Quora Insincere Questions

Team: Runtime terror

1st Rahul Modak
MT2020014
International Institute of Information Technology Bangalore
rahul.modak@iiitb.org

2nd Shabbir Sidhpurwala
MT2020148
International Institute of Information Technology Bangalore
shabbir.sidhpurwala@iiitb.org

3rd Shivraj Ganacharya
MT2020152
International Institute of Information Technology Bangalore
shivraj.ganacharya@iiitb.org

***Abstract*—A constant problem for any major online platform today is how to handle toxic and divisive content. This is a difficult task as there exists lot of different views,also natural language has ambiguities which proves to be really challenging for a machine to understand and interpret.**

In this report we built a pipeline which can perform the task of identifying the sincerity of a question and can successfully flag insincere and sincere question as accurately as possible.This pipeline consists of data preprocessing techniques and machine learning model which will finally do the task of classification.

***Index Terms*—TF-IDF vectorizer,logistic regression,Xgboost,boosting,voting classifier,Natural language processing,stemming,Stacking**

I. Introduction

In this ever growing era of social media nowadays there are uncountable platforms that are present and almost each one of them has some forums on which users can ask or post questions and discuss them which allows everyone around the whole world to interact with each other,some of them are only developed for discussions and allows users to ask questions like quora,reddit,stack overflow,Yahoo answers etc.These platforms has many advantages like it helps forming community so that they can help each other but at the same time these platforms can be misused by some people so there is need to be keep check on the content that is being posted on these platforms.Our project is focused on questions.

Presently,there exist many different Q and A platforms such as reddit,quora etc.The task of classifying questions as sincere or insincere.Sincere mean the question asked is legitimate and insincere means a question is intended to make a statement rather than look for helpful answers some characteristics maybe non-neutral tone,inflammatory,sexual content.But this classification task is challenging because natural language is highly ambiguous so to train a machine to make sense out of this natural language requires specific data preprocessing,model training techniques.

The report presents methodology for correctly classifying questions as sincere and insincere by leveraging large corpus of online data and traditional machine learning models.The corpus contains questions in text format that were posted on quora(online Q and A platform) and each question is labeled(ground truth) with 0 meaning sincere and 1 meaning insincere.

The rest of the report is as follows,the first part introduces the topic and problem that we are trying to solve,second part is different literature work related to this project that has been done in the past,third part describes about data set i.e exploratory data analysis and inferences from that are describe in fourth part observations,fifth part explains about data preprocessing and feature extraction,sixth part describes different machine learning algorithms that are used and presents comparison between these models,in section seven we conclude.

II. Related work

The problem of text classification is a classical problem that has been explored and worked upon extensively in past years.Which got branched into different applications such as article classification ,question classification that we want to do etc.But the inherent problem is always text classification.This survey [1] talks about all the methods and techniques that are currently used in this field such as bow(bag of words),TF-IDF,word2vec etc preprocessing techniques,also about models both machine learning and deep learning and also where text classification is used such as recommender systems,Document summarization and how it can be beneficial.

Now coming to the sub-problem of text classification i.e question type classification which is our problem in hand,in our case we had only two classes.Question type classification is majorly used as a subsystem in Q and A system,used in different online platforms etc.[2] They used dependency structure which describes the syntactic relationships between words in sentences shows an improvement when compared with traditional models such as bag words,wordnet,bi gram.Question classification systems are primarily used as components of question answering (QA) systems[3] presents statistical point of view of question classification they used SVM for building the question classification system and measured its correctness using TREC QA.

Text classification is under the domain of natural language processing for which more sophisticated techniques are developed by leveraging deep learning algorithms which out performs classical machine learning algorithms.GPT,GPT2,BERT[4] models developed by openAI which is currently considered to be best in performing NLP tasks which includes text classification.[5] presents how logistic regression,deep learning algorithms such as Convolutional neural networks,seqtoseq models performs in the task of question type classification.

III. DATABASE

The dataset used in this project is the part of Quora Insincere Question Classification's dataset of questions asked on Qoura. This dataset was hostes on Kaggle. Each data point in training dataset consist of 3 columns, namely qid, question-text(question asked on Quora) and target. The target column has binary values one and zero. Where one represents that the question is insincere an d zero represents question is sincere. The labeling is done by humans. There are 7,83,673 data points in the training data and primary test data consist of 5,22,449 data points for which labels are not provided. Given the categories and the data, it's understandable that the problem is a binary classification problem. Binary classification means data point either belongs to one class otherwise to the other class.

IV. OBSERVATIONS

Before getting into preprocessing and feature extraction it is very important to get to know the distribution of data in order to get better insights while feature selection. We are presenting a few of those here.

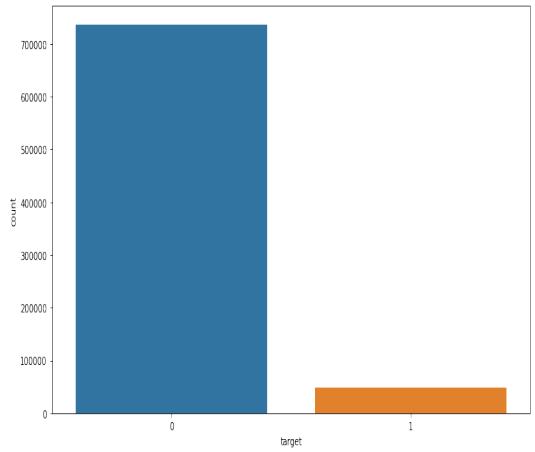

Fig. 1. Count plot on target feature

It can be observed from above diagram the number of sincere questions are more as compared to to insincere question. This kind of imbalanced data is expected in case of natural language processing classification problem.

Out of total training data 6.2 percent of data is of insincere Questions and remaining 93.8 percent of data is of sincere questions. This can be seen in Fig. 2 Pie Plot.

We have plotted two different word clouds for sincere questions and insincere questions. Word cloud show most commonly used in two categories of Quora questions.

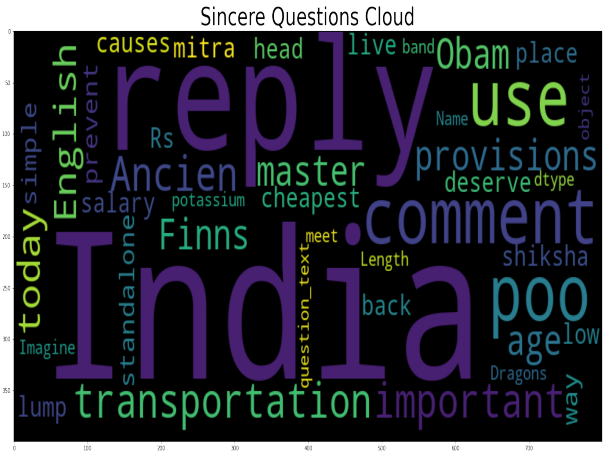

Fig. 2. Word cloud of sincere questions

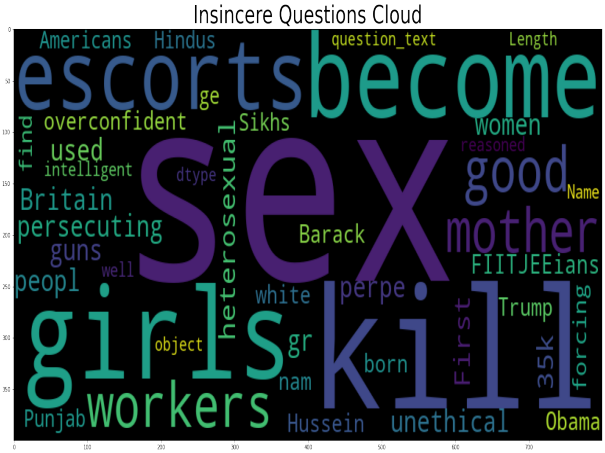

Fig. 3. Word cloud of insincere questions

Some of the most commonly used words in sincere questions are reply, India, comment, use, etc. And similarly most commonly used words in insincere questions are sex, girls, kill, escorts, etc.

We have also plotted graph for number of words used in questions, number of character used in questions and number of stop words in English used in question present in data set.

By observing figure 5 and figure 6 we can conclude that average number of words, characters and English stop words are more in insincere questions as compared to sincere questions.

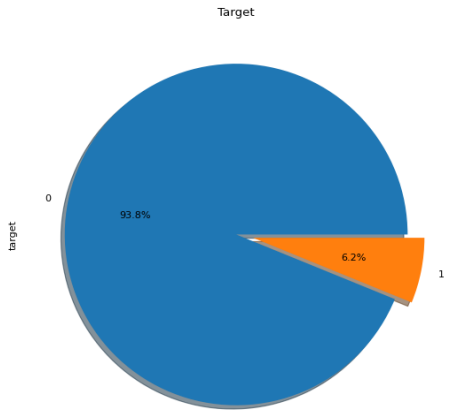

Fig. 4. Pie plot for target feature

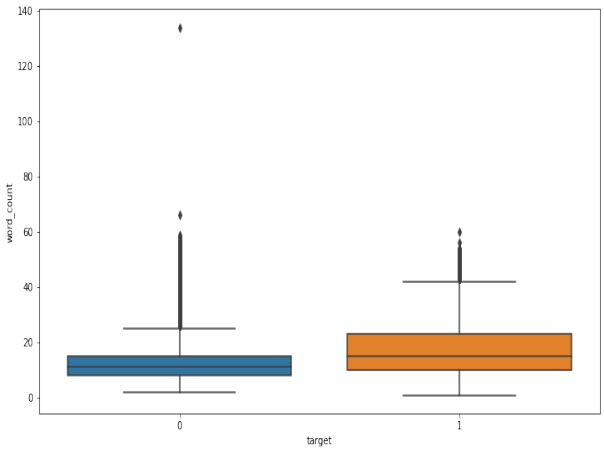

Fig. 5. Box plot for number of words

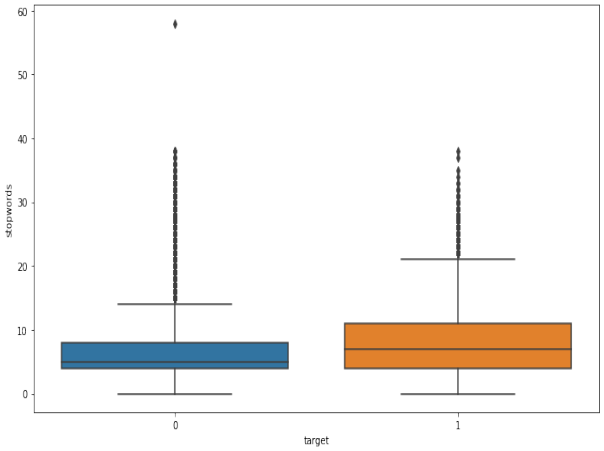

Fig. 6. Box plot for number of stop words

V. Data Preprocessin and Feature Extraction

As our data contains question text which is in a raw text format so, to use and apply mathematical algorithms i.e machine learning algorithms we need to convert this data into a machine understandable format,also we tried to clean the data and apply some standard NLP preprocessing steps.We also found out that too much preprocessing didn't show any improvement in fact it decreased our score.So,we had to find a perfect balance between amount of preprocessing and accuracy of our prediction.

We used various text processing packages such as nltk which is natural langauge processing package in python,re for regular expression matching.First we did some basic text cleaning such as removing html tags i.e removing links from the text,removing inbuilt characters such as newline,tab etc,removing punctuation like " . "," ! " etc. and removing numbers from the text.In text we had lot of contractions like "aren't" "can't","I'am" etc which are casual way of writing real words by merging them together,we removed these contractions by constructing mapping between contracted string to its correct counterpart for example "aren't" becomes "are not","I'am" becomes "I am","we'd've" becomes "we would have" and many others.Upto this point data is cleaned but it is still in text format,so we used TF-IDF Vectorizer which converts the text data into numerical vectors in our context each question we will be converted to a vector. TF-IDF is a measure that reflects how much important a word is in a collection of document(in our case it is a sentence and term means a word).TF-IDF stands for term frequency inverse document freqeuncy which is calculated in following way:-

- Calculate TF Term frequency which signifies how frequent a term appears in a document.
- Calculate IDF Inverse document frequency by taking logarithm of division of total number of documents divided by number of documents in which the term occurred.Which signifies how important a term is.
- Iterate over each document and find both TF and IDF values.
- Calculate TF-IDF score by multiplying TF and IDF [6]

Most of the information about a word have been extracted by TF-IDF but one more operation is done to inject more information into the final training and testing data known as NB-log count ratio matrix multiplication.This NB-log count matrix multiplied with TF-IDF matrix and then a discriminative calssifier is used to perform prediction tasks[7] they use svm and called it NBVM .NB-log count ratio matrix can be calculated as follows:-

- $p = alpha + \sum_{i;y(i)=1} f^i$
- $q = alpha + \sum_{i;y(i)=0} f^i$
- $norm_p = p/c_p norm_q = q/c_q$
- $m = log(norm_p/norm_q)$

- $train = m * train_{tf}$
  $test = m * test_{tf}$

f are the questions in vectorized form that we got from TF-IDF,p matrix represents the summation of all the vectors that corresponds to the class "1" and same for q but for class "0",a smoothing factor alpha(=1) is also added,both p and q are normalized and finally m is the log count ratio which is multiplied with TF-IDF matrices of train and test data to get the final train and test data.

Above describes the preprocessing pipeline that got the highest score.The shape of different training data matrices used was (783673, 1363855) with n-gram range (1,4),(783673, 49549) with n gram (1,1) , (783673, 589377) with n gram (1,4) with stemming and removing stop words.We experimented with Many different additional things such as spelling corrections,for this we used glove embedding[8] from which we found the words which are not in the vocabulary and sorted them according to their frequency,then corrected the top 50 mistakes.We also tried stop words removal i.e removing most common words like "a","the" etc but this gave us less score as compared to without removing stop words but if we look at this closely this justifies the figure 6 as orange box plot represents insincere questions which has more stop words than blue box plot of sincere questions,which helps to arrive at conclusion that stop words give some information about whether the question is sincere or insincere,we tried stemming i.e generating root words from inflected words but this gave us our second best score.

VI. MODEL SELECTION

Different model scores can be seen in Table I.We used different binary classifiers as task in hand is binary classification.For setting the base line we didn't do any data cleaning just applied logistic regression which gave us the score of 0.601.After this we applied all the preprocessing steps that are mentioned above and again used logistic regression but this time with stratified k fold in which data is divided in k folds(we used k=20) and in each fold the ratio of the number of classes is maintained as it was in the original data set and predictions on the test data is averaged over number of folds,from this we achieved best score model LR[2] .When we did stemming we got model LR[1] which is second best.

We also used voting classifier[9] which is a ensembling technique in which different models can be combined using voting i.e each model in ensemble gives its vote in the form of prediction,for getting the final prediction there are sevral methods such as averaging,weighting but we used hard voting i.e majority voting as the name suggest the class that received highest number of votes will be considered as final prediction.We used three models in ensemble sgd(Stochastic gradient descent classifier),linearSVC and logistic regression and we got the model VC [3].We also tried Stacking S[6] and blending B[7] which are also ensembling techniques,In both of these we divide the data into k folds and on each fold base models are trained and the prediction of these again used as a data set on which a meta model(just another machine leaning model) is trained ,both of these work on the assumption that the predictions of the meta-model on the test set will be better than the predictions of any of the 3 models alone on the test set.Only difference between stacking and blending is in blending meta model is applied on the hold out set,also there is no chance of information leak in blending[10].The model that are use in both the ensembles were logistic regression(as meta model),sgd(Stochastic gradient descent classifier),Random forest.

TABLE I
MODELS AND SCORES

Models	***Kaggle Score***
LR[1]	0.63525
LR[2]	0.63633
VC[3]	0.59890
XG[4]	0.56486
Lg[5]	0.54102
S[6]	0.57828
B[7]	0.44632
Final	**0.63633**

[1] Logistic Regression with stemming
[2] Logistic Regression with stratified k fold (k=20)
[3] Voting classifier ensemble of sgd,logistic regression,linearsvc
[4] XGboost with sampling
[5] Lightgbm with grid search
[6] Stacking with Random forest,logistic regression(meta) and sgd
[7] Blending with Random forest,logistic regression(meta) and sgd

Now we tried tree based boosting algorithms which are XGboost[11],lightgbm[12].These these models are applied on data in which stop words are removed and stemming is also done because they were performing well with this setup and also These models were overfiting because of imbalance data set can be seen in figure 1,so to overcome this we used RandomOversampler of imblearn package[15,16] to perform minority class over sampling.Major challenge that we faced during the training of these models was parameter tuning as it is computationally expensive and we had limited processing power so we have to constraint ourselves with tuning less number of hyper parameters at a time.With XGboost XG [4](extreme gradient boosting) using exhaustive search for parameters was taking a lot of time so we employed another hyper optimization method called hyperopt[13],it is different from grid search a brute force approach and random search a purely random approach because it combines randomness and posterior probability distribution in searching for the optimal parameters by approximating the target function using Gaussian process.Hyperopt currently offers two optimization methods random search and tree of parzen estimators(TPE) which is a Bayesian approach.We used TPE as optimization method[14].

Now,for lightgbm(light gradient boosting machine) which is

fast,uses less memory and easily able to handle large scale data but there is trade off between speed and accuracy as a result we observed it gave less score compared to XGboost.With lightgbm Lg[5] we used gridsearchcv which is a exhaustive parameter search technique which tries out every possible combinations and select the best one

VII. CONCLUSION

We would like to conclude that we were able to come up with an efficient model which is logistic regression to classify a given quora question as sincere or insincere.

CHALLENGES AND FUTURE SCOPE

Major challenges that we faced were related to computing power i.e we have to limit ourselves while tunning paramters,doing some preprocessing,We learned a lot about how to preprocess data in a NLP problem which is still a long and challenging process as we had to try out different combinations to know what works best for us. Future scope of the project would be to explore the technique of spelling correction as we briefly worked on it due to time constraints.Also we would like to explore deep learning algorithms because they could capture more complex modalities of the given classification problem as compared to what we used i.e machine learning algorithms.

ACKNOWLEDGEMENT

We would like to thank all our professors G. Srinivas Raghavan and Neelam sinha and all TA's for giving us opportunity to work on this project,we would like to thank our project guide Nikhil sai for constantly guiding us whenever we were stuck,specially those discussions after the presentation were really helpful.We would also like to thank our fellow competitors(teams) for providing such a exciting and healthy competition on kaggle leader board which pushed us to always work hard and look for improvements.

REFERENCES

[1] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, Donald E. Brown, "Text Classification Algorithms: A Survey",DOI: 10.3390/info10040150

[2] LI Xin,HUANG Xuan-Jing,WU Li-de "Question Classification using Multiple Classifiers",https://www.aclweb.org/anthology/I05-4009.pdf

[3] Donald Metzler,W. Bruce Croft "Analysis of Statistical Question Classification for Fact-based Questions",http://ciir.cs.umass.edu/pubfiles/ir-323.pdf

[4] Alec Radford,Jeffrey Wu, Rewon Child,David Luan , Dario Amodei ,Ilya Sutskever "Language Models are Unsupervised Multitask Learners",https://openai.com/blog/better-language-models/.

[5] Tamirlan Seidakhmetov,"Question Type Classification Methods Comparison",https://arxiv.org/pdf/2001.00571.pdf

[6] https://programminghistorian.org/en/lessons/analyzing-documents-with-tfidf

[7] Sida Wang and Christopher D. Manning "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification",https://dl.acm.org/doi/epdf/10.5555/2390665.2390688.

[8] Globe embedding https://nlp.stanford.edu/projects/glove/ .

[9] https://stackabuse.com/ensemble-voting-classification-in-python-with-scikit-learn/

[10] Medium Article bby Steven yu stacking-and-blending-intuitive-explanation-of-advanced-ensemble-methods

[11] Tianqi Chen,Carlos Guestrin "XGBoost: A Scalable Tree BoostingSystem"[online] Available:https://arxiv.org/pdf/1603.02754.pdf

[12] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y.Liu. Lightgbm: A highly efficient gradient boosting decision tree

[13] Bergstra, James Yamins, Dan Cox, David. (2013). Hyperopt: A Python-Library for Optimizing the Hyperparameters of Machine LearningAlgorithms. 13-19. 10.25080/Majora-8b375195-003.

[14] Wai "An Example of Hyperparameter Optimization on XGBoost, LightGBM and CatBoost using Hyperopt"[online] https://towardsdatascience.com/an-example-of-hyperparameter-optimization-on-xgboost-lightgbm-and-catboost-using-hyperopt-12bc41a271e.

[15] Guillaume Lemaˆıtre,Fernando Nogueira,Christos K. Aridas "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning"[online] Available:https://arxiv.org/pdf/1609.06570.pdf

[16] https://towardsdatascience.com/sampling-techniques-for-extremely-imbalanced-data-part-ii-over-sampling-d61b43bc4879