Quora Insincere Questions Classification

Springboard Data Science Career Track Program Capstone Project # 2

Author : Ashish Mohan SharmaReviewer: Kenneth Publish Date: 07/07/2019

INTRODUCTION

- Quora is a platform that empowers people to learn from each other
- Sincere questions are inquiries about which individuals genuinely want to know an answer or gain information from rather than argue a point or make a statement
- Insincere questions intend to make some sort of a statement and are usually asked not with the intention of receiving a helpful answer/comment. It is often the case that insincere questions target religion, gender, politics, etc. and are constructed in a non-neutral tone, are exaggerated, or use words that attack various groups.
- 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.

PROBLEM STATEMENT

- Internet trolls post negative content on public web forums
- Insincere content can have societal impacts (e.g. presidential election)
- How can we detect insincere questions for community benefit?
- The goal is to handle unethical comments, in order to improve online conversation on Quora.

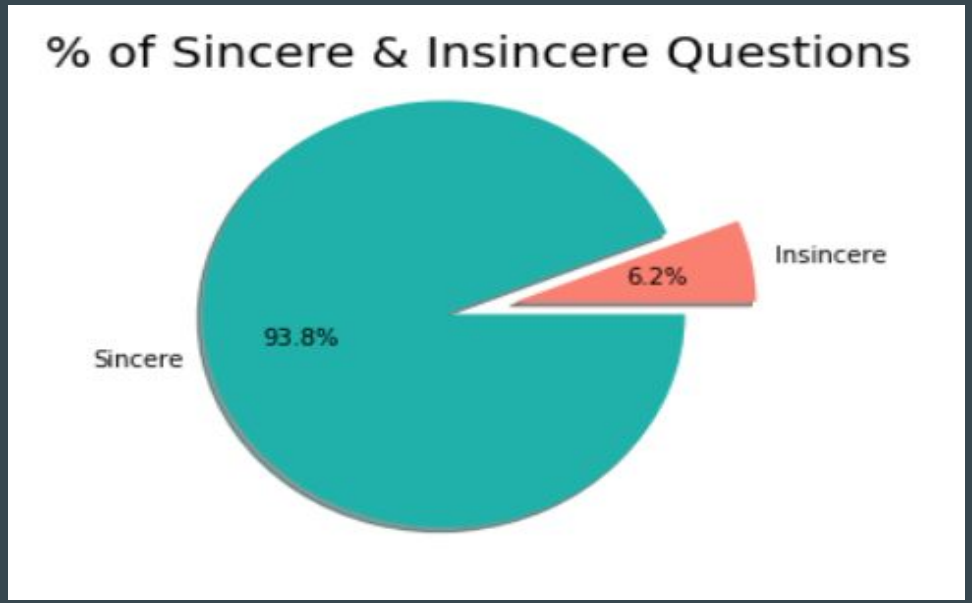
POTENTIAL CLIENTS

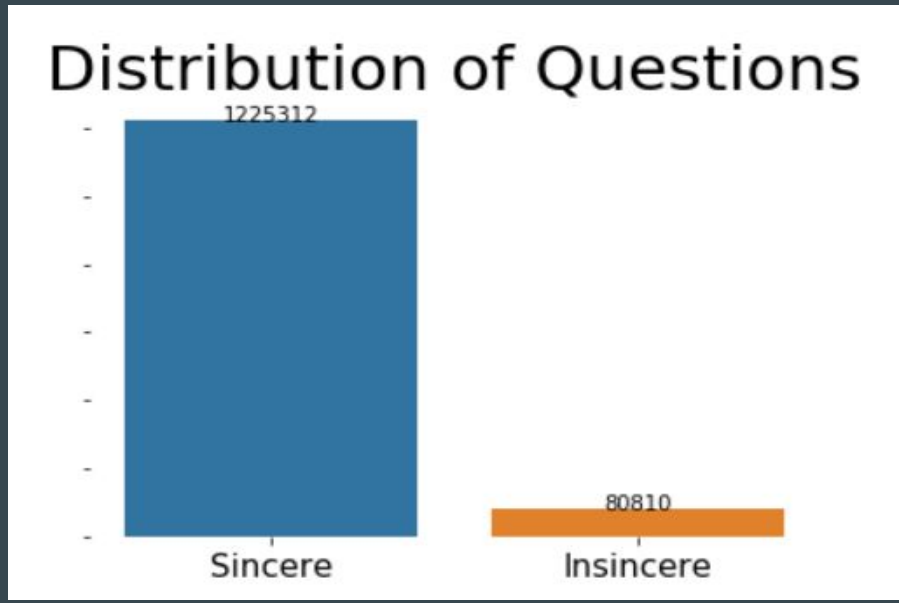
- We may take our model to Quroa and they of course can be our major clientele
- My work can be utilized by any Social Media platform, to prevent unethical comments on their platforms without manual intervention
- Companies like Facebook, Instagram etc can use it to automatically filter out people those who are giving such comments

DATA ACQUISITION

- Data is acquired from an active Kaggle Competition
- The training set consists of a list of questions along with their target values (0 for sincere and 1 for insincere).
- The testing data only contains the sample questions without the target values.
- Word embedding files: Represents a collection of words with their corresponding embed vector, i.e., the number of features that represent similarities between words.
  - Glove
  - Paragram
  - Google
- I understand that reducing unproductive and harmful content online is a challenge for many tech companies including Quora. This motivated me to gain perspective on the process of moderating online speech

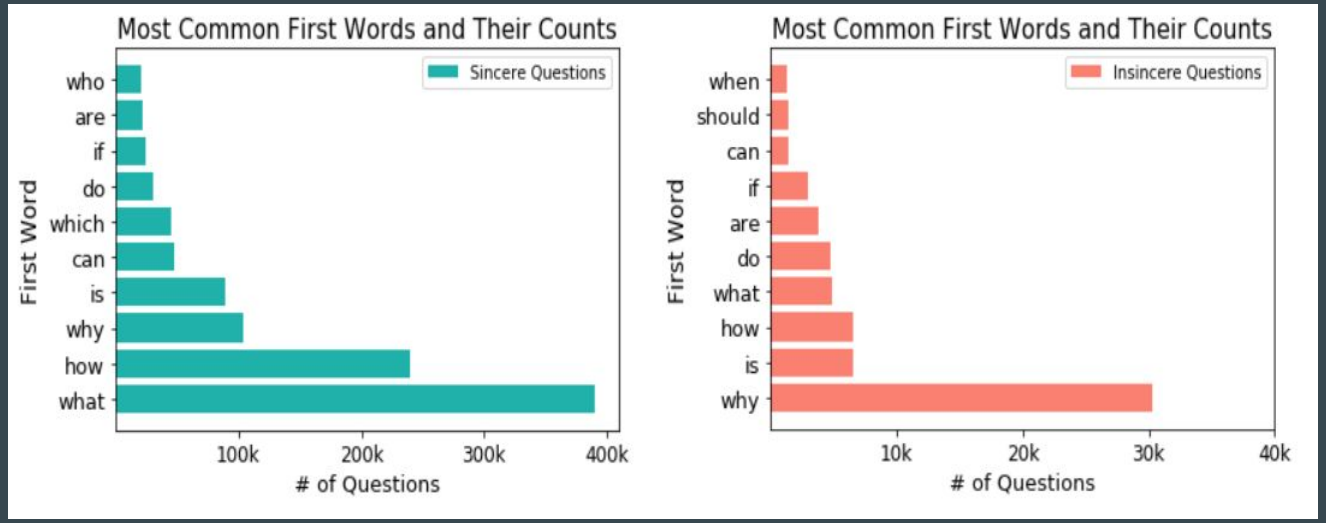
EDA - CLASS IMBALANCE





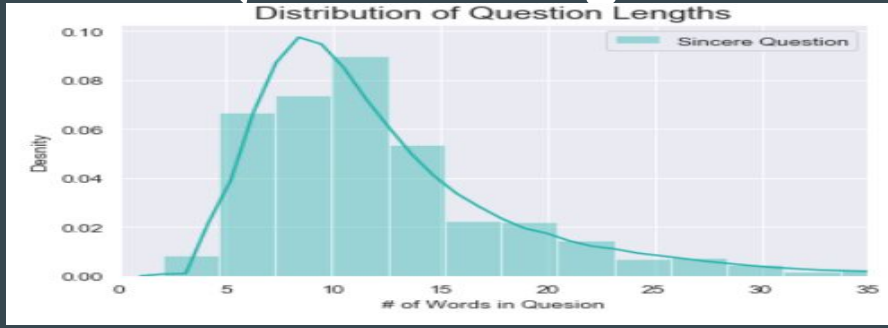
Imbalanced classes are a common problem in machine learning classification where there is a disproportionate ratio of observations in each class. With just 6.6% of our dataset belonging to the target class, we can definitely have an imbalanced class!

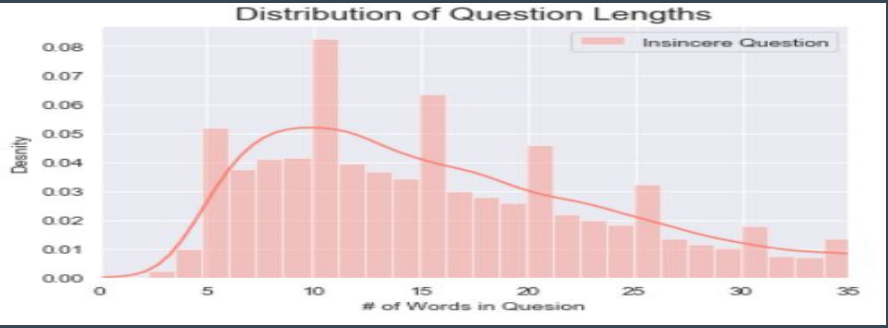
EDA - MOST COMMON FIRST WORDS



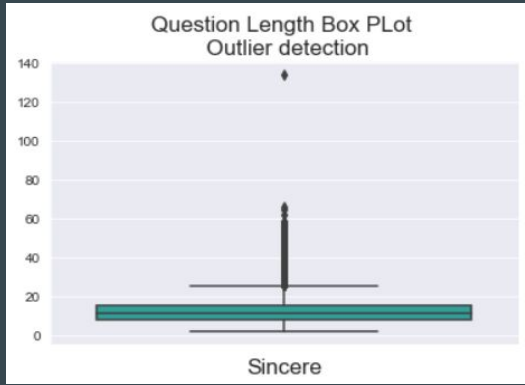
- 10 most frequent words are almost similar like *What, How, Are, Why.*
- There seem to be not much of a difference between the Sincere and Insincere Questions

EDA - Question Length & Outliers Detection





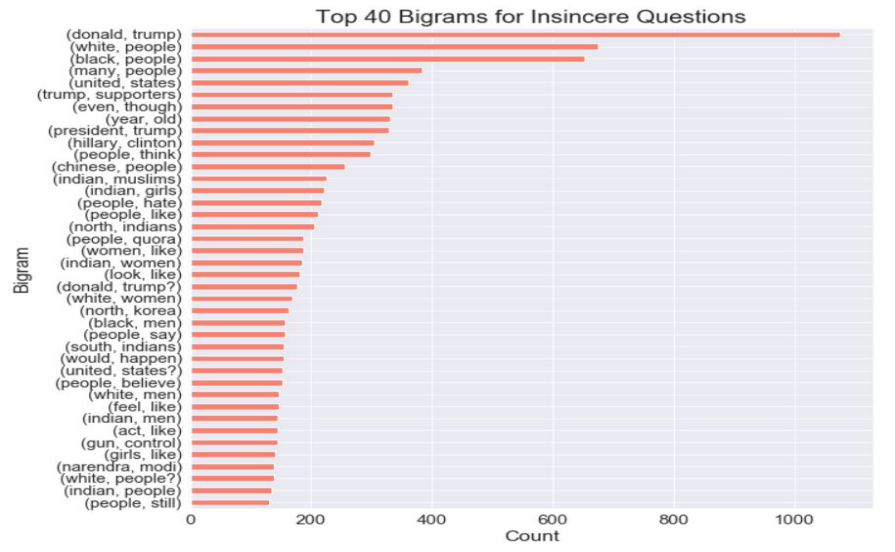
- Insincere questions seem to be longer on an average and have a larger variance on their length.
- Sincere questions seem to clump up around word length as 9 while insincere have a less pronounced peek at around 10 words.
- When an individual is hoping to assert an opinion, it generally takes more words to do this than asking a question

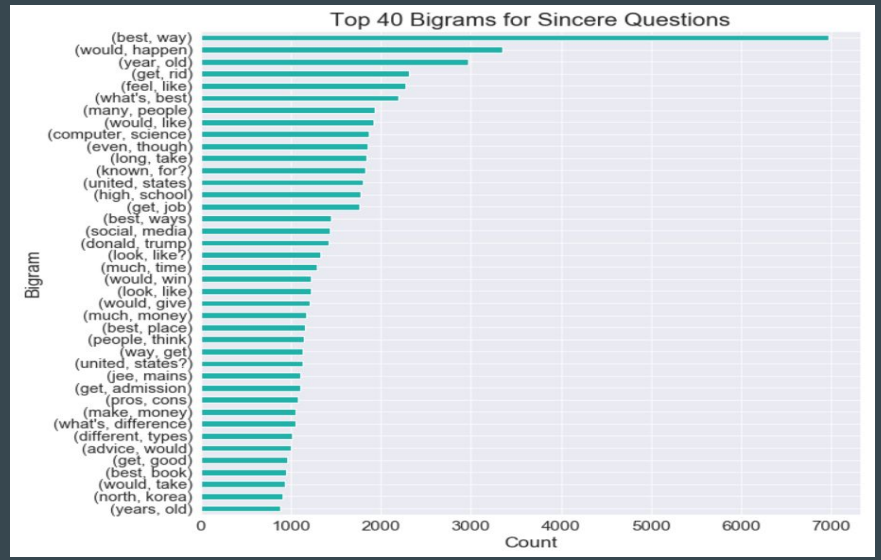


count	1225312
mean	12.5085
std	6.75069
min	2
25%	8
50%	11
75%	15
max	134

count	80810
mean	17.2778
std	6.75069
min	1
25%	10
50%	15
75%	23
max	64

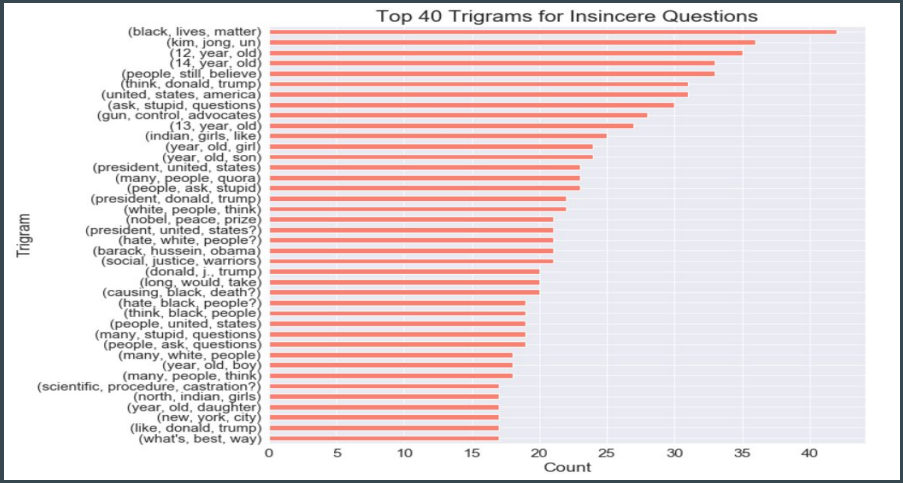
EDA - BIGRAM PLOT

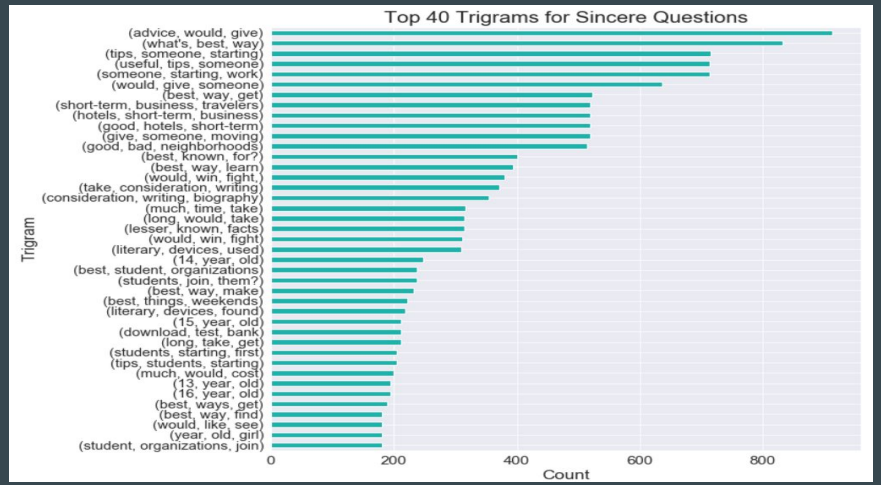




- The plots for top bigrams/trigrams in insincere questions can help us determine what kind of questions Quora would like to limit or ban.
- The top bigram/trigram plots are one way to group insincere questions in a more intelligible format by pinpointing frequent topics of insincere questions
- They can be examined repeatedly to regularly refine Quora's definition of an insincere questions.

EDA - TRIGRAM PLOT

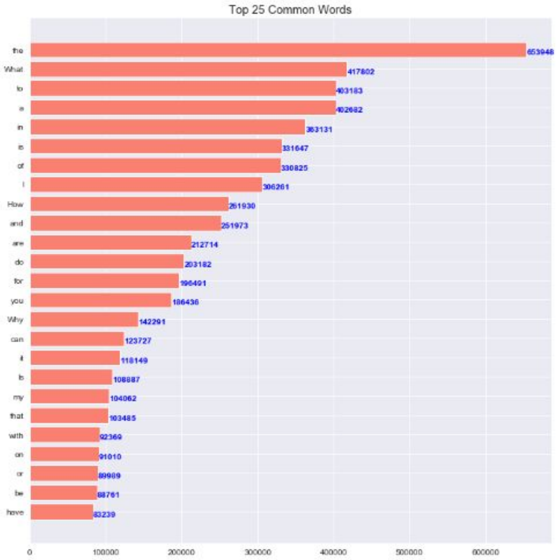


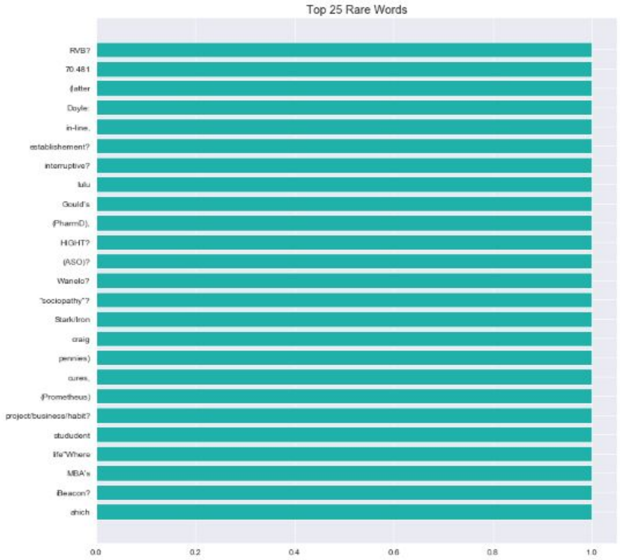


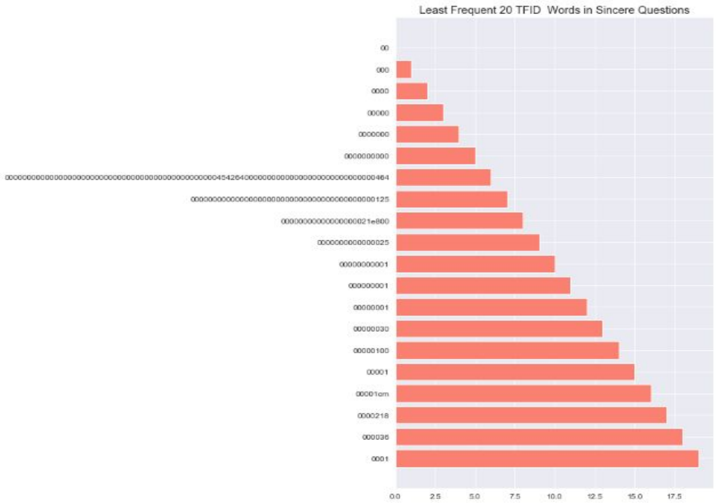
- They can give insight into the level of racism, sexism, and other discriminatory thoughts that are prevalent in the world.
- Grouping of the most frequent topics in anonymously proposed insincere questions can illustrate what some of our deepest and/or darkest thoughts are.
- Anonymity is a key factor to confession since we are often hesitant towards admitting dishonorable thoughts

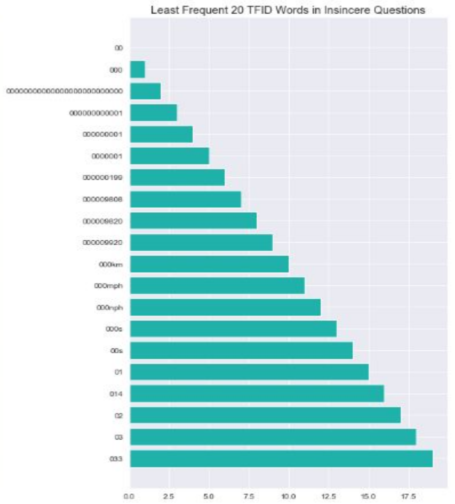
PRE-PROCESSING

- Tokenization
- Stemming/Lemmatization
- Lowering Case
- Remove Numbers
- Remove Punctuation
- Spelling Correction
- Strip whitespace
- Remove Stop Words
- Remove 25 commonly occurring words
- Remove 25 most rare words
- Remove Non English Words

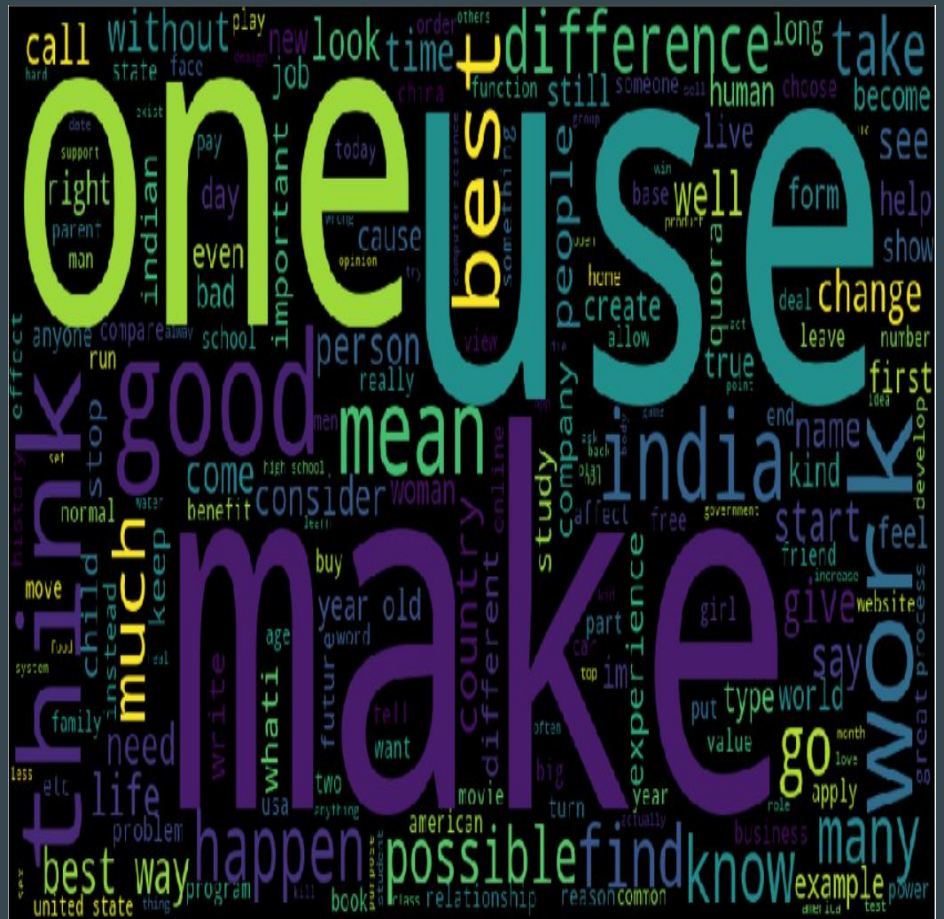








Sincere Word Cloud



InSincere Word Cloud

MACHINE LEARNING

VECTORIZER

- Counting the number of times each word appears in a document.
- Calculating the frequency that each word appears in a document out of all the words in the document.

COUNTVECT-ORIZER

- Ngram_range: n-gram is just a string of n words in a row
- min_df, max_df: The minimum and maximum document frequencies words/n-grams must have to be used as features
- Max_features: Choose the words/features that occur most frequently to be in its' vocabulary and drop everything else

TFIDF

- Term Frequency :

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document}$$

- Inverse Document Frequency:

$$IDF(t) = log_e(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it}$$

- TFIDF:

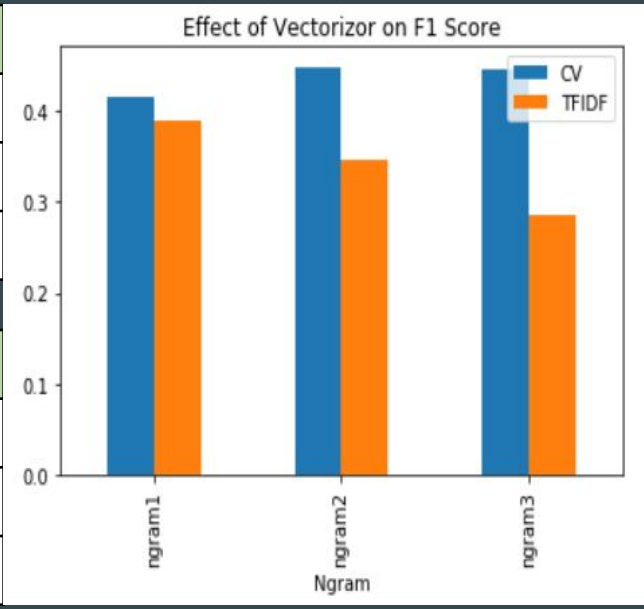
$$TF - IDFscore = TF * IDF$$

CV & LR Classifier

Ngram	Accuracy	Precision	Recall	F_Score	ROC AUC
1,1	0.946	0.642	0.308	0.416	0.914
1,2	0.948	0.651	0.343	0.449	0.917
1,3	0.948	0.657	0.338	0.446	0.916

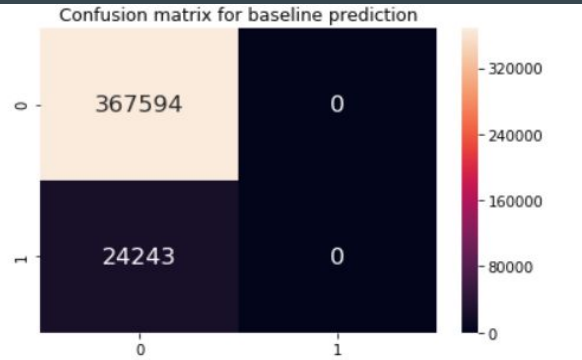
TFIDF & LR Classifier

Ngram	Accuracy	Precision	Recall	F_Score	ROC AUC
1,1	0.946	0.648	0.279	0.390	0.919
1,2	0.945	0.69	0.233	0.346	0.922
1,3	0.943	0.648	0.182	0.284	0.918



BASELINE MODEL

	precision	recall	f1-score	support
0	0.94	1.00	0.97	367594
1	0.00	0.00	0.00	24243
micro avg	0.94	0.94	0.94	391837
macro avg	0.47	0.50	0.48	391837
weighted avg	0.88	0.94	0.91	391837



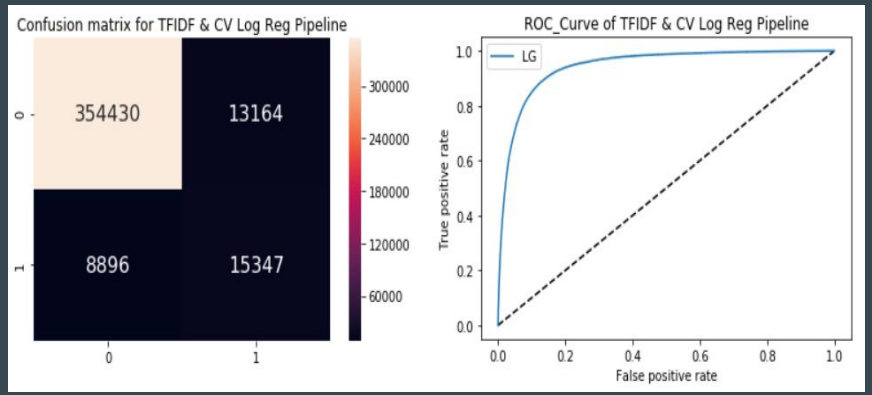
DATA PIPELINE

- Pipelines are a way to streamline a lot of the routine processes, encapsulating little pieces of logic into one function call, which makes it easier to actually do modeling instead just writing a bunch of code.
- Pipelines are set up with the fit/transform/predict functionality, so you can fit a whole pipeline to the training data and transform to the test data, without having to do it individually for each thing you do.
- No need to carry test dataset transformation along with your train features
- Hyperparameter tuning made easy - set new parameters on any estimator in the pipeline, and refit - in 1 line. Or use GridSearchCV on the pipeline

```
pipe1 = Pipeline([
    ('cv', CountVectorizer()),
    ('tfidf', TfidfTransformer()),
    ('logit', LogisticRegression()),
])
```

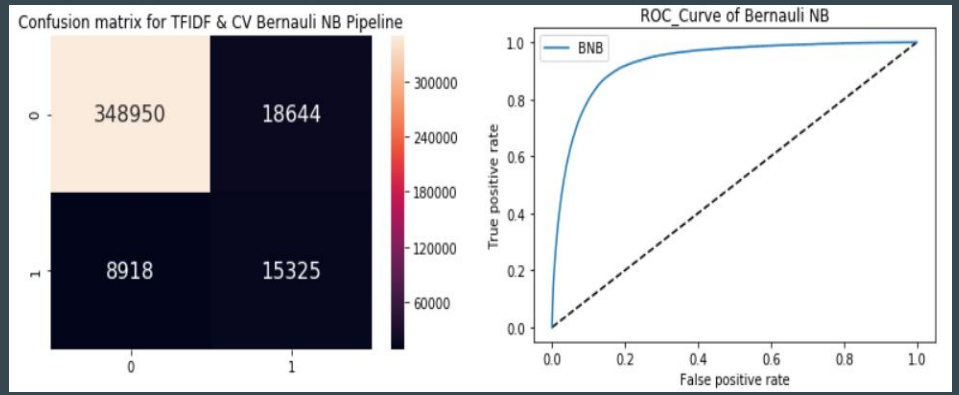
```
hyperparameters = [ 'cv__max_df': [0.9, 0.95,1],
                    'cv__ngram_range': [(1,1), (1,2),(1,3)],
                    'logit__C': [1.0,1.5,2.0,2.5],
                  ]
clf = GridSearchCV(pipe1, hyperparameters, cv=3, n_jobs=-1)
```

CV & TFIDF LR MODEL



Accuracy	Precision	Recall	F_Score	Threshold
0.950	0.538	0.633	0.582	0.237

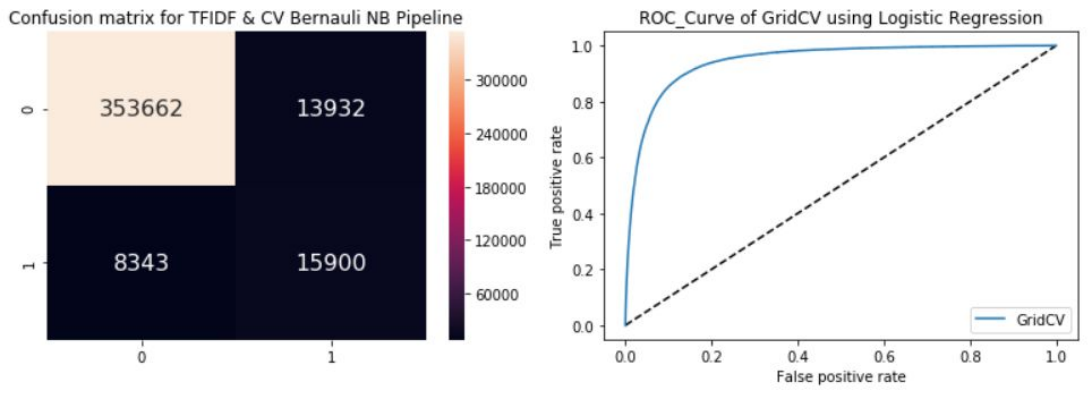
CV & TFIDF BNB MODEL



Accuracy	Precision	Recall	F_Score	Threshold
0.929	0.451	0.632	0.526	0.150

Logistic Regression has outplayed the Naive Bayes classifier. One thing to be noted is using both the vectorizer has considerably increased the F_Score of our prediction.

HYPERPARAMETER TUNING



Accuracy	Precision	Recall	F_Score	Threshold
0.943	0.533	0.656	0.588	0.233

- A probability score below the classification threshold was classified as sincere, while a probability score above the threshold was classified as insincere.
- Adjusting threshold values from the default 0.5 greatly improved F-scores
- Threshold values that maximized F-scores varied between models.

CONVOLUTION NEURAL NETWORK

- Convolutional neural networks are deep artificial neural networks that are used primarily to classify images, cluster them by similarity, and perform object recognition within scenes
- The term convolution refers to the mathematical combination of two functions to produce a third function
- Used pre trained Word Embeddings (GLOVE) to feed to the network to boost performance and dimensionality reduction
  - If two words are similar, they appear in similar contexts
  - Word vectors are computed taking into account the context (surrounding words)
  - Given the two previous observations, similar words should have similar word vectors
  - Using vectors we can derive relationships between words
- Word Coverage percentage and Out of Vocab rate for all the provided Embeddings

Glove

Found embeddings for 87.93% of vocab
Found embeddings for 99.90% of all text
glove oov rate: 0.12371350556605755

Paragram

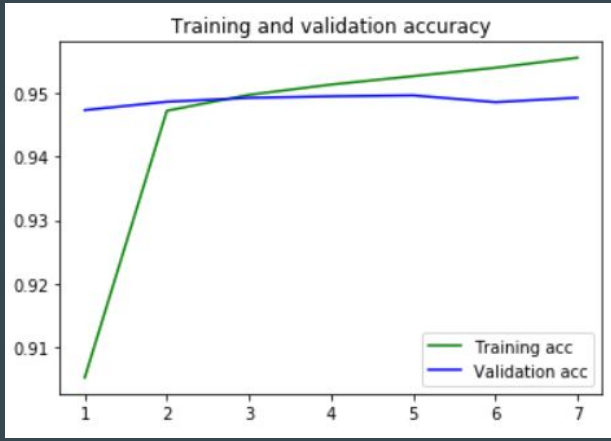
Found embeddings for 87.63% of vocab
Found embeddings for 99.89% of all text
paragram oov rate: 0.12073093887838689

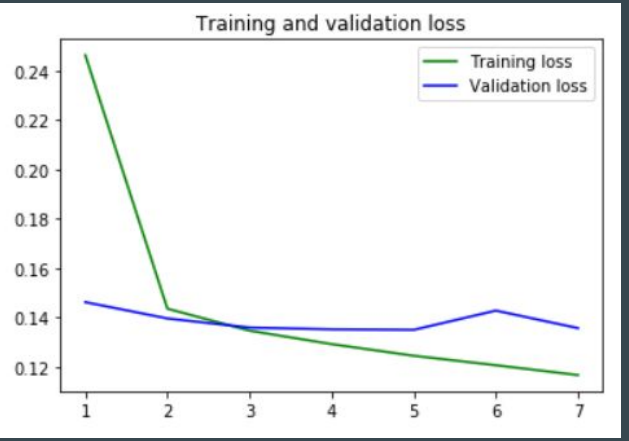
CNN IMPLEMENTATION

- Maximum number of words is 40000
- The embedding matrix will contain the vectors of these words selected from the embedding file if they exist, and empty vectors if not, both of size 300
- The final embedding matrix size will be 50,000 by 300
- Input layer equal to the max length set 70
- the embedding layer are three 1D convolutional layers. The reason why these layers are 1D and not 2D is because we are working with text data and not images
- The last layer is a fully connected layer with one node (binary classification) and a ‘sigmoid’ activation function.
- The loss function, optimizer and evaluation metric defined for this network are ‘binary_crossentropy’, ‘adam’ and F1-score respectively
- Optimization of Number of Epochs

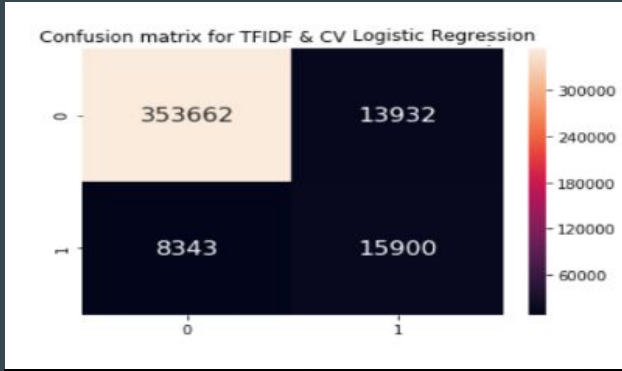
TRAINING & VALIDATION DATASET

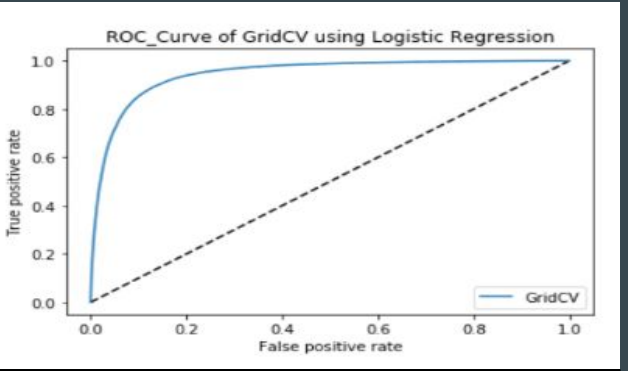
ACCURACY & LOSS VARIATION WITH NUMBER OF EPOCHS





CNN SCORES AND ROC CURVE





Accuracy	Precision	Recall	F_Score	Threshold
0.943	0.633	0.656	0.602	0.230

CONCLUSION & IMPROVEMENT

- As expected CNN model outplayed all other traditional models i.e. Naive Bayes and Logistic Regression
- The final f1 score for this model is a maximum of 0.602 on the validation set which is a good one especially since the dataset tested on has an uneven class distribution.
- Model's predictions have a low percentage of 'False Positives' and 'False Negatives' and a high percentage of 'True Positives'.
- The CNN model is also capable of classifying reliably on unseen data with uneven class distribution.
- For our traditional model, we could have tune more hyperparameter, add more features like length of sentence & number of stopwords or use few more classifiers like SVN or RandomForestClassifier.
- Neural network classifiers can also be improved by expanding the embedding matrix by using additional pre-trained embedding files.
- We can use Recurrent Neural Network (RNN) instead of CNN
- Attack the problem with an ensemble method incorporating the LDA to the Logistic Regression model or including it as another layer in the Neural Network
- Explore how the other embeddings like word2vec or Paragram could improve our model, perhaps using a combination of the four.