Quora Insincere Questions Classification

INTERNSHIP PROJECT

Sumit Mishra | Research Intern | 10 August 2020

Table of Contents

Introduction

An existential problem for any major website today is how to handle toxic and divisive content. Quora wants to tackle this problem head-on to keep its platform a place where users can feel safe sharing their knowledge with the world.

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. A key challenge is to weed out insincere questions -- those founded upon false premises, or that intend to make a statement rather than look for helpful answers.

This is a data science project which is about classifying those Insincere questions from the rest and provide users a better experience and give them the quality answers they are looking for. Hence the dataset consists of the questions and the labels if they are Sincere or Insincere. This is a binary classification task on the text data. In this project 4 Machine Learning Models are trained which are Logistic Regression, Naïve Bayes, CNN, and BERT (Bidirectional Encoder Representations from Transformers). The Preprocessing and Exploratory Data Analysis is done on the dataset to get the best insights.

Dataset

- The Dataset is taken from a competition named Quora Insincere Questions Classification on Kaggle. In the dataset an insincere question is defined as a question intended to make a statement rather than look for helpful answers. Some characteristics that can signify that a question is insincere are Has a non-neutral tone, is disparaging or inflammatory, isn't grounded in reality and Uses sexual content (incest, bestiality, pedophilia) for shock value, and not to seek genuine answers.
- The Dataset contains 1,306,122 questions. The questions are in different languages like Hindi, Japanese, Thai, Chinese and others but most of the questions are in English language.
- The dataset contains three columns named "Qid" which is the question ID, "Question_text" which are the questions and the "target" which is the label if the question is sincere or not. The target variable is label-1 which indicates Insincere Question.
- The dataset is skewed as there are 93.81% of the Sincere questions and 6.19% of Insincere Questions.
- The dataset does not contain any null value in any of the columns so there won't be any requirement of filling null values.

Preprocessing

The text data is processed before the feeding it to the Machine Learning model. The preprocessing is done in two parts the preprocessing-I is done for the Logistic Regression and Naïve Bayes. The preprocessing-II is done using the GloVe Word Embedding from Stanford NLP for the CNN Model. Both the types and their steps are given below.

Preprocessing – I

In this Preprocessing, I've done all the necessary steps for processing the text data.

- Lowercasing – The text data is lower cased before any step so that other steps will be even for all the data points.
- HTML Removal – HTML is removed from the text data.
- Email ID Removal – Email IDs are removed from the text.
- URL Removal – URLs starting from https and www are removed from the dataset.
- Contraction Expansion – Contractions are expanded so the stop word removal will be effective for all the data points.
- Stripping the Possessives – Possessives are stripped and are interchanged with whitespace.
- Punctuation Removal – Punctuations are Removed.
- Special Character Removal – Special character which are the non-ASCII characters are removed.
- Stopwords Removal – Stopwords are removed from the text.
- Removal of Leading and trailing Whitespaces from text.

- Word Lemmatization – The words in the text are lemmatized so that only the main word with the meaning remains.

Preprocessing-II

In this Preprocessing Step, the text data is processed according to the GloVe Embedding from Stanford NLP. The main aim of this preprocessing is to increase the Vocabulary Coverage of the words in embedding vector. The initial Coverage of the words were 88.159% of all the words (repeating or non-repeating) and 33.159% of all the unique words in the vector. The text data is processed so as to increase the vocabulary coverage. The steps are given below.

- The punctuations are checked to see what punctuations are already given in the embedding vector. Most of the punctuation are covered in the embedding vector expect some.
- The punctuation mapping is done so as not to remove too many of them and after mapping. Those punctuations which are not present are removed from the text.
- Expansion of Contractions is also done on the text.
- The Non-English text data is also given in the dataset so they are converted in to English Language using the Google Translate API.

After Doing all the steps the Final Coverage after the steps was 99.593% of all the words and 75.342% of all the unique words in the vector.

Exploratory Data Analysis

- The Exploratory Data Analysis is done on the question to get more insights about the data. Different bar graphs, word clouds, violin plots, pie charts and heatmap are plotted. The plotted figures are Distribution of the target, word clouds for Insincere and Sincere question, word Count plots, engineered features plot, ngrams consisting –unigrams, bigrams, trigrams, Distribution of the Polarity & Subjectivity of the questions and Correlation heatmap for the features.
- The preprocessing-I is done before the Exploratory data analysis but the original questions are used in some of the plots as it was required.

1. Distribution of the question types in the dataset

   A bar graph and a pie chart is plotted to see the distribution of the target in the dataset.

   We can see that the Dataset is skewed where the quantity of the insincere questions is very small i.e. 6.19% only. On the other side the quantity of the Sincere questions is fairly larger i.e. 93.81%.

   Hence Accuracy will not be the best metric for this classification.

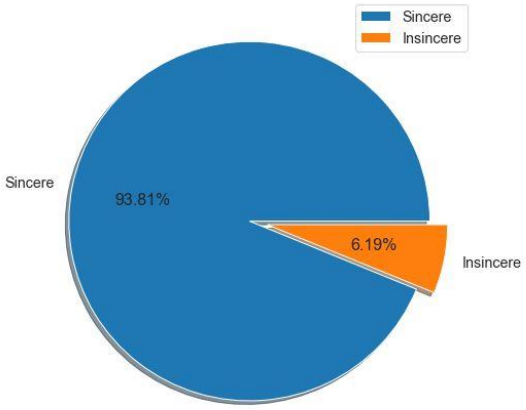

Figure-1 Pie chart

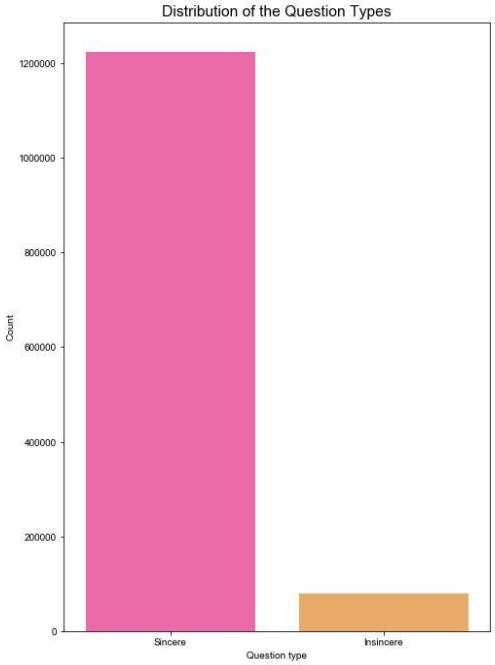

Figure-2 Bar graph

2. Word Clouds

The first Word Cloud is for the Insincere Questions given in the figure -3. Here the most Highlighted words are, “Trump”, “Muslim”, “People” and “Woman”.

The Second Word Cloud is for the Sincere Questions given in the figure -4. Here the most Highlighted words are “Best” and “One”.

Figure - 3

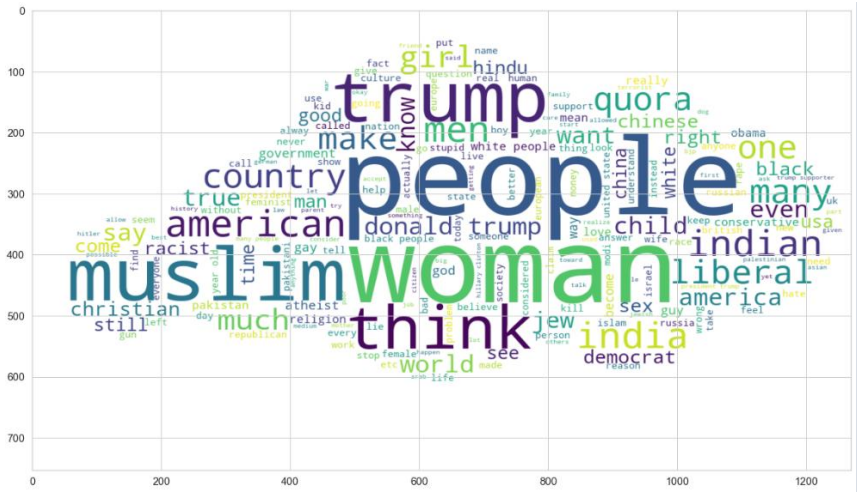

Figure – 4

3. Word Count Plot for Insincere Questions

   The Word Count plot for the Insincere Questions shows the top 50 most occurring words in the Insincere questions. Here the most occurring words are like people, Woman and Trump.

4. Word Count Plot for Sincere Questions

   The Word Count plot for the Sincere Questions shows top 50 most occurring words in the Sincere questions. Here the most occurring words can be seen like Best, get, people, good and India.

5. Polarity of the questions

   Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. We can see that the most of the questions are neutral. This polarity distribution is given in the figure -5.

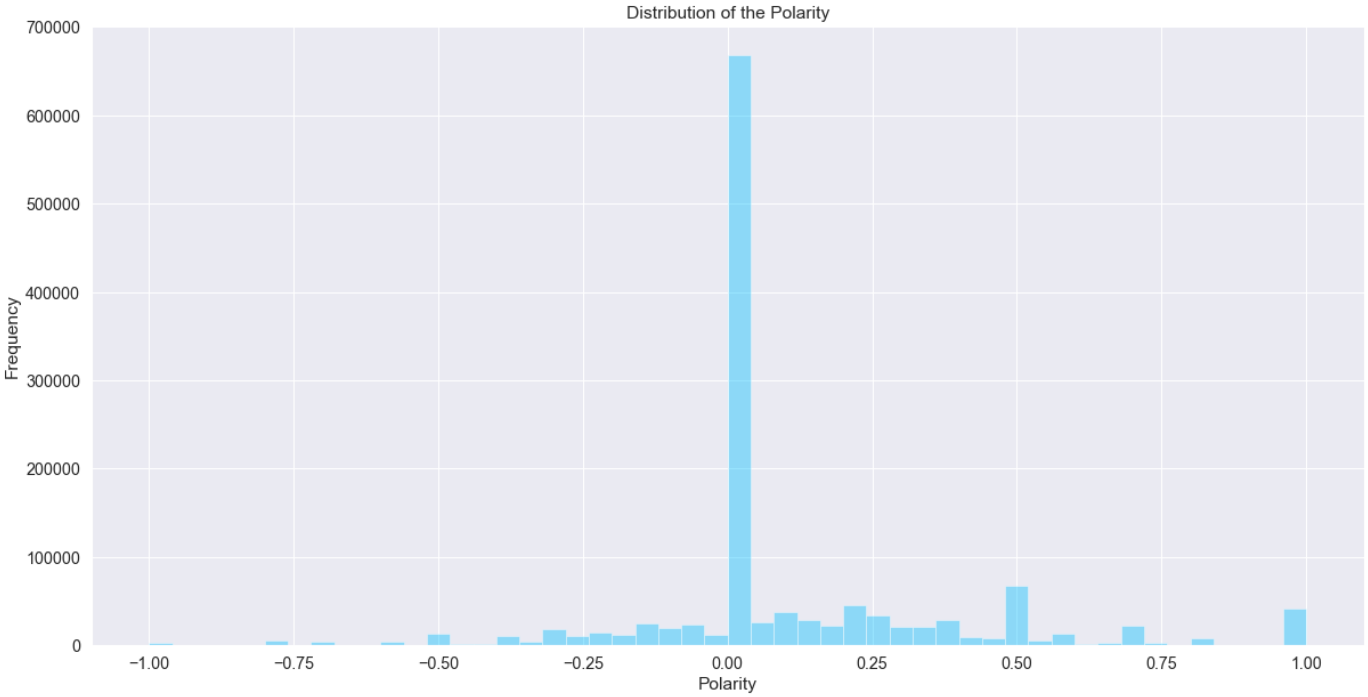

Figure – 5

6. Subjectivity of the questions

Subjective sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information. **Subjectivity** is also a float which lies in the range of [0,1].

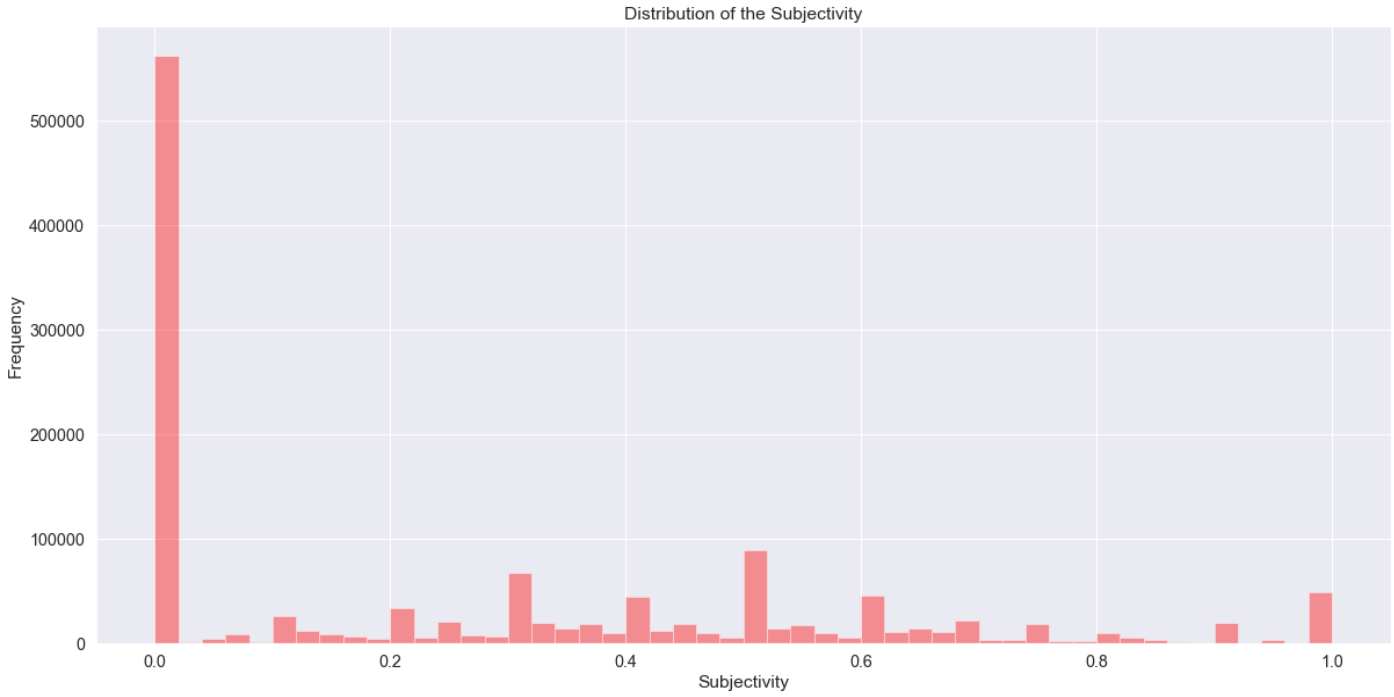

Figure - 6

Feature Engineering

The Feature Engineering is done on the text data and certain Features are developed from the data and their distribution are also plotted during the Exploratory Data Analysis. These Features are given below:

1. Number of Words in each Question
2. Number of Unique Words
3. Number of Character
4. Number of Stop Words
5. Number of Punctuations
6. Number of Title case words
7. Average Length of Word
8. Polarity of each Question
9. Subjectivity of Each Question

Machine Learning Models

I. Logistic Regression

Logistic regression is a powerful machine learning algorithm that utilizes a sigmoid function and works best on binary classification problems, although it can be used on multi-class classification problems through the “one vs. all” method.

Logistic Regression in this, is trained two times with two different method. The first method is using the TFIDF Vectorizer for converting the processed text data in to numerical vectors.

Hyperparameter of the logistics regression in this case were:

```
LogisticRegression (C=1, solver='sag', penalty = 'l2', random_state
= 0)
```

The accuracy of this model is 95.180% with the F1 Score of 0.643 and precision 0.59.

In the second method the engineered features are used, this model has the same hyperparameter as the above model.

The accuracy of this model is 95.180% with the F1 Score of 0.248 and precision 0.19.

We can see that the LR model with the engineered features is giving a very poor result and the one with TFIDF is giving a good result.

II. Naïve Bayes

Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models. Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem.

In Naïve Bayes Classifier I've used the Multinomial Naïve Bayes classifier. Like the logistic regression the Naïve Bayes classifier is also trained for two methods. The first method is using the Count Vectorizer. The hyperparameters of NB in this method were:

```
MultinomialNB (alpha = 1)
```

The accuracy of this model is 93.707% with the F1 Score of 0.562 and precision 0.49.

In the Second method the TFIDF Vectorizer is used. The hyperparameter for this model were:

```
MultinomialNB (alpha = 0.3)
```

The accuracy of this model is 93.932% with the F1 Score of 0.571 and precision 0.51.

III. CNN

A convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, natural language processing, and financial time series.

Two methods were used for the CNN model. In the first methods the GloVe embedding is used without increasing the vocabulary coverage and in the second method the same neural network is used with the vocabulary coverage increased.

The Structure of the CNN used in both the methods is:

```
model = Sequential ()
model.add(Embedding(num_words, 300, input_length=100, weights= [embe
dding_matrix], trainable=True))

model.add(Dropout(0.2))
model.add(Conv1D(64, 5, activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=
['accuracy', f1])
model.summary()
```

The accuracy by using the first method is 95.46% with the F1 Score 0.630 and precision 0.59.

The accuracy by using the second method with increased vocabulary coverage is 95.53% with the F1 Score 0.664 and precision 0.62.

IV. BERT

Bidirectional Encoder Representations from Transformers (BERT) is a technique for NLP (Natural Language Processing) pre-training developed by Google. The original English-language BERT model used two corpora in pre-training: Book Corpus and English Wikipedia. I've used the BERT from the hugging face library using pytorch.

The Text data is firstly prepared for the BERT model using the BERT encoders. The data masking is also done. The number of epochs is equal to 1 due to the resource limitations.

It was trained on 783,673 questions and for validation it was 208,979 and for testing 313,470 questions are used.

```
class QuestionClassifier(nn.Module):
    def __init__(self, n_classes):
        super(QuestionClassifier, self).__init__()
        self.bert = BertModel.from_pretrained(PRE_TRAINED_MODEL_NAME
)
        self.drop = nn.Dropout(p=0.3)
        self.out = nn.Linear(self.bert.config.hidden_size, n_classes
)
    def forward(self, input_ids, attention_mask):
        _, pooled_output = self.bert(
          input_ids=input_ids,
          attention_mask=attention_mask
        )
        output = self.drop(pooled_output)
        return self.out(output)
```

The Accuracy of the BERT model is 95.54% with the F1 Score 0.603 and precision 0.67.

Result

Logistics Regression

Model Type	Training Samples	Testing Samples	Accuracy	F1_score	precision	Recall
With Processed Text and TFIDF	1,044,897	261,225	0.951	0.643	0.59	0.71
With Engineered Features	1,044,897	261,225	0.950	0.248	0.19	0.36

Naïve Bayes

Model Type	Training Samples	Testing Samples	Accuracy	F1_score	precision	Recall
With Processed Text and Count Vectorizer	1,044,897	261,225	0.937	0.562	0.49	0.66
With Processed Text and TFIDF	1,044,897	261,225	0.939	0.571	0.51	0.66

Convolution Neural Network

Model Type	Training Samples	Testing Samples	Accuracy	F1_score	precision	Recall
GloVe Embedding Without the Vocab Increase	1,044,897	261,225	0.954	0.630	0.59	0.68

GloVe Embedding With Vocab Increase	1,044,897	261,225	0.955	0.664	0.62	0.71

BERT

Model Type	**Training Samples**	**Testing Samples**	**Accuracy**	**F1_score**	**precision**	**Recall**
Hugging Face BERT epoch - 1	783,673	313,470	0.955	0.603	0.67	0.55

Conclusion and Future Work

The main aim was to predict if the question on Quora is Sincere or Insincere. Hence Exploratory Data Analysis was done to get more insight into the dataset and preprocessing was done. Preprocessing was done in two parts; In the first part all the basics processing is done and in the second part the preprocessing is done with respect to the GloVe Embedding by Stanford. Initially, only the text data was given, several features were engineered during the EDA which was used in the Logistics Regression. The feature correlation heatmap was also plotted. In machine learning modeling, 4 Classification models were trained which are Logistic Regression, Naïve Bayes Classifier, CNN, and BERT. The best performing model was CNN with increased vocabulary coverage. The accuracy of the model was 95.53% with the F1- score of 0.664. The Classification metric used was f1-score and precision with accuracy as the dataset was skewed. The classification report is also there for a deeper analysis of the model performance.

For future work, the new models can be trained and more epochs can be incorporated in the BERT model. Different types of embedding can also be used and different approaches for increasing the vocabulary can be implemented or new preprocessing techniques can also be implemented.