Hooman Shiranimehr

# Final Report: Quora Insincere Questions Classification

**Problem Statement**

Any major website should be able to handle toxic and divisive content. One of such websites is Quora. On Quora, people can ask questions and connect with others who contribute unique insights and quality answers. A key challenge is to identify and remove insincere questions -- those founded upon false premises, or that intend to make a statement rather than look for helpful answers.

Our goal here is to develop models that identify and flag insincere questions. These models when optimized and combined with manual review by moderators can effectively address this problem.

In the following sections we first discuss data wrangling. Then we talk about Exploratory Data Analysis. After that, we investigate several models and compare their performance. Finally, we explain our conclusion and future work.

**Data Wrangling**

Data is obtained from <https://www.kaggle.com/c/quora-insincere-questions-classification>. The raw dataset contained 1,306,122 rows with 3 columns ‘qid’, ‘question\_text’ and ‘target’. ‘qid’ is the id for each question, ‘question\_text’ is the question text and ‘target’ is the question label (0 for sincere and 1 for insincere). Figure 1 and Figure 2 show sample data for sincere and insincere questions, respectively.

A screenshot of a cell phone

Description automatically generated

Figure 1: Sample data for sincere question.

A screenshot of a cell phone

Description automatically generated

Figure 2: Sample data for insincere questions.

We can easily verify that there is no Null or duplicate in the dataset. Therefore, no row needs to be removed. However, the dataset is too large and difficult to process. Hence, we sample data and keep only 20% of it. As we will see in the next section, data is not balanced as 96% of questions are sincere and only 4% are insincere. To make sure that we will have same distribution of labels after sampling, we use random stratified sampling. So, at the end, we have 261,224 rows of data with 3 columns.

**Exploratory Data Analysis**

As discussed earlier and illustrated in Figure 3, after sampling 20% of data labels are not balanced (96% sincere and 4% insincere).

A screenshot of a cell phone

Description automatically generated

Figure 3: Distribution of labels.

Next, we compare the two classes (sincere and insincere) using different parameters such as number of words, letters, punctuations and uppercase letters to understand if there is some difference between the two classes. Figure 4, Figure 5, Figure 6 and Figure 7 demonstrate this for the two classes.

A screenshot of a cell phone

Description automatically generated

Figure 4: Comparing two classes based on number of words.

A screenshot of a cell phone

Description automatically generated

Figure 5: Comparing two classes based on number of characters.

A picture containing sitting

Description automatically generated

Figure 6: Comparing two classes based on number of punctuations.

A screenshot of a cell phone

Description automatically generated

Figure 7: Comparing two classes based on number of uppercase letters.

From figures above, it seems that insincere questions may have more words and letters when compared to sincere questions. However, the difference is not too much and therefore, we cannot only rely on these parameters for classifying questions.

Finally, let’s create word cloud for the two classes to see if there is a clear difference between the two. Figure 8 and Figure 9 depict the word clouds for sincere and insincere questions, respectively. As it can be observed, there are some words that are used more frequently in insincere questions such as “Indian” and “Muslim”. We will further investigate this in the next section.

A close up of a piece of paper

Description automatically generated

Figure 8: Word cloud for sincere questions.

A picture containing table

Description automatically generated

Figure 9: Word cloud for insincere questions.

**Model Selection**

In this section, we consider 3 different models for our classification problem: Logistic regression, SVM and Naïve Bayes. We split the data as 75/25 for training/test. This means that we will have 195,918 questions for training and 65,306 for test.

As the first step, we study the performance of these 3 models without any hyper-parameter optimization. Each model is trained on the whole training set and then the performance is obtained for the test set. Figure 10, Figure 11 and Figure 12 show the results for each classifier. Note that, although accuracy value is high for all models, it is not a good metric to consider here as the classes are not balanced and even a simple model that classifies every question as sincere, could achieve 94% accuracy. On the other hand, confusion matrix is more useful as it can show mis-classification probabilities.

A screenshot of a cell phone

Description automatically generated

Figure 10: Accuracy and confusion matrix with logistic regression.

A screenshot of a cell phone

Description automatically generated

Figure 11: Accuracy and confusion matrix with SVM.

A screenshot of a cell phone

Description automatically generated

Figure 12: Accuracy and confusion matrix with naïve Bayes.

For the rest of this work, we focus on Precision-Recall curve as shown in Figure 13 and try to optimize each model by turning hyper-parameters.

A close up of a map

Description automatically generated

Figure 13: Precision-Recall curves.

In order to optimize hyper-parameters, we use 5-fold cross validation and perform grid search to find optimal set of parameters for each model. Then, we compare the models based on their average precision and choose the one that results in the highest average precision. That optimal model is then used to classify questions in the test data. Based on our results, the optimal model is logistic regression with parameters shows in Figure 14.

A close up of a newspaper

Description automatically generated

Figure : Optimal model.

Figure 15 and Figure 16 show ROC and Precision-Recall curves for optimal model, respectively.

A close up of a map

Description automatically generated

Figure 15: ROC curve for optimal model.

A screenshot of a map

Description automatically generated

Figure 16: Precision-Recall curve for optimal model.

We can also find the most important features (words) that our optimal classifier has identified and used for classifying questions. Figure 17 shows the top 20 words for each class. This seems reasonable as many words in the right table are offensive and ideally questions with these words should be removed from the website.

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

Figure 17: Top features for sincere question (left) and insincere question (right).

**Conclusion and Future Work**

In the previous sections we discussed data wrangling, EDA and modeling. As we saw, for our problem, logistic regression seems to achieve the best precision-recall curve. Therefore, based on business needs, one can use logistic regression classifier with a certain threshold to achieve the best acceptable/desired levels of precision and recall.

For example, we may we may want to be very aggressive and minimize false negatives (maximize false positive). In this case, we need to operate at a point with high recall/low precision. This would lead to flagging many questions as insincere and depending on how we decide to treat inappropriate questions, can lead to removing many innocent questions and/or users. Of course, we can still manually check the flagged questions, but this would increase our operation cost for the website.

On the other hand, when the goal is to be more relaxed, we may tolerate some false negatives. In that case, we can operate at higher precision/low recall and consequently, remove/flag fewer sincere questions as insincere. However, the downside would be missing some of the truly insincere questions which can be offensive to some of the user.

Overall, it can be a challenging task to find the optimal operating point (threshold) and other factors such as user experience, business and finance need to be considered in order to make the best decision.

Although we considered 3 different classifiers, other methods such as tree-based methods can be considered to expand this work. Also, it would be interesting to use other metrics such as F1-score as well as additional hyper-parameters to optimize models.