# Project 2: Stack Exchange

**Executive Summary:**

We are tasked with analyzing a relational dataset containing 50,000 unique observations of answered and unanswered questions from Stat Exchange, a forum for posting statistics related questions. Each observation contains information on the post such as the user's ID, the post title, the tags associated with the post, the main body of the post, number of comments, and more. As data science students who may want to post a single question to Stack Exchange in the future, we want to create a model that will maximize the likelihood that our questions will be answered to our expectations by another user of the forum.

This report describes some of the observed trends within the dataset, illustrates our methodology for building a model that predicts whether a post will be answered with an "Accepted Answer", and provides recommendations on how to interpret each of the features in order to instruct future post submissions on maximizing the likelihood of getting an accepted response. We compare this model to both the null and saturated models for sufficiency and goodness of fit tests, and we have determined that this model surpasses the null model's performance.

**Introduction to Problem Statement:**

Stack Overflow is an internet forum founded in 2008 by Jeff Atwood and Joel Spolsky. The intention surrounding Stack Overflow was to create an internet forum for people to discuss programming-related topics. Stack Overflow differentiated itself from other online programming forums in its community's ability to moderate content by flagging questions that were subjective in nature, reducing the chances of long-winded, opinion-based threads. Stack Overflows model quickly took off, and by 2009 Atwood and Spolsky had created a network, named Stack Exchange, that connected variety of communities on topics spanning from science to literature. As of the time of this report, over 175 communities exist on Stack Exchange. One of these communities, Cross Validated[1], focused on the statistics-related topics.

Given this background of how sites like Stat Exchange value specific and objective questions, and given our desire to learn and grow as students, we hypothesized that we could build a model that would predict the likelihood of a question being answered on Cross Validated based on features that would measure the specificity of a given question. If we could build a model that informed us on how likely a question would be answered, we could alter the phrasing of our question and formation of our posts to maximize the likelihood that our question would be

[1] Cross Validated (stats.stackexchange)

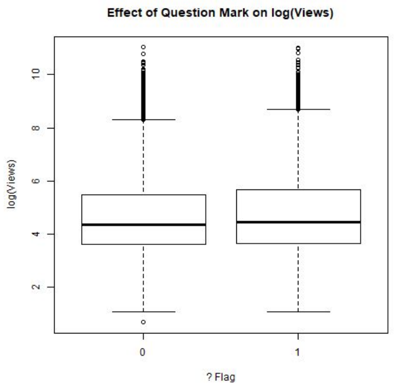
answered sufficiently. In short, **what features matter most in ensuring questions get answered on Cross Validated, and how does altering these features impact the odds of getting a question answered?**

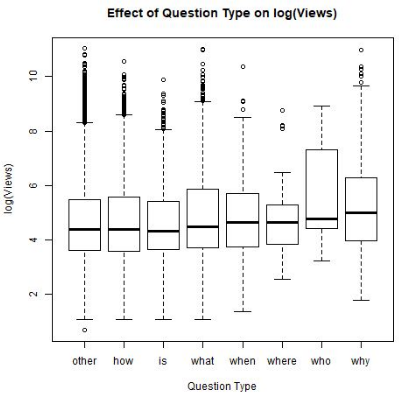
**Description of Data:**

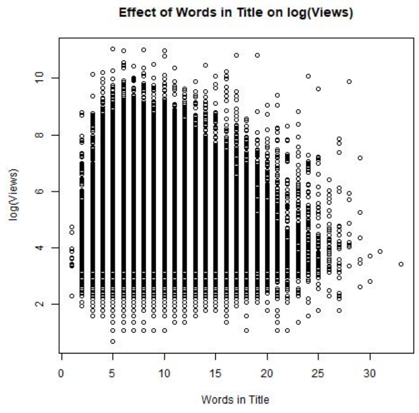
This dataset was exported from Stack Exchanges[2] query engine, which operates using a SQL-like language to pull data from a relational data model. The data model contains information on Posts, Users, Badges, and more. For our goal of creating targeted questions that we hope would be answered, we had to limit the scope of our data to information that would we could control before submitting a post. The information that we had to inform our model included the posts title, the body of the post, tags associated with the post, and time of day submitted. Additionally, we considered all posts from January 1, 2017 in order to obtain the most relevant data for our model to be used in present day.

Post Title

Arguably the most important part of any submission to Stack Exchange is the title, for it is the first impression that other users get of your submission. We wanted to capture the conciseness and pointedness of the title using a few features: title word length, a flag if the title contained a question mark, and finally the type of question being asked (what vs. when vs. how). In order to visualize the impact that these features had on a post, we compared these features to the number of views the post would get as a proxy for whether or not the question was answered appropriately.







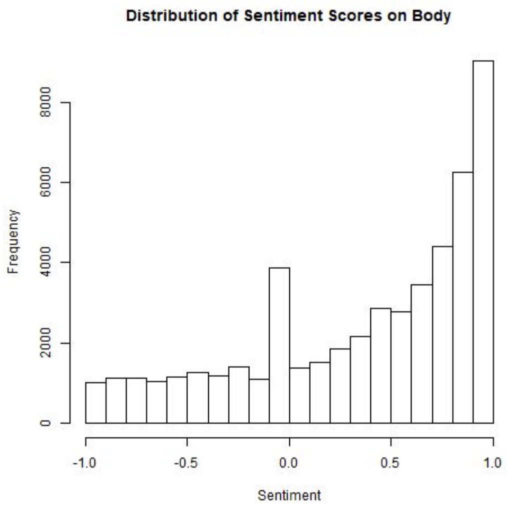
The figure on the left shows that the inclusion of a question mark slightly improve the number of views the post gets, suggesting that post titles with question marks are more targeted and specific because they have a question in mind. The center chart displays the different types of

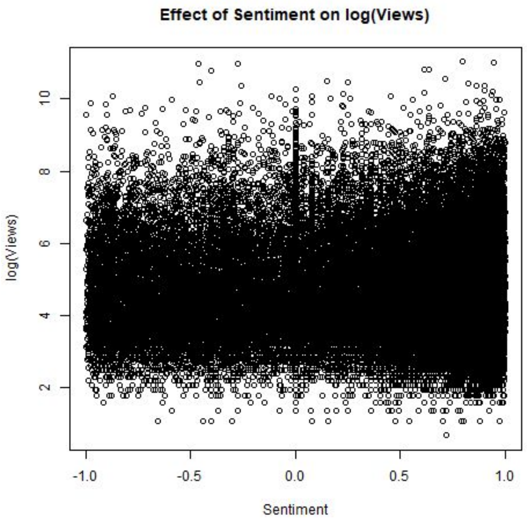
[2] Stack Exchange Data Explorer - https://data.stackexchange.com/stats/query/new

questions, which are categorized by which qualifying word they start with (who, what, when, where, why, how, is, and other if none of the above). This chart tells us that questions that start with "Who" or "Why" tend to get higher amounts of views compared to other types of questions. The figure on the right shows the effect of the number of words in the title versus views. This chart tells us that there is generally an increase in views between the range of three words to ten words, and then a gradual decrease in views as the number of words increases beyond ten.

Post Body

The body of the submission contains the details of the question posed by the submitter. This could contain code, formulas, or theoretical questions that the submitter has a question about. This data is structured in a markup-like language, likely HTML. We wanted to measure the overall "sentiment" of each post, or in other words whether the general feeling of the post was positive or negative. To obtain this type of information, we leveraged an already-trained sentiment analysis algorithm called VADER[3]. This tool measures a set of text's overall sentiment between negative one and one, with negative one being entirely negative and one being entirely positive. We applied the VADER model to the body of each post, and obtained the following results:

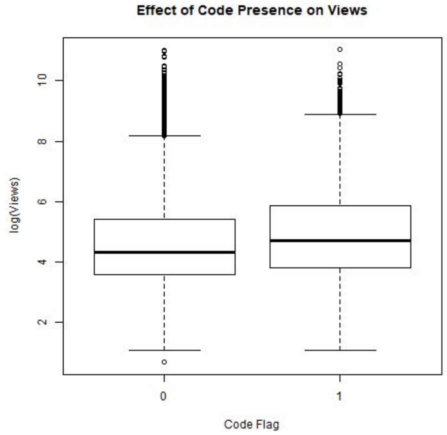


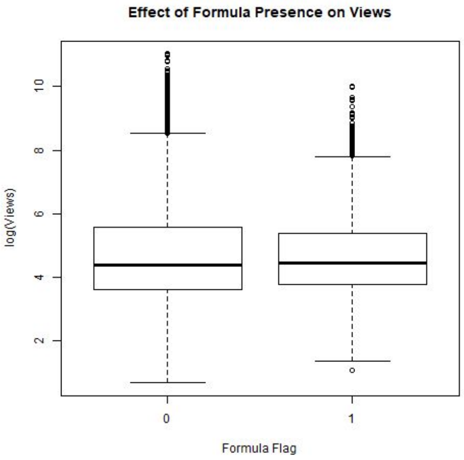


The chart on the left demonstrates that the overall sentiment of this dataset is highly skewed towards positive posts. There is also a spike around a sentiment score of zero, which can be attributed to the VADER model's incapability to extract sentiment from special cases, such as just code and formulas provided. The chart on the right shows that sentiment score does not have a strong effect on the number of views a post receives.

[3] VADER Sentiment (github link)

Additionally, we considered whether or not the body included code samples and formulas, with the thought that providing more concrete examples would improve the odds that a question would be sufficiently answered.





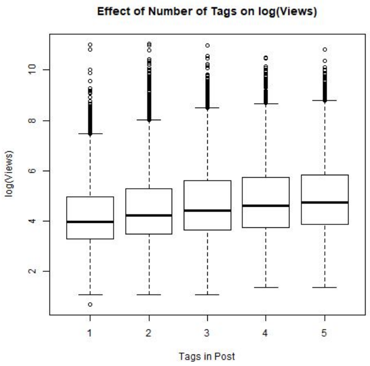
The figure on the left suggests that the presence of code snippets has some impact on the number of views the post receives. The figure on the right does not suggest that the presence of formulas leads to more views.

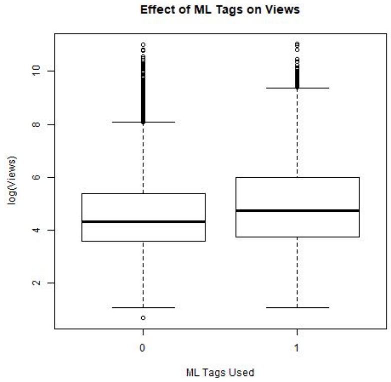
## Tags

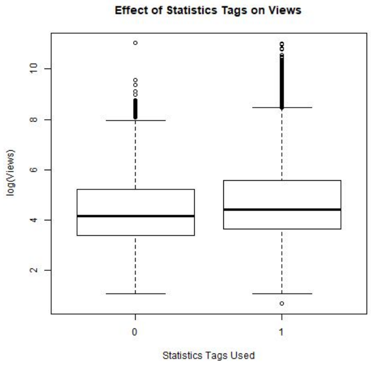
Tags are additional attributes assigned to a post that help categorize the submission based on the submissions content. A post can be assigned with multiple tags (up to 5), and could also have no tags. We wanted to determine if certain subject matters would garner more attention than others. Here, we looked at two groups of tags: those that traditionally belong to machine learning techniques (neural-networks, boosting, etc), and those to belong to more traditional statistical studies (regression, hypothesis tests, etc). In total, Cross Validated has supported approximately 1350 tags since 2017. The groups of tags provided below were grouped based on their frequency of appearance in our dataset and on the general connotations surrounding these terms.

**Machine Learning Tags:** machine-learning, neural-network, clustering predictive-models, deep-learning, classification, boosting, svm, feature-selection, random-forest, python, conv-neural-network , scikit-learn, k-means, unbalanced-classes

**Statistics Tags:** r, regression, time-series, hypothesis-testing, bayesian, correlation, statistical-significance, mathematical-statistics, normal-distribution, anova, multiple-regression, mixed-model, confidence-interval, t-test, generalized-linear-model



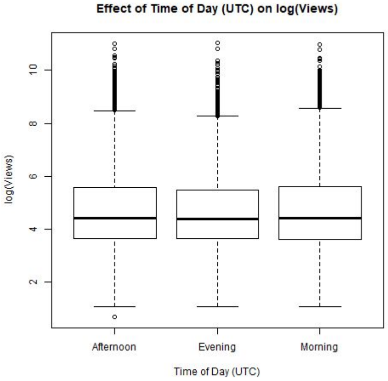


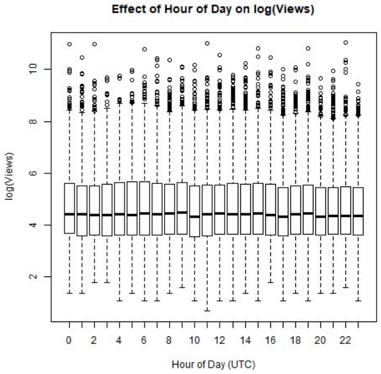


The figure on the left demonstrates that the addition of tags has a strong effect on the number of views a post receives. Additionally, the presence of the popular Machine Learning tags also drives views, but this could be confounded by the number of tags these posts have compared to the posts that do not have ML tags. Finally, the presence of traditional statistics tags has a small influence on the number of views received by a submission.

Time of Day

Time of day could be a controlled factor when trying to optimize the visibility of a post on Cross Validated. For Americans, one would expect that a post submitted at 3 a.m. would garner less attention than a post submitted at 8 p.m. For the moment, let's assume that the majority of the user-base for Cross Validated lies on the east coast. In this case, posts submitted at 3 a.m. would receive very little attention, and would be pushed down the queue by posts submitted later in the morning. By the time that most users would start to browse Cross Validated, the post in question would likely be bumped off the main page of the site.





Both the left and right figures do not demonstrate strong evidence that the time of day the post is submitted has a strong effect on the number of views it receives.

**Analytic Approach:**

Our model attempted to predict where the post had an "Accepted Answer" for the post, regardless of the time it took for the answer to be provided. This response value takes upon a value of zero when no answer is accepted by the user, and a value of one when an answer is accepted by the user.

The features we tested for our set of models are as follows:

- TitleWordCount : number of words in a title
- TitleQMark: 1 if title contains "?", 0 if not
- QType: categorical variable for type of question posed in Title. Possible values are:
    - Who, what, when, where, why, how, is, other (if none of the above)
    - Determined by the first word of the post's title
- TagCount : number of tags in a post
- MLTagFlag : 1 if post contains common ML tags, 0 if none
- StatTagFlag: 1 if post contains common statistics tags, 0 if none
- Compound: overall sentiment of body of submission according to VADER model, scored between -1 (negative) and 1 (positive) where 0 is neutral (and default)
- HourOfDay: Value between 0 and 23 representing the hour of day (UTC) that the post was created
- FormulaFlag: 1 if post's body contained a formula (indicated by $$), 0 if not
- CodeFlag: 1 if the post's cody contains a code tag (indicated by <code>), 0 if not

Below are the set of models we generated, and tested their performance by measuring their Akaike Information Criterion (AIC) as well as their Area Under Curve (AUC) when plotted on a ROC plot (Receiver Operating Characteristic). We randomly split our dataset of 50,000 posts into a 45,000 post training set and a 5,000 post test set. The results below are the models that were trained on the training set.

| # | **Formula** | **AIC** | **AUC** |
|---|---|---|---|
| 1 | $logit(\pi) = \beta_0 + \beta_1 x_{TagCount} + \beta_2 x_{TitleWordCount} + \varepsilon$ | 53354.5 | 0.505695 |
| 2 | $logit(\pi) = \beta_0 + \beta_1 x_{TagCount} + \beta_2 x_{TitleWordCount} + \beta_3 x_{MLTagFlag} + \varepsilon$ | 53353.3 | 0.505644 |

| | | | |
|---|---|---|---|
| 3 | $logit(\pi) = \beta_0 + \beta_1 x_{TagCount} + \beta_2 x_{TitleWordCount} + \beta_3 x_{TitleQMark} + \varepsilon$ | 53314.5 | 0.505463 |
| 4 | $logit(\pi) = \beta_0 + \beta_1 x_{TagCount} + \beta_2 x_{TitleQMark} + \varepsilon$ | 53312.5 | 0.505857 |
| 5 | $logit(\pi) = \beta_0 + \beta_1 x_{TagCount} + \beta_2 x_{TitleQMark} + \beta_3 x_{MLTagFlag} + \varepsilon$ | 53310.0 | 0.505535 |
| 6 | $logit(\pi) = \beta_0 + \beta_1 x_{TagCount} + \beta_2 x_{TitleQMark} + \beta_3 x_{MLTagFlag} + \beta_{4-10} x_{QType} + \varepsilon$ | 53176.2 | 0.508457 |
| 7 | $logit(\pi) = \beta_0 + \beta_1 x_{TagCount} + \beta_2 x_{TitleQMark} + \beta_3 x_{MLTagFlag} + \beta_{4-10} x_{QType} + \beta_{11} x_{HourOfDay} + \varepsilon$ | 53157.9 | 0.508484 |
| 8 | $logit(\pi) = \beta_0 + \beta_1 x_{TagCount} + \beta_2 x_{TitleQMark} + \beta_3 x_{MLTagFlag} + \beta_{4-10} x_{QType} + \beta_{11} x_{HourOfDay} + \beta_{11} x_{VADER} + \varepsilon$ | 53131.8 | 0.508626 |
| **9** | $logit(\pi) = \beta_0 + \beta_1 x_{TagCount} + \beta_2 x_{TitleQMark} + \beta_3 x_{MLTagFlag} + \beta_{4-10} x_{QType} + \beta_{11} x_{HourOfDay} + \beta_{11} x_{VADER} + \beta_{11} x_{CodeFlag} + \varepsilon$ | 53089.9 | 0.508677 |
| **10** | $logit(\pi) = \beta_0 + \beta_1 x_{TagCount} + \beta_2 x_{TitleQMark} + \beta_3 x_{FormulaFlag} + \beta_{4-10} x_{QType} + \beta_{11} x_{HourOfDay} + \beta_{11} x_{VADER} + \beta_{11} x_{CodeFlag} + \varepsilon$ | **52894.9** | **0.509580** |

Model 10 proved superior when measuring each model's performance based on the AIC and AUC. Model 10 had the lowest AIC value as well as the highest AUC value, both of which are good indications that Model 10 is the best model that we generated for predicting the likelihood of an Accepted Answer. Let us compare this model against the null model as well as the saturated model to determine the sufficiency of our model.

Versus the Null Model:

| | Resid. Df | Resid. Dev | Df | Deviance | Pr(>Chi) |
|---|---|---|---|---|---|
| 1 | 44992 | 53368.75 | | | |
| 2 | 44979 | 52866.88 | 13 | 501.87 | 0.0000 |

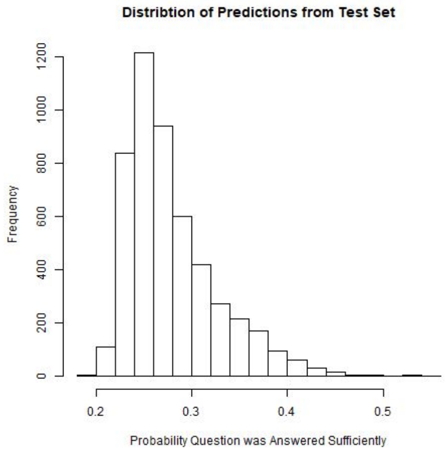
From the above Analysis of Variance (ANOVA) table, we reject the null hypothesis that the null model is the true model, where the null model is:

$$logit(\pi) = \beta_0$$

Now, let us compare Model 10 to the fully saturated model to determine goodness of fit. Our model's deviance is 53356.59 and has 44,991 degrees of freedom. The resulting p-value from calculating the Chi-squared test statistics is equivalent to zero, stating that we reject the null hypothesis that our model sufficiently explains the variance in the data relative to the fully saturated model. In summary, while our model does a significantly better job than the null model, we cannot say that it does a sufficient job compared to the saturated model.

**Results:**

In evaluating the accuracy of our final model, we first needed to determine a cutoff value that would tell us if our model predicted that the post was sufficiently answered.

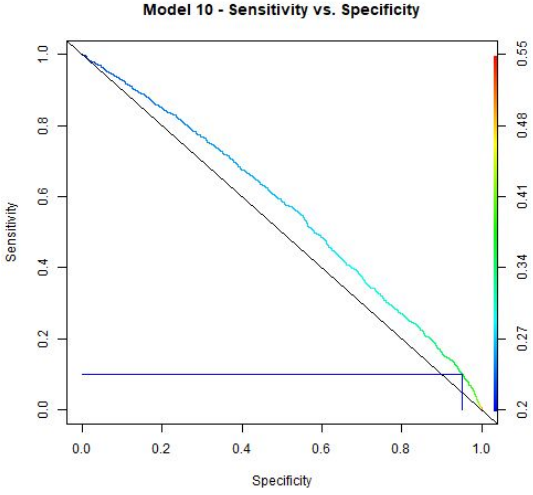


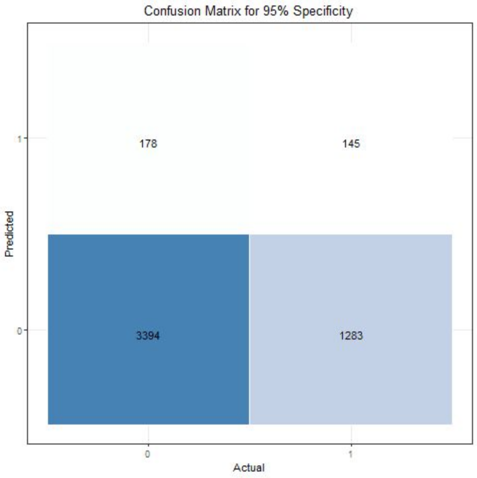
The above figure shows that the range of predictions our model makes regarding the probability a post was answered is approximately between 20% and 50%, with a heavy skew towards the lower end of the distribution. To determine an appropriate cutoff value, we need to consider a values that maximize the sensitivity or specificity of the model.:

$$sensitivity = \frac{True\ Positives}{True\ Positives + False\ Negatives},\ specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

For our purposes, we wanted to choose a cutoff value that provided higher specificity, since this would lead to higher precision rates. Keeping our goal in mind in maximizing the likelihood of our question getting sufficiently answered, we want our model to be as precise as possible so we are confident that when we go to post our question that the post will be answered. To demonstrate the relationship between sensitivity, specificity, and our cutoff value, we plotted our model's ROC curve [4]:

[4] Traditionally, ROC curves are plotted using False Positive Rates and True Positive rates, but in this case we wanted to demonstrate the tradeoff between Sensitivity and Specificity

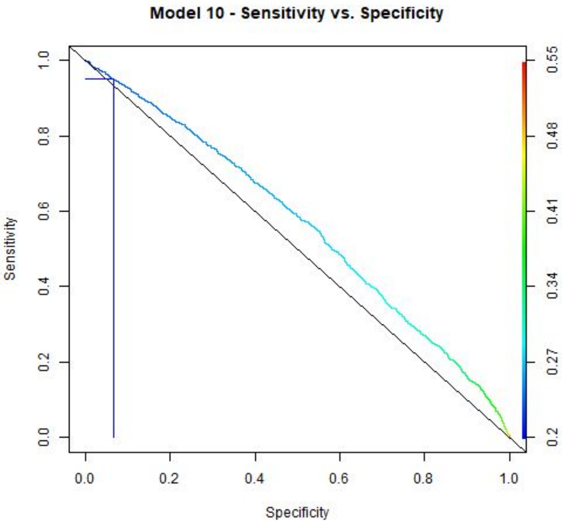


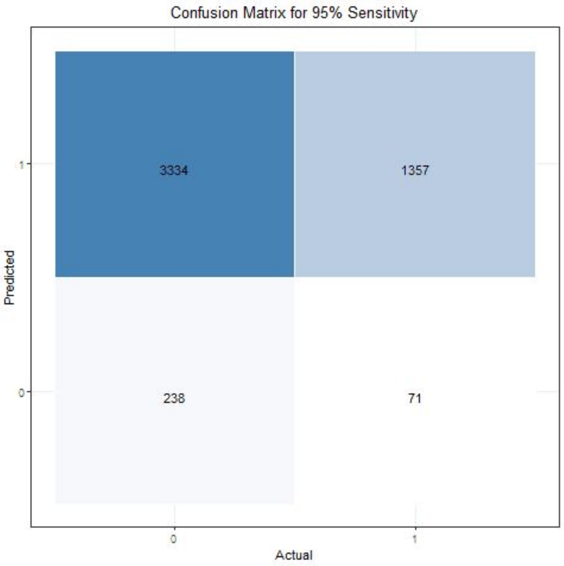

Confusion Matrix for 95% Specificity

| Predicted \ Actual | 0 | 1 |
|---|---|---|
| 1 | 178 | 145 |
| 0 | 3394 | 1283 |

In the left figure, we see that there is an expected negative relationship between our models sensitivity and specificity as we vary our cutoff value. At high cutoff values of >50%, our model's sensitivity incredibly low, but the specificity is very high, meaning we are identifying nearly all of the true negatives in the test set at the cost of missing the majority of the true positives. Conversely, at low cutoff values of <20%, we our model achieves incredibly high sensitivity at the cost of low specificity. The blue lines indicate the cutoff value where the specificity is at 95%, a value arbitrarily chosen to ensure that our model is stringent in predicting positive cases. **We chose to value specificity over sensitivity since we, as potential posters, only have one shot to pose our question on Cross Validated, and we want to ensure that we have done everything we can to increase the odds that our question gets answered**. At sensitivity level of 95%, our cutoff value for our predictions is approximately 36%. Using this cutoff value, we can categorize our predictions into boolean values of whether or not the question was sufficiently answers according to our models predictions. We can then compare these predictions to the actual test set's values and measure our model's accuracy.

In the right figure, we see of the 5000 posts in our test set our model only classified 323 posts to have an accepted answer. However, of these 323 posts, 45% of them were correctly classified as having accepted answers. Let's contrast this confusion matrix with one that has a sensitivity of 95%, and a cutoff value of 22.8%:





In the left figure, we set the desired sensitivity level to 95%, giving us a cutoff value of 22.7%. Applying this cutoff to our test set produces the confusion matrix on the right. With this low cutoff value, our model estimates that the majority of the posts have been answered, and only has a precision value of 29%. For our purposes, the lower precision rate as a result of the lowered cutoff value tells us that this cutoff value does much worse in informing us on whether our question will actually get answered or not.

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | -1.0520 | 0.0377 | -27.94 | 0.0000 |
| TagCount | 0.0227 | 0.0088 | 2.59 | 0.0096 |
| titleQMark1 | 0.1068 | 0.0264 | 4.04 | 0.0001 |
| QTypehow | -0.1508 | 0.0361 | -4.17 | 0.0000 |
| QTypeis | 0.2334 | 0.0563 | 4.14 | 0.0000 |
| QTypewhat | 0.2107 | 0.0469 | 4.49 | 0.0000 |
| QTypewhen | 0.0322 | 0.1257 | 0.26 | 0.7978 |
| QTypewhere | 0.4654 | 0.3045 | 1.53 | 0.1264 |
| QTypewho | 0.7323 | 0.6730 | 1.09 | 0.2766 |
| QTypewhy | 0.4928 | 0.0560 | 8.80 | 0.0000 |
| HoursContinuous | -0.0076 | 0.0017 | -4.54 | 0.0000 |
| compound | -0.1006 | 0.0188 | -5.35 | 0.0000 |
| code_flag1 | 0.2016 | 0.0244 | 8.26 | 0.0000 |
| formula_flag1 | 0.4624 | 0.0320 | 14.44 | 0.0000 |

Finally, we want to provide recommendations on how to tailor a post to improve the odds of your post getting answered. The most impactful things you could do to ensure your question gets sufficiently answered are including a formula in the body of your text and phrasing your question to start with “Why”. Including each of these will increase the odds of your question getting answered by 1.6 and 1.64 respectively. Additionally, including sample code and a question mark in your title increases the odds of your question getting answered by 1.2 and 1.1 respectively. Interestingly enough, there seems to be a negative relationship between sentiment of the post and the likelihood of one's question getting answered, suggesting that more negative posts tend to garner more adequate questions.

**Next Steps:**

There are some additional steps that we would have liked to have taken given more time to examine the problem. Primarily, we would have liked to have explored the use of Ridge, Lasso, and Elastic Net regression as a means of reducing the variance of our model at the cost of increasing the bias. Additionally, we would have liked to have performed cross-validation on each of our models to reduce the amount of potential sampling bias that could have been incurred by randomly splitting the data into training and test datasets. Finally, we would have liked to obtain User information at the time that the post was made. While User information is available on Stack Exchange's query engine, this data is only available for point-in-time, up-to-date information on Users, and did not reflect what the User's account looked like at the time of posting. Because of this, we felt we could not rely on this data to inform our regression model. However, if we had this information at our disposal, we likely could have explained more variance in the data, as users with higher engagement with the Cross Validated community would likely have a higher probability that they questions get answered.