Quora Insincere Questions

Team:XG

Jaymeen Kachrola
MT2020035
International Institute of Information Technology
Bangalore,India
jaymeen.kachrola@iiitb.org

Parth Patel
MT2020057
International Institute of Information Technology
Bangalore,India
parth.patel@iiitb.org

Yagnik Bharadwa
MT2020130
International Institute of Information Technology
Bangalore,India
yagnik.bharadwa@iiitb.org

***Abstract*—Quora is a platform that empowers people to learn from each other. On Quora, people can ask questions and connect with others who contribute unique insights and quality answers. But sometimes questions ask on the platform are disparaging, intended to insult and disrespect a particular community, uses sexual content for spreading nuisance, founded upon false premises, or intend to make a statement rather than looking for helpful answers. These questions are considered insincere questions. A key challenge is to weed out insincere questions to make the platform a trustworthy and genuine source of information for all the people who use it.**
We present a project to classify sincere and insincere questions using different ML algorithms and ensembles with the use of different embedding. We have shown that logistic regression and TFIDF embedding technique perform best among others by giving an F1 Score of 0.6320.

***Index Terms*—Feature Engineering, Grid Search, MultinomialDB, Decision Tree, Logistic Regression, Passive Aggressive Classifier, Random Forest, XG Boost, Light GBM, Weighted Averaging, Voting, Tokenization, Lemmatization, Stemming, TFIDF, BoW, Word2Vec, WordCloud.**

I. Introduction

Before three decades the word internet was unknown to a major part of the human population. But now, the internet has gained popularity over the years. It has now become one of the most important things in the daily lives of people. The main reason for this popularity is the way the internet simplifies tasks that were very difficult before. One of the most popular uses of the internet is to search for solutions to questions but this was soon upgraded to asking questions on the website. These types of websites are known as question forums. Quora, Stack Overflow, Wiki Answers are some examples of question forum websites. Quora can be used to ask simple, personal, professional questions. Some people use the website for getting advice about the career of a personal opinion. This allows in forming a community and helping each other through tough times. But these forums have a huge drawback of people misusing them. People tend to ask questions that do not sound proper. They may be targeted at a group of people or make no sense. These questions tend to disturb the main cause of creating such websites. Therefore it is important to remove such questions before they harm someone emotionally. We propose a project to distinguish between sincere and insincere questions, which helps to remove the insincere question from the platform.
The report represents the entire workflow we followed for the classification of insincere questions. We started with Exploratory data analysis(EDA), then done pre-processing of data to convert data to Machine Learning models feedable format. Then We employed differently in supervised machine learning algorithms for classification and lastly, we used ensemble technique for score boosting.

The report proceeds as follows: **Section 2** describe our Dataset. **Section 3** covers EDA and Observation from Dataset, **Section 4** will discuss about Pre-Processing and Feature Engineering. **Section 5** will discuss about training methods and compare result with among the models and **Section 6** concludes the report.

II. Dataset

The dataset contains a training set of over 784000 labeled examples and a test set with over 522000 unlabeled examples. Each example in the training set has a unique id, a question, and a label of '0' or '1' to represent 'sincere' or 'insincere' respectively.

A. What is an insincere question?

- Is disparaging or inflammatory.
- Uses sexual content for spreading nuisance, and not to seek genuine answers.
- Is intended to insult and disrespect a particular community or specific group of people.

The evaluation metric used is F1-Score which is the harmonic mean of precision and recall. It is a better metric to use if we need a good balance between Precision and Recall.

F1-score is preferable over accuracy because there is an uneven class distribution in the dataset.

III. Observations

Dataset itself is highly imbalanced as there are only 7% of insincere questions available to us as compared to 93% sincere questions.

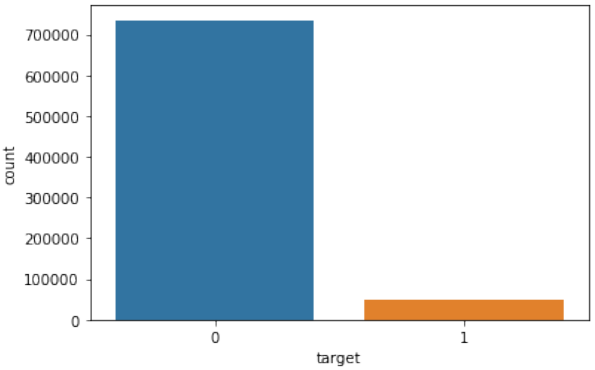

Fig. 1. Imbalance Between Insincere and Sincere Questions

Figure 2 word-cloud shows that insincere questions contain lots of offensive words like brainless, non-whites, shoving, asinine, gazans, etc. The bigger the word more frequent it appears in insincere questions.

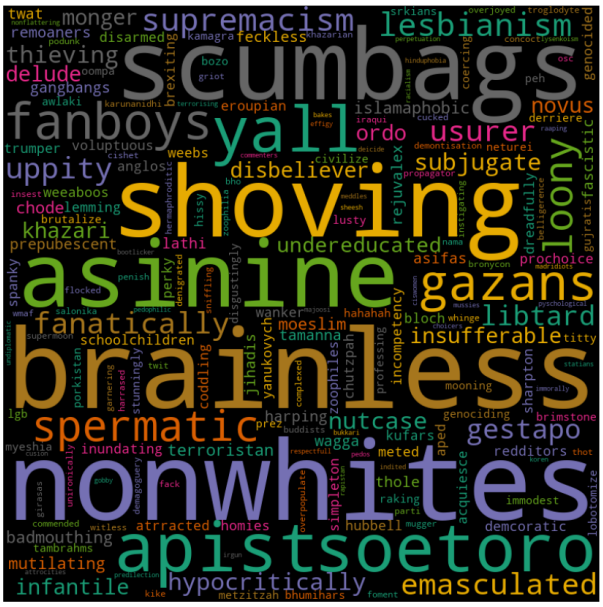

Fig. 2. Words Exclusive to Insincere Questions

Figure 3 word-cloud shows that sincere questions contain lots of positive words that are related to academics, music, designing, etc. These words collectively represent genuine questions asked by curious minds around the world.

Figure 4 word-cloud represents those words which are common to both sincere and insincere questions. These words are mostly verbs adverbs, adjectives, and nouns.

Fig. 3. Words Exclusive to Sincere Questions

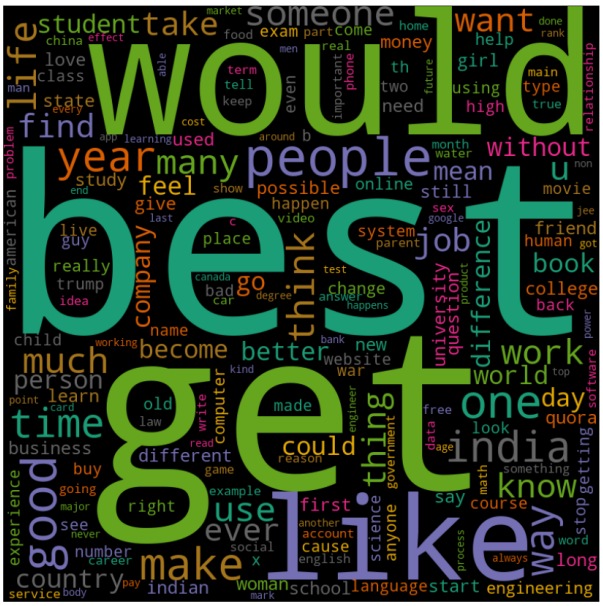

Fig. 4. Words Common to Both Sincere and Insincere Questions

We tried to see some statistical properties of the question by their length to get some insights. The length distribution of both Insincere and Sincere Questions are slightly right-skewed as shown in the Figure 5 and Figure 6. The median, mode, and mean of the length of the Sincere question is 36, 29, and 42.19 respectively. And for the Insincere question median, mode, and mean is 53, 34, and 61.40 respectively. It shows that the Insincere Question are tend to be longer than the Sincere question. Above statistical measure and word-cloud generated based on prepossessed that we will discuss in the next section.

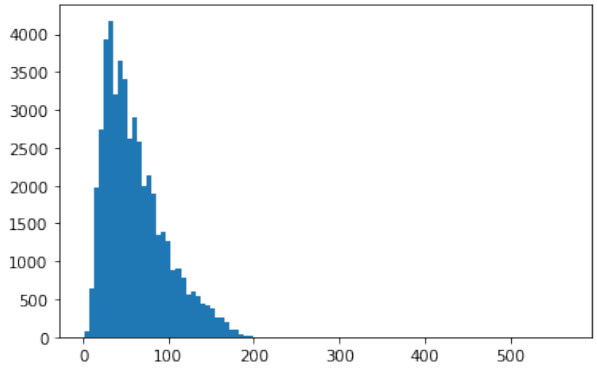

Fig. 5. Length Distribution of Insincere Questions

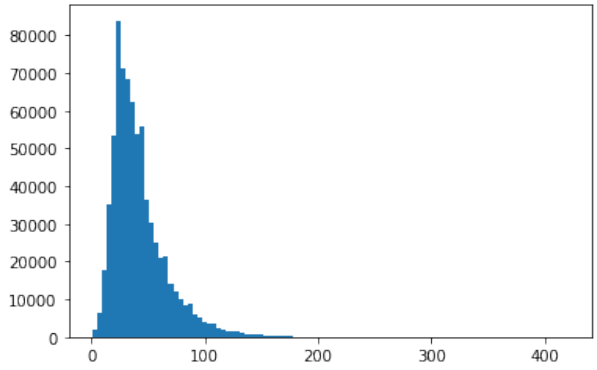

Fig. 6. Length Distribution of Sincere Questions

IV. Pre-Processing and Feature Engineering

Models trained on a meaning full data always performs well compared to raw data, so pre-processing is an important part before training the model. We started with Tokenization where we divided the entire sentence into words that are called token, then moved on to normalization, normalization means to convert all words into lower case, “Good”, “goOD”, “gOOd” will be treated the same after normalizing them. Then after we had tokens where some token were in contracted form like “I’ll”, “won’t”, “haven’t”, “don’t” etc. We removed this contraction converting them back to “I will”, “will not”, “have not”, “do not” etc respectively. Words like “trouble”, “troubled”, “troubling” simply means “trouble”, but ML models treat different words(features) so converting them to the common word or root form will significantly reduce the overall vocabulary size hence feature dimension. There are 2 techniques namely stemming and lemmatization. Stemming cut off the end of the beginning of the word taking into account a list of common prefixes and suffixes to make similar meaning words to have the same form, whereas Lemmatization convert words to their root form. We used lemmatization as stemming to generate the words that may mean nothing. There were some words like “am”, “is”, “was”, “were”, “being”, “has”, “does”, “but”, “if”, etc, which has low importance for differentiating between sincere and insincere question type. These words are called stopwords, we removed such stopwords so that model can focus on more important words during training. Punctuation is good for human reading and enhances our reading experience but it doesn’t contribute to the semantic of the sentence. Above all processing is done by nltk and spaCy python libraries[7]. In many questions, there were spelling mistakes. So to address the problem we used the following approach. We used Google news 300 dimension word2vec embedding. We checked that every word of the dataset has a vector associated with word2vec embedding or not. If there is no vector associated with a particular word then, we kept that word in the dictionary. Then we applied a spell check library for that dictionary and then replaced corrected words in the dataset itself. We added the following meta-features to the dataset.

- Negative Words - belonging to predefined set of words with negative sentiment.
- Positive Words - belonging to predefined set of words with positive sentiment.
- Unique Words - count of total distinct words in the given question
- Stop words - count of total number of stop words in the given question.
- Uppercase words - count of total number of uppercase words.
- Unseen_words - count of words which are not present in the vocabulary.
- Length - length of the sentence.

For training, model words need to be converted into a numerical value. We used an embedding concept where each word is represented as vectors. We have used common embedding techniques like a bag of words(BOW), Term frequency-inverse document frequency(TFIDF), and Word2Vec to train the ML models.

V. TRAINING METHOD

We first used Bag of Words(Bow), implemented using sklearn count vectorizer, with different ML models but the best F1 score we got by using Logistic Regression which was around 0.53, other models like Multinomial NB and RandomForest gave F1 score 0.5086 and 0.442 respectively. Above mentioned scores are the best result that we got after employing a grid search on the respective model for optimal hyperparameter search. Then we used Google news 300 dimension word2vec for embedding. In this case, Random Forest gave 0.4093 F1 score and Logistic Regression gave 0.5259 F1 score. We tried to remove meta-features and spell checking we did for pre-processing to check whether we are unnecessarily increasing the complexity of the model or not. But it turns out that without spell check and meta-features we were only able to get the F1 score around 0.47. We tried to use SVM in but it was taking so much time to converge so we did not use SVM in further experiments.

Then we moved our focus to Term Frequency Inverse Document Frequency(TF-IDF) which is a numerical statistic

that is intended to reflect how important a word is to a document in the corpus. We first used Logistic Regression with TFIDF and we got a very good F1 score around 0.58 that beats all the previous score we got so far. We were pretty impressed with the performance of Logistic Regression it gave this beautiful result at its default setting. We also employed other models to see whether using TFIDF do they give better F1 score or not. We used Multinomial NB, Random forest, and Passive-Aggressive Classifier next and they gave F1 Score 0.4428, 0.4618, and 0.53 respectively. These models are not still able to outperform the Logistic Regression F1 score at their default setting. Now we have to tune the hyperparameter of the respective model to get the best result from them, so we employed a grid search above each model and let it run overnight. We found the best hyperparameter for each of the above-mentioned models but still, they were not able to outperform Logistic Regression. We thought of using the thresholding technique to change the default threshold 0.5 to a value when it gives the best F1 score. Unfortunately passive-aggressive doesn't provide a probability of class prediction so we were not able to use thresholding on that model but the other 2 models(Random Forest and Multinomial NB) were giving the probability of class prediction. Using thresholding those 2 models gave a slightly better F1 score but still not able to reach the F1 score of Logistic Regression. We found that all model threshold around 0.75 to 0.8 was able to give the best F1 score. Now we are pretty much sure that logistic regression is the only model that can be best suited for a given classification problem but we are yet to experiment with ensembling techniques. We did a grid search for the best parameter for Logistic regression and then we applied thresholding on top of that so we got an F1-score 0.61.

Now we finally tried out ensembles. We first tried to use Light GBM(LGBM)and XGBoost(XGB) at their default setting but they were crashing the notebook because there were consuming more ram than which was available on google colab. So we reduced its number of estimators to 10 and it worked. Then using grid search we found optimal parameters for both the model and the F1 score corresponding to LGBM and XGB was 0.5823 and 0.5898 respectively[1][4]. Then we used Max voting of Logistic Regression, XG Boost, and Light GBM which gave an F1 score of 0.5987. Then we applied Weighted Averaging of Logistic Regression, XG Boost, and Light GBM, which gave a score of 0.6054[2][3].

After all these efforts we were able to get the best 0.61 F1 Score using Logistic Regression. Then we were suspicious about it may possible that we have removed some of the important information during pre-processing. And We talked to a group, they had a good leader board score, about that and they told us that our suspicion was correct and they also told us to do hyper parameter tuning in TFIDF vectorizer. Then to do just punctuation removal and contraction removal and nothing else and we tune some hyperparameter of TFIDE vectorizer and magic happened !! We got an F1 score of around 0.6320. We tried to apply this process to another model in hope that they might perform well but those models were not kind enough to give us a better score than what we got using Logistic Regression. In the *Table 1. ML algorithms and F1 score* we mentioned the best score we got so far for a particular model using the respective embedding technique.

ML Algorithm	**Embedding**	**F1 Score**
Logistic Regression (LR)	BoW	0.53
Multinomial NB	BoW	0.5086
Random Forest	BoW	0.442
Logistic Regression	Word2Vec	0.5259
Random Forest	Word2Vec	0.4093
Logistic Regression	**TFIDF**	**0.6320**
Multinomial NB	TFIDF	0.4428
Random Forest	TFIDF	0.4618
Passive Aggressive Classifier	TFIDF	0.53
Light GBM (LGBM)	TFIDF	0.5823
XGBoost (XGB)	TFIDF	0.5898
Max Voting (LR, XGB, LGBM)	TFIDF	0.5987
Weighted Averaging (LR, XGB, LGBM)	TFIDF	0.6054

Table 1. ML Algorithms and F1 Score

VI. Conclusion

We would like to conclude that we came up with a decent model to classify insincere and sincere questions. Considering the constraints of using only traditional approaches, logistic regression performed best among other ML models and ensembles. We strongly believe the use of deep learning gave us a better F1 score than what we got till the date.

Acknowledgment

We would like to thank professor G. Srinivas Raghavan and our machine learning teaching assistants, for giving us a challenging opportunity to work on the project. We learned many new things that will help us in the future. They helped us whenever we were stuck, provided support when needed. We would like to thank them for providing us ideas and resources which helped in learning new concepts. We are grateful to them for explaining the critical aspects of topics related to the project.

It was a great learning experience while working on the project. The friendly competition motivated us to work and come up with new ideas and try different approaches.

REFERENCES

[1] Mandot, Pushkar. "What Is LightGBM, How to Implement It? How to Fine Tune the Parameters?" Medium, Medium, 1 Dec. 2018, medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc.

[2] S. Yu, "Stacking and Blending-Intuitive Explanation of Advanced Ensemble Methods," Medium, 30-Sep-2019. [Online]. Available: https://medium.com/@stevenyu530_73989/stacking-and-blending-intuitive-explanation-of-advanced-ensemble-methods-46b295da413c.

[3] T. Kotha, "frenzytejask98/ML_TA_IIITB_2020," GitHub. [Online]. Available: https://github.com/frenzytejask98/ML_TA_IIITB_2020/blob/master/November_27/Ensemble_Techniques.ipynb.

[4] MJ Bahmani Data Scientist amp; Machine Learning Researcher, M. J. Bahmani, and Data Scientist amp; Machine Learning Researcher, "Understanding LightGBM Parameters (and How to Tune Them)," neptune.ai, 30-Nov-2020. [Online]. Available: https://neptune.ai/blog/lightgbm-parameters-guide.

[5] "1. Supervised learning," scikit. [Online]. Available: https://scikit-learn.org/stable/supervised_learning.html.

[6] Aishwarya SinghAn avid reader and blogger who loves exploring the endless world of data science and artificial intelligence. Fascinated by the limitless applications of ML and AI; eager to learn and discover the depths of data science., "Ensemble Learning: Ensemble Techniques," Analytics Vidhya, 28-Nov-2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/.

[7] How to solve 90% of NLP problems: A step-by-step guide. (n.d.). Retrieved December 20, 2020, from https://www.kdnuggets.com/2019/01/solve-90-nlp-problems-step-by-step-guide.html.