San Jose State University

Project Report on,

Disingenuous Question Detection on Reddit

Under guidance of,
Prof. Jorjeta Jetcheva

255 Data Mining
Spring, 2021

:Team 11:

Aditya Inampudi,
Danesh Vijay Dhamejani,
Yusuf Juzar Soni,
Zeel Jayeshkumar Soni

1. Introduction:

Motivation:

As we know, Reddit is an online platform where people ask questions and anyone can answer them. However, some users use such platforms to spread inflammatory questions or spread hate speech. This belies the original purpose of such platforms, which are primarily used to convey genuine information/news to users.

Objective:

To prevent spam users from spreading rumours or falsely influencing people, we have built a data mining algorithm which predicts which question thread is disingenuous.

Approach:

Further analysis of the problem statement revealed that this category of problems could be solved using data mining and machine learning methodologies. The problem essentially involves analyzing the text, understanding its context, and identifying certain words or phrases that would help in detecting whether a particular question is appropriate or inappropriate.
ML makes textual analysis much faster and more efficient than manual processing of texts. It allows to reduce labor costs and speed up the processing of texts without compromising on quality.
Since this is primarily a classification problem, classification algorithms like logistic regression, SVM, KNN, etc. are potential approaches that can be leveraged to obtain the desired results.

Our approach can be summed up as follows:

- Understanding the data
- Identifying potential solutions (Algorithm selection)
- Comparing results to identify optimal solutions

Literature (Market Review):

- As the number of questions increases, the need for cleaning questions also should be automated.
- The quality of the questions should be ensured so that it might increase, engage and attract more users.
- The hate spread among the people should be reduced. As, they might get influenced by the questions and their answers.

2. System Design And Implementation:

Algorithms:

- Logistic Regression:

  Logistic regression is named for the function used at the core of the method, the logistic function. The logistic function, also called the sigmoid function, was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.
  Like all regression approaches logistic regression provides outputs in terms of probabilities. However, logistic regression is intended for binary (two-class) classification problems. It will predict the probability of an instance belonging to the default class, which can be snapped into a 0 or 1 classification.

- Multinomial NB:

  Typically used in text mining and classification. It uses the bag of words approach. It treats the language like it's just a bag full of words and each message is a random handful of them. Naive Bayes is based on Bayes' theorem, where the adjective Naïve says that features in the dataset are mutually independent. Occurrence of one feature does not affect the probability of occurrence of the other feature. Multinomial NB is a supervised learning technique that classifies every new document by assigning one or more class labels from a fixed or predefined class.

- Neural Network:

  The Neural Network consists of 3 layers:
  1. Embedding Layer: Converts the training data into glove vector form.
  2. Bi-directional LSTM: Contains two independent RNN's, in opposite directions.
     Bi-directional GRU: Ensures efficient memory use.
  3. Output Layer: The activation function employed in our Neural Network is Sigmoid Function.

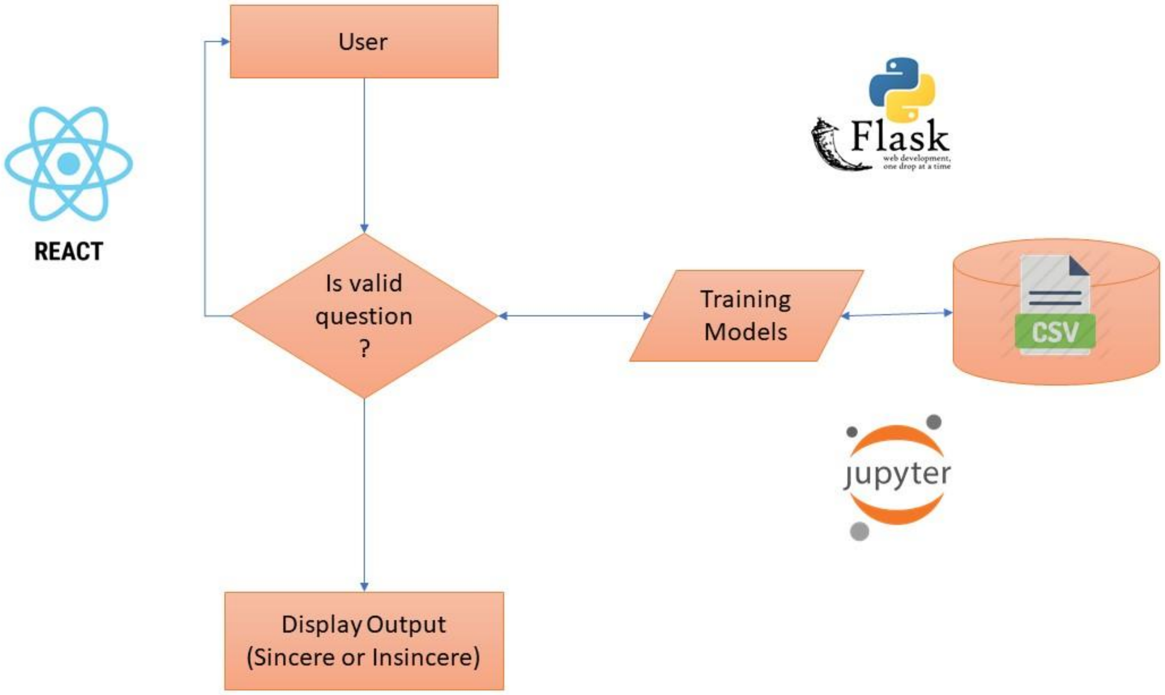
Technologies and tools used:

- Jupyter Notebook/VS Code : IDE used for coding the solution.
- Python was used to implement the algorithms. Various predefined libraries like Sci kit learn, NLTK, Pandas, Numpy etc. were used to perform data preprocessing, feature engineering etc.
- React JS and Flask were used for building the frontend and backend of the GUI/ Web Application respectively.
- Postman was used to test the API.

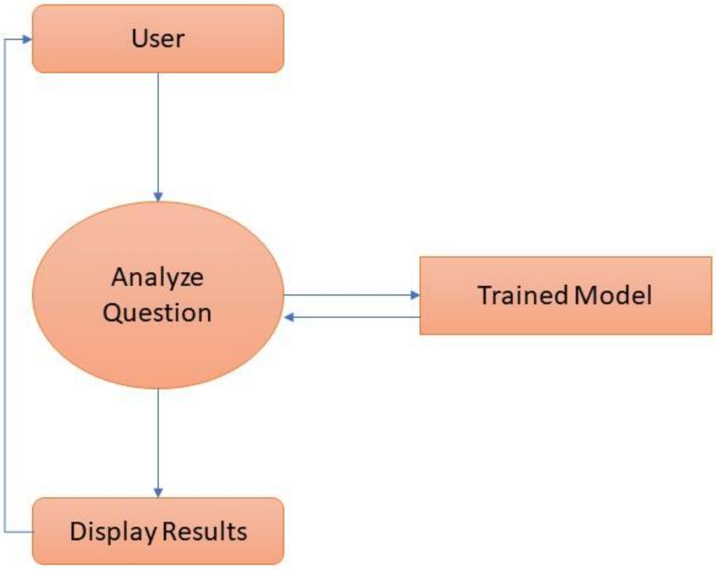
Architecture related decisions:

We were working with datasets in the .csv format. All analysis and prediction related tasks were first run in the jupyter notebook to check whether the predictions were correct. We then used the trained models to perform predictions in the Web Application. The dataset is relatively large with more than 120000 tuples. We decided to choose three approaches to demonstrate our solutions. Although there are more approaches like SVM, and KNN keeping in mind the resource constraints we had we chose three algorithms that best described our findings.

System Design:


User
REACT
Flask
web development, one drop at a time
Is valid question ?
Training Models
CSV
jupyter
Display Output (Sincere or Insincere)

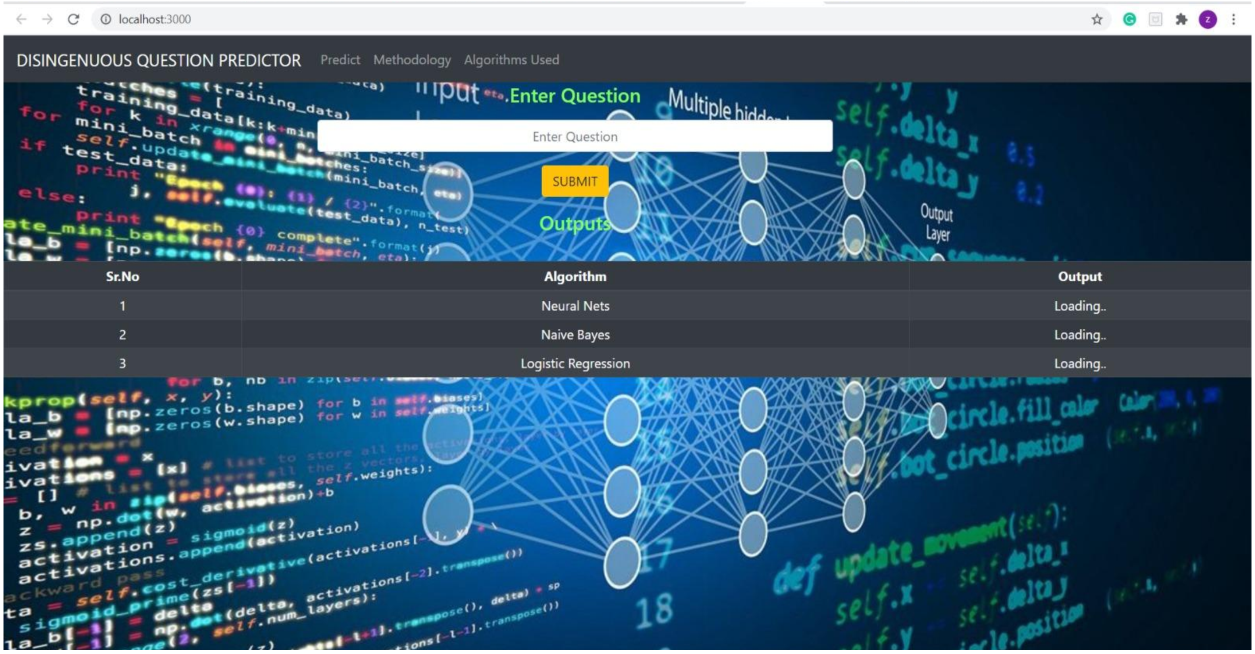
Data Flow Diagram:



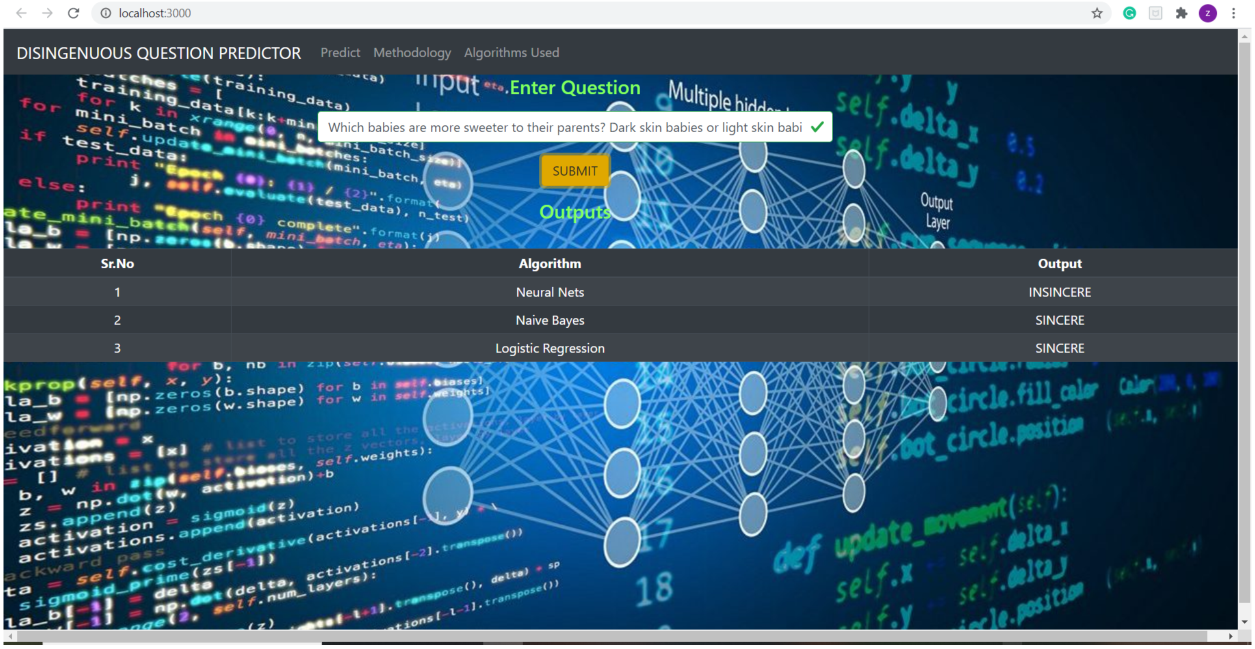
Use Cases:

The user enters a question and receives a response as to whether the given question is sincere or not.

GUI:



Sr.No	Algorithm	Output
1	Neural Nets	Loading..
2	Naive Bayes	Loading..
3	Logistic Regression	Loading..



Which babies are more sweeter to their parents? Dark skin babies or light skin babi

Sr.No	Algorithm	Output
1	Neural Nets	INSINCERE
2	Naive Bayes	SINCERE
3	Logistic Regression	SINCERE

To run flask:

1. `Git clone https://github.com/DaneshDhamejani/Data_Mining.git`
2. Go to ./backend
   `flask run`
3. Open new terminal
4. To run react app:
   Go to ./frontend
   ```
   npm install
   npm run build
   npm start
   ```

3. Experiments/ Proof of Concept Evaluation:

Dataset:

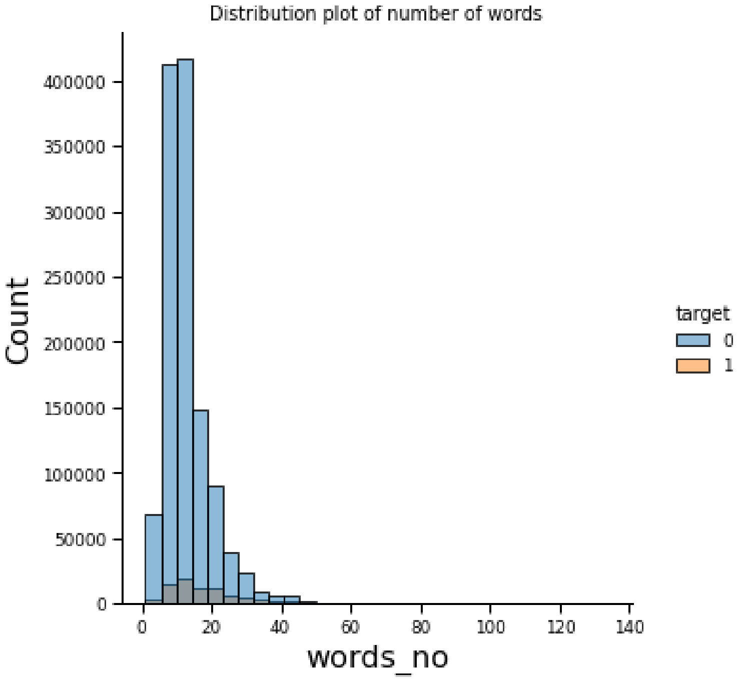
https://github.com/DaneshDhamejani/Data_Mining/blob/main/backend/reddit.csv

Methodology:

- We had collected our dataset from multiple sources.
- We had added new features like number of words, number of unique words, number of stopwords, number of punctuation symbols, number of uppercase words, number of lowercase words for analysis

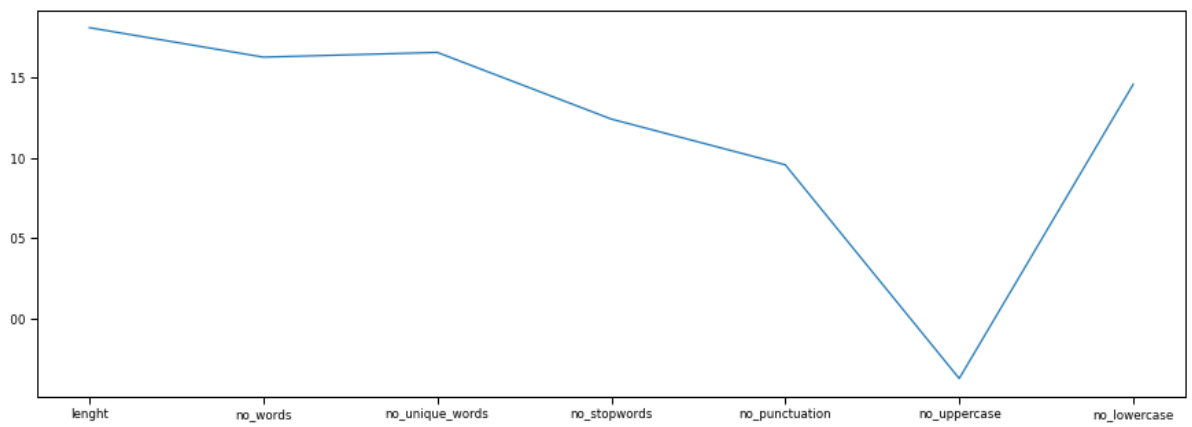
- We had done text preprocessing for removing the stopwords, contractions and punctuation marks.
- We had done lemmatization and stemming to remove the redundancy and get the context of words.
- We trained logistic regression and multinomial naive bayes using both tf-idf and bow vectors text representation and compared the accuracies and f1-scores.
- We trained lstm using glove vector representation.

Graphs and Analysis:



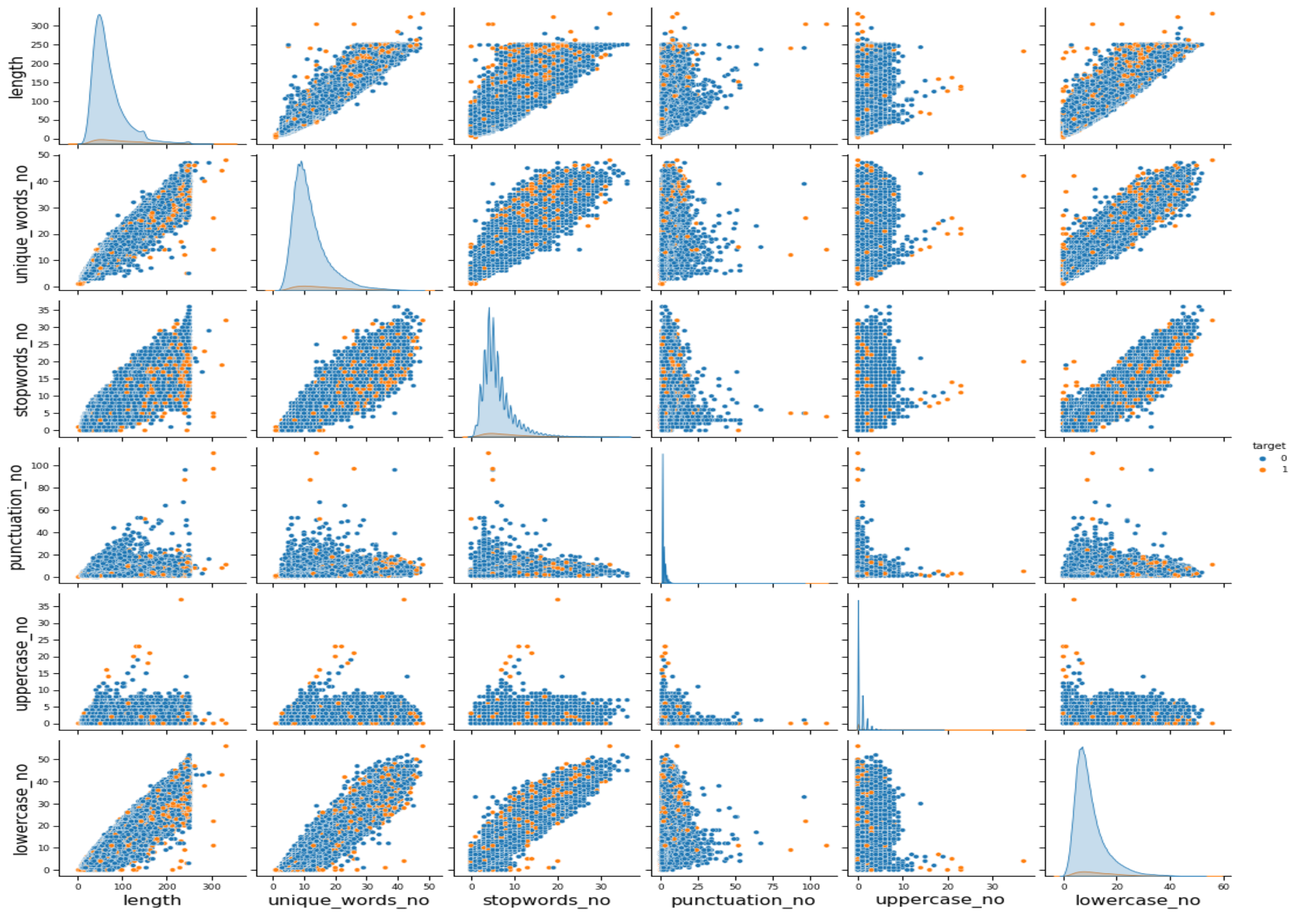
Graph 3.1: Number of Words Comparison

As seen in the graph, both sincere and insincere have the same distribution, but length of insincere questions is much shorter than sincere questions.



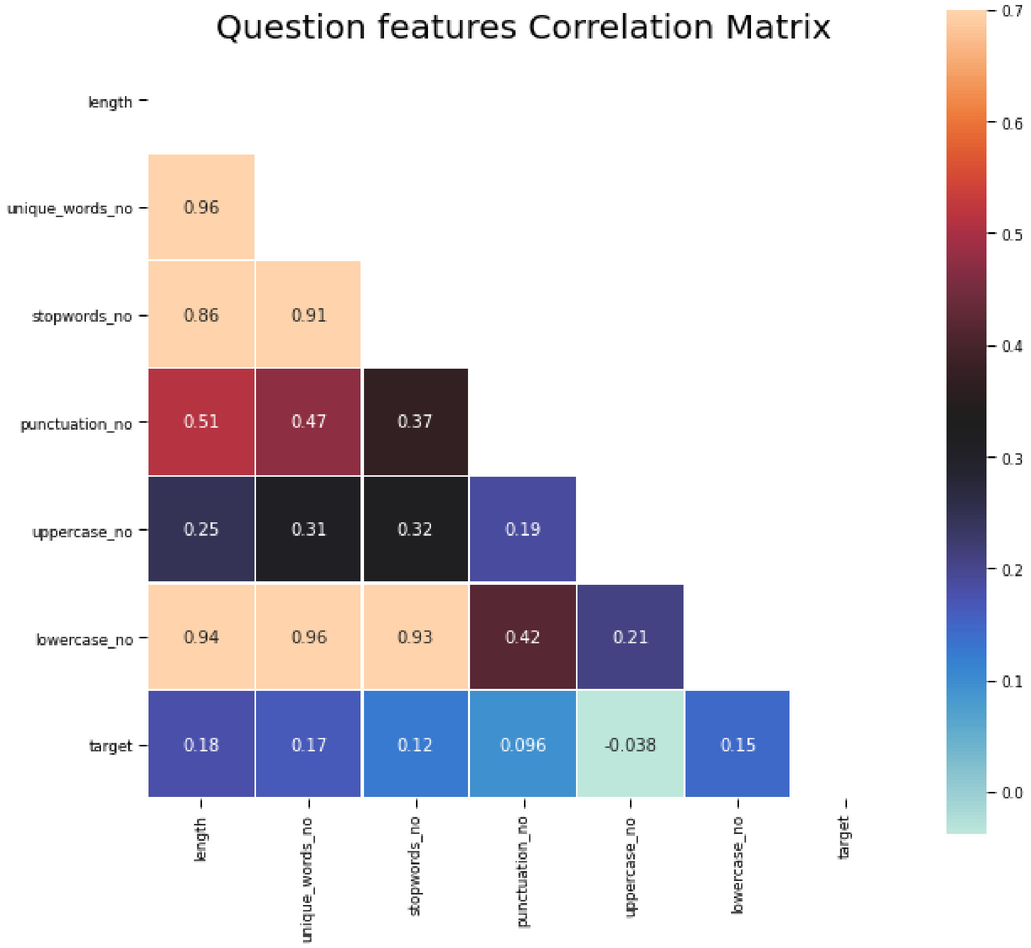
Graph 3.2:Target Correlation with Features

Graph 3.2 illustrates target correlation vs newly added attributes. We can see the length of sentences has the highest correlation, but then too it is 0.18, we can't completely rely upon that. Also, the number of lowercase letters plays a vital role in prediction as seen in the graph.



Graph 3.3:Pair Plot of customized features

A pair plot allows us to see both distribution of single variables and relationships between two variables. Pair plots are a great method to identify trends for follow-up analysis. Here, we can figure out that sincere questions have linear correlation with respect to length of question. We can also see that lowercase letters have linear graphs. Then too no sharp differentiation between two target classes can be found.



Graph 3.4:Heatmap of customized features with respect to length

As seen in graph 3.4, lowercase_no is highly correlated with length of the question. Thus, converting the question to lowercase will give promising results. The number of uppercase words are negatively correlated with target. Thus, we can get to know using uppercase words for prediction is not useful.

Word Cloud:

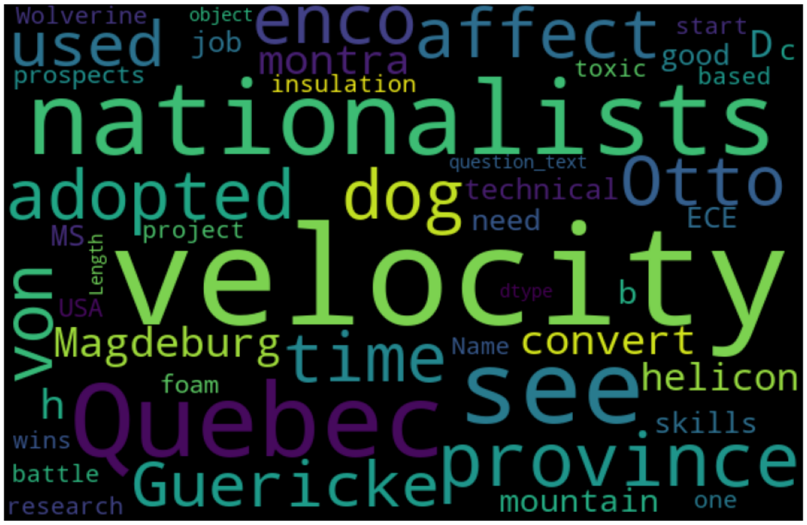


Image 3.5:Sincere

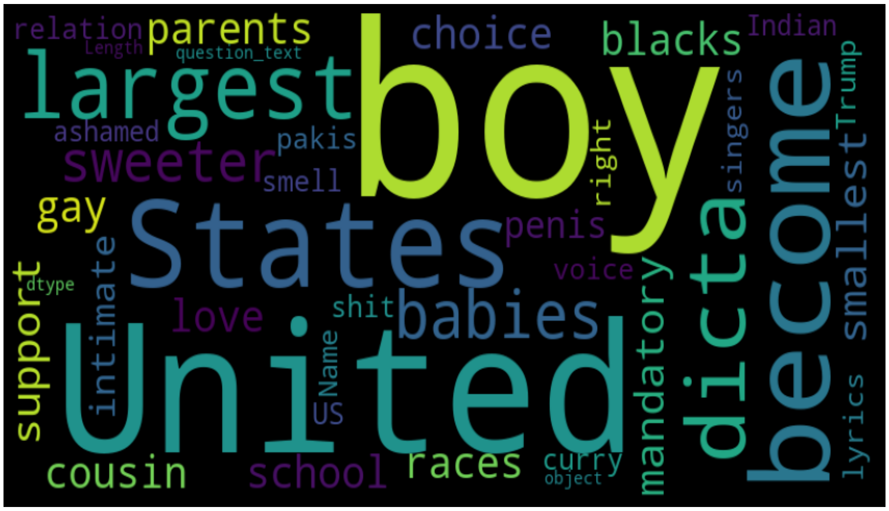


Image 3.6:Insincere

4. Discussion and Conclusion:

Decisions:

The following questions needed to be answered during the development process:

1. Which algorithms were best suited to evaluate the problem statement?
2. Considering the resource constraints, which algorithm is optimal?
3. Is the data enough to produce a relatively sound decision?
4. Trade off between speed and accuracy?
5. Does increase in complexity often produce better results?
6. Which technology stack needed to be used to build the web application?

Difficulties encountered:

1. Analyzing the dataset as size was very huge.
2. Integrating the trained models into our flask backend.
3. Slight modification in API response needed to display data correctly in the frontend.

Insights:

Approach	BOW (Accuracy)	TF-IDF (Accuracy)	Glove (Accuracy)
Multinomial Naive Bayes	94.5	93.8	-
Logistic Regression	94.4	80.3	-
Neural Network	-	-	94

1. The dataset was imbalanced. So, though we got good accuracies using Multinomial Naive Bayes and Logistic Regression models. They got less f1 scores.
2. So, in order to mitigate the imbalanced dataset we trained lstm models with glove vectors.
3. Glove vectors are one way to represent words as numerical text vectors. The text vectors are created from predicted surrounding words by maximizing the probability of a context word occurring given a center word by performing a dynamic logistic regression.
4. We then trained the model using k-fold cross validation and have taken the lstm model which gave good f1-score and accuracy.

Extension:

1. Many more approaches like SVM and KNN can be used to evaluate the problem.
2. More advanced text analysis techniques can be used to understand the nuances of the questions better, to provide more fine grained results.
3. Here we are considering only two classes of questions. We can use more advanced techniques to identify what kind of question it is (which topic does it belong to)

4. Further analysis can be carried out on the same dataset to infer the mood of the people based on the kinds of questions being asked. This in turn can be used by policy makers, law enforcement officials to understand people's sentiments, without any privacy violations.

Conclusion:

As inferred from above, we observe that Neural Network has much better accuracy as compared to Multinomial Naive Bayes and Logistic Regression. Glove vector word representation had better performance as compared to TF-IDF and Bag of Words. For high accuracy Neural Network is useful, on the other hand, for high speed either Multinomial Naive Bayes or Logistic Regression can be used.

5. Task Distribution:

- Data Preprocessing and Cleaning: Zeel Jayeshkumar Soni
- Applying ML Models: Danesh Vijay Dhamejani
- Applying Deep learning Models: Aditya Inampudi
- Frontend: Yusuf Juzar Soni