post because the author of the post may be deemed as more credible and knowledgeable. We can justify this by looking at the "number of words in question body vs. question quality" and "number of syllables in question body vs. question quality" bar charts. These charts show that higher quality questions tend to have question bodies that contain more words and syllables, on average, compared to lower quality questions. FRS is calculated using the following equation: Reading Ease = 206.835 – (1.015 x Average Sentence Length) – (84.6 x Average number of Syllables/ Word). From this equation and the insights from these bar graphs, it makes sense that higher quality questions would have a lower FRS score since they tend to have longer sentences and more syllables. Also, question bodies that contain more words may be regarded as more descriptive which increases their quality. However, there was no relationship between the accepted answer quality and FRS score.

In addition, as suspected, higher quality questions tend to have more tags, on average, compared to lower quality questions. This makes sense since adding more tags will increase the visibility of a question.

Nina Kaake, Harrison Collins, and Jung Hoon Seo
24 November 2020

Lasso Regression Analysis:
From our initial lasso regression, we learned that the data is incredibly sparse and couldn't generate that many "good" correlations. Therefore, we decided to gather more variables and create new variables. Despite gathering more variables, our coefficients were still quite low. However, we realized that certain features have some correlation with answer scores and question scores. Namely, the question score, the presence of code in the question, the presence of code in the answer, and the accepted answer's author's reputation are good predictors of answer quality. In addition, question views, answer score, and number of words in the question body are good predictors of question quality. See the graph below for a visualization of our lasso regression findings for question and answer score.



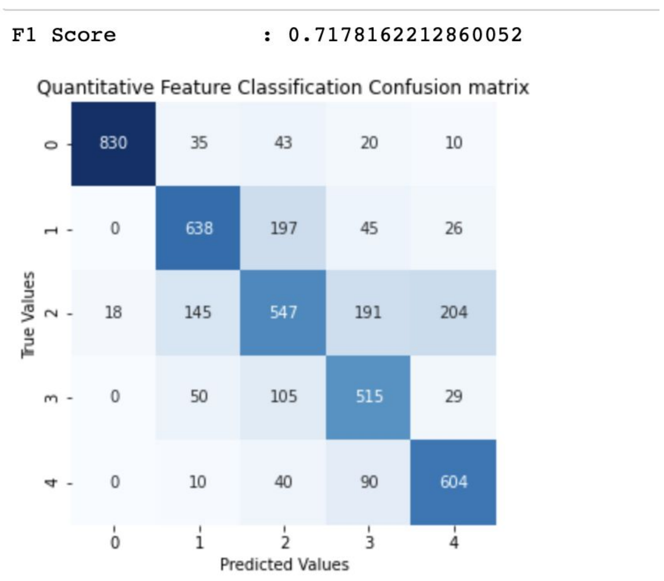
Text Visualization Analysis:
To visualize the most frequently used words in excellent quality and poor quality Stack Overflow questions, we created word clouds. It was quite interesting to us that the most found word across poor quality questions is the word "error". This tells us that a question may be considered poor if an asker is only concerned with an error in their code. On the other hand, the most found word across excellent quality questions was the word "change". Questions that contain the word "change" may be positively regarded since the asker seems to have made a genuine attempt to solve their problem by attempting different alternatives.

Nina Kaake, Harrison Collins, and Jung Hoon Seo
24 November 2020

Text Classification Analysis:
Without tuning our Random Forest Classifier, the classifier outputted an accuracy score of around 97% which told us that the classifier was significantly overfitting the data since the model was considering almost 30,000 features. As a result, we tuned the classifier and decreased the accuracy score to a more reasonable value of 70%. If the accuracy score was 20%, then we would conclude that the classifier was randomly classifying the text data. However, the accuracy score tells us that although around ¼ of the predictions will be incorrect, the classifier is not just randomly predicting values.

Quantitative Classification Analysis:
Our decision tree classification showed us that it is easier to predict the success of an answer using these numerical features rather than the text itself. However, both classifiers are significantly better than randomly guessing since their accuracy scores are around 70% rather than 20%. Both of these models were evaluated by scikit-learn's respective score functions.

We also printed a confusion matrix to check how many predicted values match with the real values, and then we checked our approximation by printing the F-1 score. For example, from this matrix we can see that our model predicted 90 posts as having a quality of "Very Good" when they were supposed to be "Excellent".



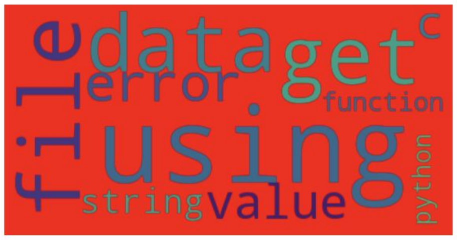
```
F1 Score           : 0.7178162212860052
```

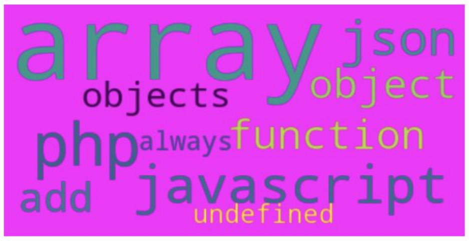
	0	1	2	3	4
0	830	35	43	20	10
1	0	638	197	45	26
2	18	145	547	191	204
3	0	50	105	515	29
4	0	10	40	90	604

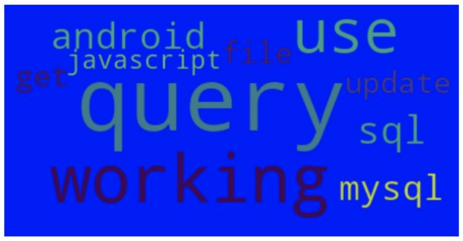
Nina Kaake, Harrison Collins, and Jung Hoon Seo
24 November 2020

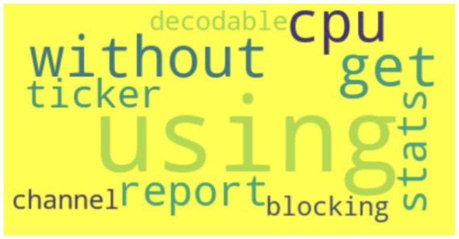
Agglomerative Clustering Analysis:
The clusters we obtained from agglomerative clustering were not clearly defined. However, The clusters held roughly different locations in the two dimensional graph obtained by PCA. By viewing the top ten words from each of the clusters, we attempted to understand if the clustered titles contained distinct themes. We observed the following:

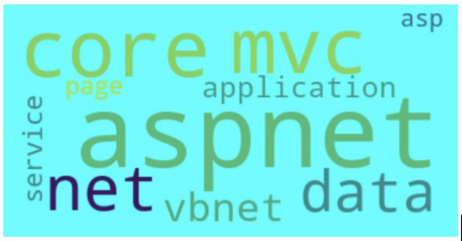
1) The first (red) cluster is related to obtaining data from a file.
2) The second (blue) uses words common when working with databases.
3) The third (teal) has a strong relation to web development.
4) The fourth (yellow) cluster uses words related in outputs and statistics, but also contains some unrelated noise.
5) The fifth (magenta) cluster mentions javascript and terms related to it.


file
data
get
c
error
function
using
python
string
value


array
json
objects
object
php
always
function
add
javascript
undefined


android
use
javascript
file
get
query
update
sql
working
mysql


decodable
cpu
without
get
ticker
using
stats
channel
report
blocking


core
mvc
asp
page
application
service
aspnet
net
vbnet
data