University of Adelaide

Academic Year 2021-22

Master of Data Science

Quora Insincere Questions Classification

Name: Yash Kasundra

Id: a1838670

Subject: Big Data Analysis & Projects

Contents

1 Abstract

The purpose of this project is to classify the questions asked on Quora to be sincere or insincere. What is Quora? Quora is an online platform where a person can learn from other's experience. On Quora people can ask different types of questions and connect with different contributors who provide different insights and solutions/answers. But the main problem is to weed out insincere questions. So what are insincere questions? So the questions that have rhetorical nature, that suggest a discriminating idea about a group of people or particular society, or questions that may hurts someone's religious beliefs, cultural values or political standpoints. Questions that contains false information or preposterous assumptions or sexual content are also classified as insincere questions.

The first step of this project was to explore the data before making any assumptions. Few activities that I performed during EDA (Exploratory Data Analysis) are checking for null values or duplicates if any. Checking the distribution of data among 2 classes i.e., sincere or insincere, making a word collage of most frequent words from those 2 classes using word cloud. Plotting few ngram plots to again check the frequency of most used words. My next step was to extract features from the data and plotting few graphs to visualize them. Since it was text data, I checked for mathematical formulas, HTML tags, punctuation marks, emojis and how they were distributed among the 2 classes..

So based on insights gained from EDA it was clear that my data too impure to directly pass into a model. As if u pass junk and train your model on it then your prediction would also be useless. Thus, I started with creating a few functions for replacing maths equations and urls with a tag, cleaning punctuation marks and emojis, correcting misspells, changing word contractions, lemmatizing the data and removing stop words like is, a, of etc. which adds no meaning to a sentence when we are training a basic model. So now my data was ready to be used for training purpose, next step was to learn and investigate about few models mostly about the theoretical working and parameters that they used instead of complex mathematical formulas. Next step was to train several different models and evaluate then compare their performance. For the last step of my project was to use 2 different methods to handle skewed data and use the best model from above generated models.

2 Introduction

This project involves predictions on questions asked on Quora. The training data set contains more than 1.3 million samples and 3 column: 1st column is just question ID, 2nd column is question texts from Quora and 3rd column are labels for those texts. This is still an ongoing competition on Kaggle with the price of $25,000.

The key constraint of this competition was to not use prebuild models, but since this is for my Big Data project, we are allowed to use pre-build models. The project started with EDA, in which I found that my data was highly skewed and contained no null or duplicate values but since it was text data it had a lot of misspells, contractions, punctuation marks, emojis, mathematical formulas, HTML tags, and urls. So my next step involved creating different functions to handle above problems and get a clean data that can be fed to my model. For training and testing part I created many different models and used F1 score as validation method since our data was skewed. I also used few methods to handle skewed data and used the best model from the above F1 score.

3 Methodology

3.1 EDA

Understanding the data or getting the feel of data by exploring the data set is an important part of project, because it helps us to find errors or mistakes in our data sets and gain valuable insight to achieve clean data which is ready to be used by our models. So the more we understand our data, the more features we can extract from it. During this process I also checked correlations between different features that I expected.

3.2 Data Preprocessing

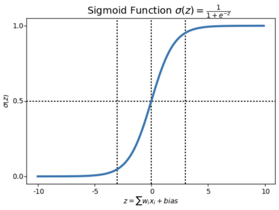
This step is the most important step of all, if this is not executed properly we might still have junk values which would decrease the accuracy or might make our model over fit. This task is all about cleaning the data of unnecessary information (punctuation marks, emojis, formula etc.). This happens because data is gathered from different users or multiple platforms and it is in raw format, so this data is not viable for analysis.

3.3 Research models

Now we have clean data to be used for training purpose, but we still need to figure out which models to use. Since there are a lot of models out there today, so the best way is to first classify our problem from classification or regression. Mine is a binary classification, so research few models mentioned below.

3.3.1 Logistic Regression

This is a concept that has been borrowed from maths. In mathematical terms, Logistic regression is a process of modeling the probability of a discrete outcome given an input variable [1] and in machine learning it is one of the simplest models that can be used for binary classification problems, which is easy to create and use. Input values are combined linearly using weights or coefficient values to predict an output value, just like Linear Regression. Only key difference is it uses Sigmoid function as its core.



Sigmoid Function Graph

$$f(x) = \frac{1}{1+\mathrm{e}^{-(x)}}$$

Figure 1: [2]

3.3.2 Linear Support Vector Classification

Linear Support Vector Classifier is a machine learning model, that takes multidimensional vector and class labels for those data. It then tries to figure out the boundaries for different classes, thus the prediction on new test data can be done just by checking in which class boundary the value falls on.

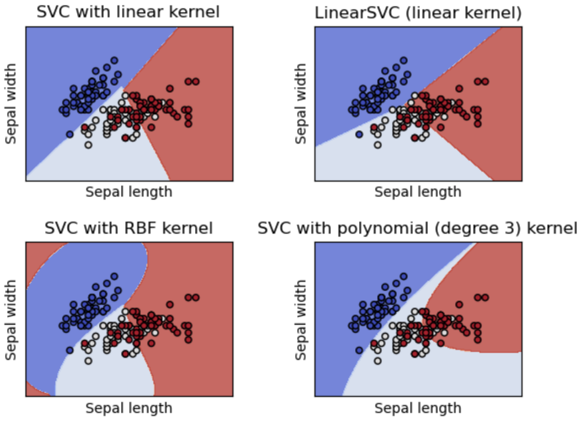


Figure 2: [3]

As you can see in the image there were 3 classes and based on how training data for those classes were scattered it created different boundaries and then uses this boundaries to predict the classes of testing values.

3.3.3 Stochastic Gradient Descent

Stochastic gradient descent forms the basis of Neural Networks, so I though before moving directly to neural networks it would be better to check performance of SGD model. It is also very common algorithm used in various Machine Learning problems. Gradient means slope of a surface and thus Gradient decent means descending a slope. The objective of gradient descent algorithm is to find the value of "x" such that "y" is minimum. So, to achieve that in a gradient decent algorithm it will select a random place to start, then it would start to iterate downwards to the lowest point.

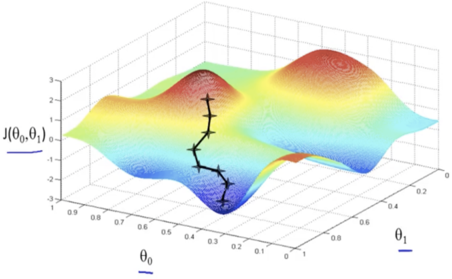


Figure 3: [4]

As the above figure shows how gradient decent works, each horizontal line is a new iteration and it keeps on learning and descending towards to lowest focal point and it will not descent to the global lowest point to the nearest lowest point (based on its starting position). But the main problem with just gradient descent is its huge computational necessity. That is where Stochastic Gradient Descent comes to picture, while selecting data points at each step to calculate the derivatives, SGD randomly picks one data point from the whole data set at each iteration to reduce the computations enormously.[5]

3.3.4 Artificial Neural Network

The inspiration of these Artificial Neural Network comes from biological neural networks. Just like human/animal brain contains many neurons connected to each other, similarly this ANN also has many layers and each layers contains different set of neurons which is fully connected to other neurons from different layers. ANN basically has a input layer, 1 or more hidden layer and an output layer at the end. Each neuron in this Artificial Neural Network is a model (like linear regression) composed of input data, weight, bias and an output. Now, once an input layer is determined (use pretrained, build your own or hybrid), weight are assigned which in turn helps in determining importance of a any variable. Ones that have larger values have more influence on the output. This output is passes to an activation function, which decides which output to select. If the value of that output exceeds the defined threshold then it activates the node, passing it to next layer. This was forward propagation and similarly there is backward propagation.

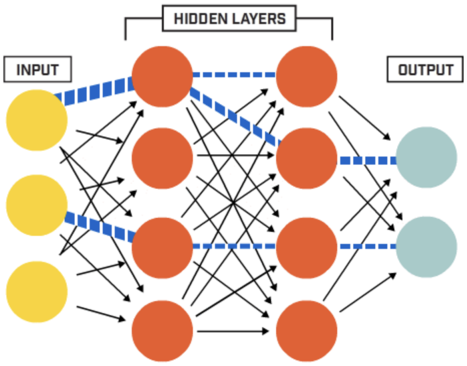


Figure 4: [6]

3.3.5 Convolutional Neural Network

Convolutional Neural Networks (ConvNets) have shown break-through results for some NLP tasks in past few years, one particular task is sentence classification, i.e., classifying short phrases. It was originally developed for image classification and they achieved great results in recognizing an object from pre-defined categories. It has 2 operations: convolution and pooling, the output of this sequence of operations is then typically connected to fully connected dense layers.

In convolution layer it takes windows of pre-defined kernel size from the input matrix. Which is then multiplied to a pre-defined kernel matrix and all output is added to get a numeric value. Just as shown in the picture below.

Then in pooling layer we downsize the matrix that was formed in convolution layer by taking a pre-defined window size and then either taking the average of those values or taking the maximum value. Some what similar to what we do in convolutional layer.

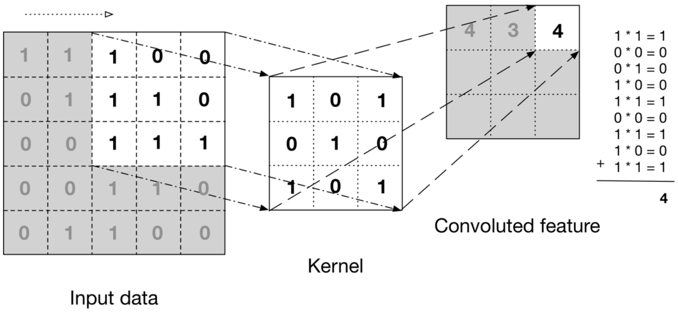


Figure 5: [8]

3.3.6 Training and Testing

This step is about splitting training data to get train, test and validation datasets. In this project, I am splitting my training dataset into 3 parts. It is achieved by first splitting data into 80:20 ratio to create a train and test dataset, then taking this train split and again splitting into 90:10 ration to get train set as well as validation set. These split ratios remain same but number of data will change after implementing Undersampling and Oversampling to decrease imbalance between the 2 classes. And using sklearn's train_test_split library to achieve the result.

3.3.7 Evaluation

3.3.7.1 F1 Score

Since the dataset is highly imbalanced/skewed we have to use F-1 accuracy in order to evaluate if or not our models are performing well. Because if we use accuracy as a measure then if our model will just predict 0 for all values then also it will get 90% accuracy. While F-1 score is a metric that will use recall and precision as shown below.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad \text{F1} = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Figure 6: [7]

3.3.7.2 Matthews correlation coefficient:

- The coefficient takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes.

- The MCC is in essence a correlation coefficient between the observed and predicted binary classifications; it returns a value between -1 and +1.
- A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and -1 indicates total disagreement between prediction and observation

$$MCC = \frac{TN \times TP - FN \times FP}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Figure 7: [9]

4 Project Implementation

In this step will be listing all the steps or implementation involved to complete that process. And whole description can be seen on jupyter notebook.

4.1 Activities in Exploratory Data Analysis (notebook-1)

- Importing Library and Dataset
- Checking for null or duplicate values
- Checking basic stat like size, column names and actual data
- Checking sample distribution among the 2 class
- Visualize same thing for better understanding (sample distribution in diff class)
- Plotting word collage of most frequent words in both sincere and insincere class from training data using wordcloud library.
- Plotting frequency plot for 1-gram(single word) , 2-gram(double word), 3-gram(triple word) for both classes
- Creating unique features based on the given data
- Plotting box and violin plots for these features between those 2 classes to check the relation
- Plotting Correlation Matrix to better understand relations between different features that were created
- Created a function to print top-10 questions with most amount of punctuation marks
- Created a function to see which different punctuation are present in the data

4.2 Activities for Pre-processing data

Creating functions to clean the data of the anomalies found from EDA

- Replacing math equations and URL’s with common abbreviation like a MATH EQUATION tag or URL tag.
- Cleaning contractions. Like ”ISN’T” is replaced to ”Is not”, ”Would’ve” is replaced to ”Would have”
- Spelling Correction. Like “organization” is corrected to “organization” , “Litecoin” is changed to ”bitcoin”
- Removing punctuations. All punctuations including emojis, symbols etc
- Removing Stopwords. Words that occur too frequently in English language and adds a very little meaning to the sentence like A, An, The, is etc

- Using WordNet Lemmatizer, Lemmatization means converting the word to its root word like caused to cause, changing to change etc.

After Clean the data again visualize it to see how the distribution in both class changed after data cleaning by plotting box plot, violin plot as well as distribution plot.

4.3 Research Models and Methods to handle imbalance class

- Researched all the models mentioned in the 3rd step and tried experimenting with few more models
- Then learnt about Undersampling method used to handle skewed data. In this method we select few random samples from larger class like lets say class-0 has 300,000 samples while class-1 has 50,000 samples so we will try to select 100,000 random samples from class-0, so that the imbalance is less. At first imbalance was in the ration 6:1 for class-0:class-1 and now It is 2:1. But the main problem with this approach is we can't use full data and if smaller is too small like on 8,000 record in this case then obviously we can't get good result.
- Another method is Oversampling method, in this we try to duplicate the records of smaller class. Let's consider above mentioned example, in this method we will take all samples from class-0 and then try to duplicate random records from class-1 to match record size of class-0.

4.4 Training/ Test/ Validation split

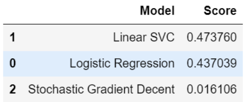
- Split training data into train/test split in
- Then we convert these splits into vectors/tokens through TF-IDF for base models, embedding layer in NN, and tokenizer library in CNN
- TF-IDF (Term Frequency-Inverse Document Frequency): is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is a mix of 2 functions TF which check for the frequency of that word in the given document or text and IDF which checks how important a term is, the weight of each word is normalized by the number of times it appears in the corpus/input data , so a word that appears in only 10% of all documents will be assigned a higher value than a word that appears in 80% of all documents.
- There are a tons of good neural networks out there which has great accuracy in classifying text data, since they have been trained on billions of data they have a better understanding of words and sentences than our models which would have been trained on lesser amount of data and using lesser computational powers. So I tried using embedding layer from 1 of the pre-built model by google for my neural network.
- I used tokenizer library of Tensorflow which helps in tokenizing words to be fed into our model. Since this is public library it can be used by anyone and then we can train our models using these tokens.

4.5 Activities Evaluation

- Analyze performance matrix (using f-1 score for all models as well as Matthews correlation coefficient)

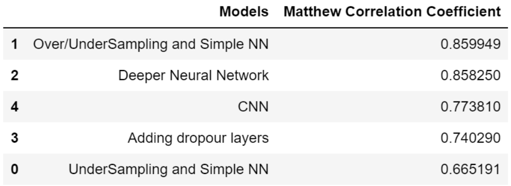
5 Analysis and Solutions

- After EDA it was clear that data had a lot of misspells, math's equations, URLs, punctuation marks and small preprocessing issues.
- Another major issue with data was its distribution among the class i.e. sincere and insincere.
- First problem was addressed appropriately in the preprocessing task.
- Next we applied basic models like logistic regression, LinearSVC and SGD. As for the first 2 f-1 score was 0.4-0.5 for class-1 which not a lot. But for SGD it was worst just 0.04 for class-1. So it was overfitting on class-0 with f-1 score of 0.97.



	Model	Score
1	Linear SVC	0.473760
0	Logistic Regression	0.437039
2	Stochastic Gradient Decent	0.016106

- Thus before moving to ANN, since SGD is like basic NN, I thought that this imbalance in the class needs to be addressed.
- So I implemented undersampling as well as mix of both oversampling / undersampling to get better results.
- I even tried experimenting with few hyper parameters and layers of neural network.
- Then at the end implemented 1D- Convolution Neural Network.
- Comparison of Matthews Correlation Coefficient of different models is shown below:



	Models	Matthew Correlation Coefficient
1	Over/UnderSampling and Simple NN	0.859949
2	Deeper Neural Network	0.858250
4	CNN	0.773810
3	Adding dropour layers	0.740290
0	UnderSampling and Simple NN	0.665191

6 Conclusion

Most models that were included and few that were not included had almost same f-1 scores for class-1 i.e. between 0.40 to 0.50. Except SGD whose performance was too poor to add here. Few other models that are not included here are:

- KNN – Due to highly skewed dataset set with 1.2 million samples classified as 0 (sincere) and only 88,000 classified as 1 (insincere). F-1 score of KNN was too low to be included
- Decision Tree – Due to sheer volume of data it was not able to converge
- Naïve Bayes – Too low F-1 score to be included.

Data Preprocessing was the most crucial part of this project. Since pass text data as it was provided was not viable since I would have not been able to change formulas and URL into tokens/vectors. Plus bad input data might have lead to bad output or false predictions and models might overfit or underfit due that.

Actually, there are still 2 ways to improve this F-1 score, firstly is experiment with different ratio of oversampling and undersampling of different classes or use some online available libraries to deal with imbalance classification of samples in train data. Another way is to still continue experimenting with hyperparameters for Neural networks as well as CNN, we can also use transfer learning i.e. use a pretrained models first few layers and then define last dense layers according to our need.

We can also use advance machine learning models like Berta from Google or roBERTa from facebook ai both of these are deep Neural Network models and can be used on Natural Language Processing problems.

References

[1] Thomas W. Edgar, David O. Manz, in Research Methods for Cyber Security, 2017

[2] https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

[3] https://scikit-learn.org/stable/modules/svm.html

[4] Stanford's Andrew Ng's MOOC Machine Learning Course

[5] https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31

[6] https://towardsai.net/p/machine-learning/introduction-to-neural-networks-and-their-key-elements-part-c-activation-functions-layers-ea8c915a9d9

[7] https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

[8] "Deep Learning" by Adam Gibson, Josh Patterson

[9] https://en.wikipedia.org/wiki/Matthews_correlation_coefficient