FLIP ROBO

NAME OF THE PROJECT

Malignant Comments Classifiers

Submitted by:

Robinson P

ACKNOWLEDGMENT

Towards Datascience

Kaggle Articles

INTRODUCTION

- ## Business Problem Framing

Internet comments are bastions of hatred and vitriol. While online anonymity has provided a new outlet for aggression and hate speech, machine learning can be used to fight it. The problem we sought to solve was the tagging of internet comments that are aggressive towards other users. The proliferation of social media enables people to express their opinions widely online. However, at the same time, this has resulted in the emergence of conflict and hate, making online environments uninviting for users. It is Vital for any Organization/Person/Political Party to sense the pulse of people about them which helps them out to come up with policy and strategy

- ## Conceptual Background of the Domain Problem

The proliferation of social media enables people to express their opinions widely online. However, at the same time, this has resulted in the emergence of conflict and hate, making online environments uninviting for users.

- ## Review of Literature

A. Akshith Sagar et.all concluded that with the Internet being a platform accessible to everyone, it is important to make sure that people with different ideas are heard without the fear of any toxic and hateful remarks. And after analyzing various approaches to solve this problem of classification of toxic comments online, it is found that CNN model works slightly better than LSTM and NB-SVM with the accuracy of 98.13%. Future scope for this analysis would be integrating such classification algorithms into social media platforms to automatically classify and censor or toxic comments.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

Evaluation Metrics Selection

During the modeling process, we choose multiple different evaluation metrics to evaluate the performance of models based on the nature of our data:

- Recall
- F Score
- Hamming Loss

Basic Model Comparison

Using Multinomial Naive Bayes as our baseline model, we first used k-fold cross validation and compared the performance of the followingi three models without any hyperparameter tuning: Multinomial Naive Bayes, Logistic Regression, and Linear SVC. Logistic Regression and Linear SVC perform better than Multinomial Naive Bayes.

.

- Data Sources and their formats

  Data Sources are train.csv,test.csv in csv formats

- Data Preprocessing Done

  Since there were no null values data cleaning is simpler

Data Inputs- Logic- Output Relationships

Data here are the comments made by the users in social media to troll/poke/comment an Individual or organization the output data was classified to find the degree of extent the harsh comments were made to a person/company using malignant, highly – Malignant, Abuse, Threat, loathe. This helps to identify vulgar negative comments and track for abusing person

- State the set of assumptions (if any) related to the problem under consideration

It’s a classification problem which needs complete understanding of NLP to come up with strategic solution on finding the ML model so that it can be controlled and restricted from spreading hatred and cyberbullying.

Hardware and Software Requirements and Tools Used

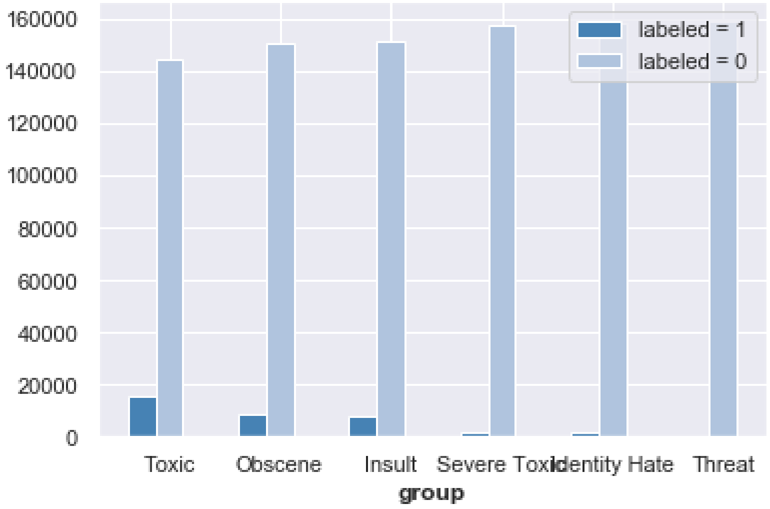
Jupiter notebook, python libraries

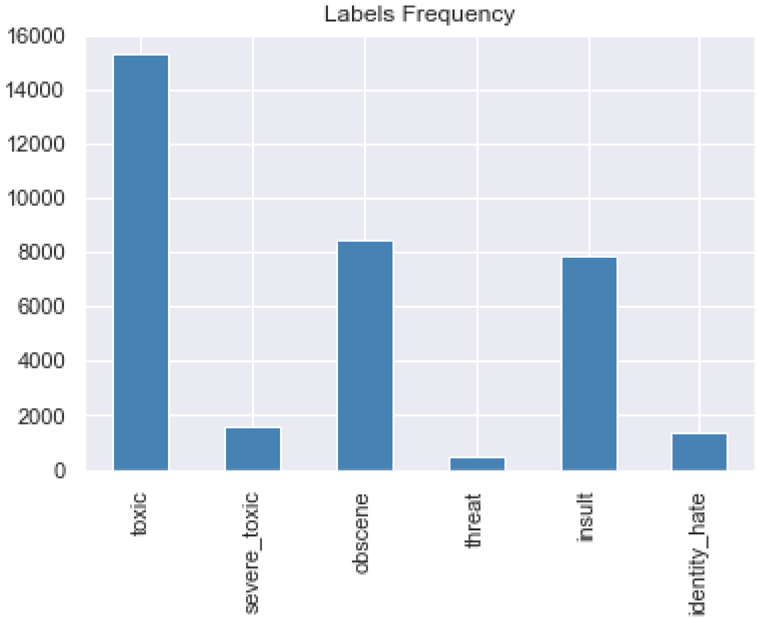
Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

Data Preprocessing and EDA

Since all of data are text comments, I wrote our own `tokenize()` function, removing punctuations and special characters, stemming and/or lemmatizing the comments, and filtering out comments with length below 3. After benchmarking between different vectorizers (TFIDFVectorizer and CountVectorizer), we chose TFIDFVectorizer, which provides us with better performance.





The major concern of the data is that most of the comments are clean (i.e., non-toxic). There are only a few observations in the training data for Labels like `threat`. This indicates thatI need to deal with imbalanced classes later on and indeed,I used different methods, such as resampling, choosing appropriate evaluation metrics, and choosing robust models to address this problem.

Model Fitting

Evaluation Metrics Selection

During the modeling process, we choose multiple different evaluation metrics to evaluate the performance of models based on the nature of our data:

- Recall
- F Score

- Hamming Loss

- ## Testing of Identified Approaches (Algorithms)

  Logistic Regression

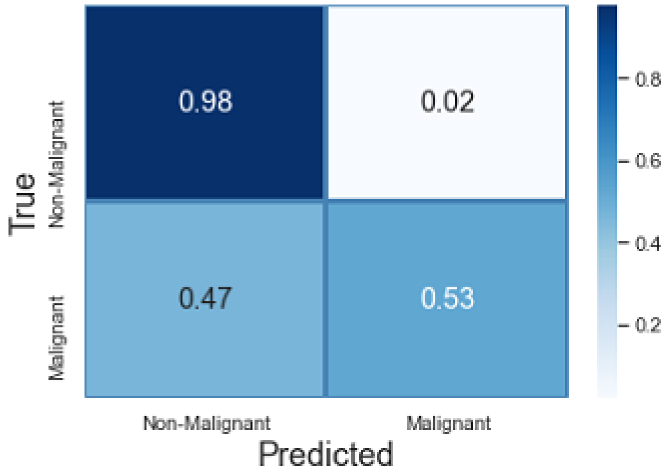
  Linear SVC

- ## Run and Evaluate selected models

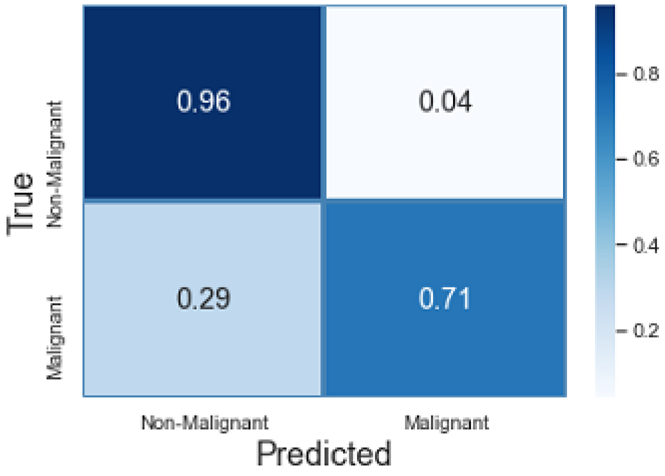
  Below shows the confusion matrix for label toxic. Notice that all models predict Non-toxic labels pretty well because most of our data are non-toxic. However, Multinomial NB tends to predict more toxic comments to non-toxic while Linear SVC is doing a great job in classifying toxic comments.

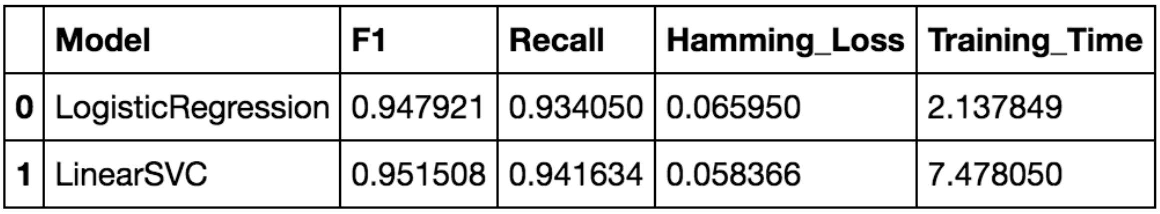
```
Choose a class for the Confusion Matrix: malignant
*************** malignant labelling ***************

****  MultinomialNB  ***
```

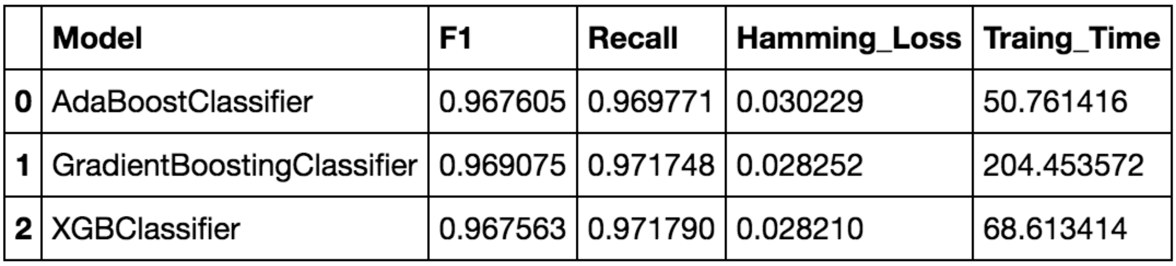


```
****  LogisticRegression  ***
```

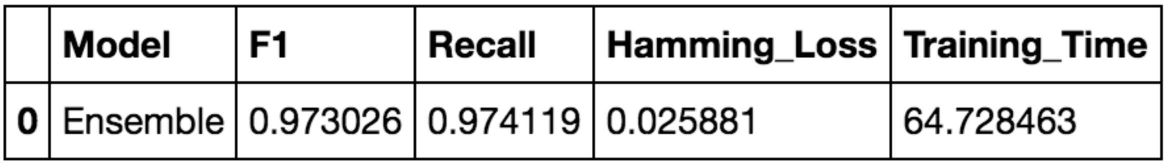




	Model	F1	Recall	Hamming_Loss	Training_Time
0	LogisticRegression	0.947921	0.934050	0.065950	2.137849
1	LinearSVC	0.951508	0.941634	0.058366	7.478050



	Model	F1	Recall	Hamming_Loss	Traing_Time
0	AdaBoostClassifier	0.967605	0.969771	0.030229	50.761416
1	GradientBoostingClassifier	0.969075	0.971748	0.028252	204.453572
2	XGBClassifier	0.967563	0.971790	0.028210	68.613414



	Model	F1	Recall	Hamming_Loss	Training_Time
0	Ensemble	0.973026	0.974119	0.025881	64.728463

- Key Metrics for success in solving problem under consideration
  Confusion metrics –used for identifying true positives and negatives
  Recall-
  F1 Score-
  Hamming Loss-

- Visualizations

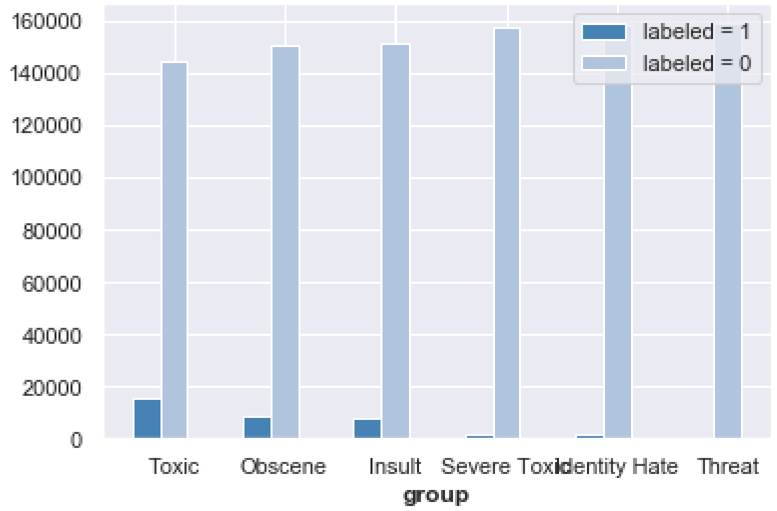
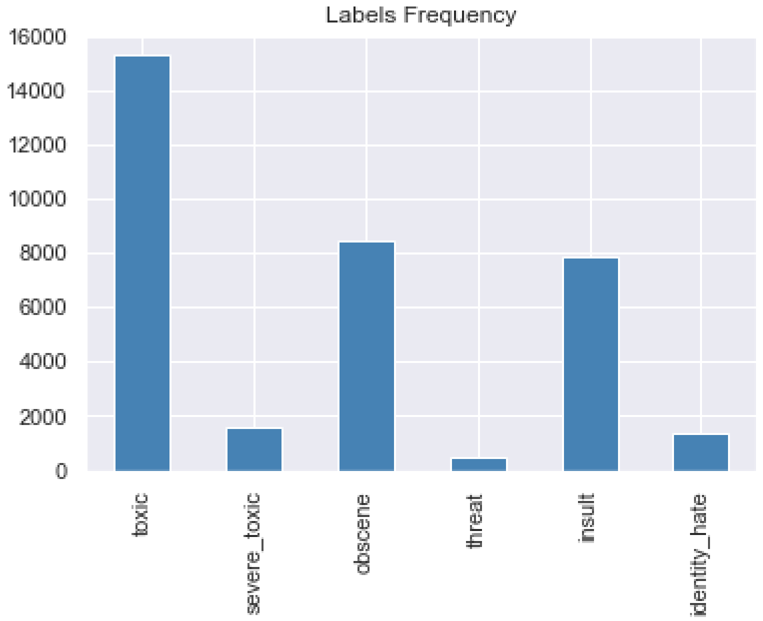
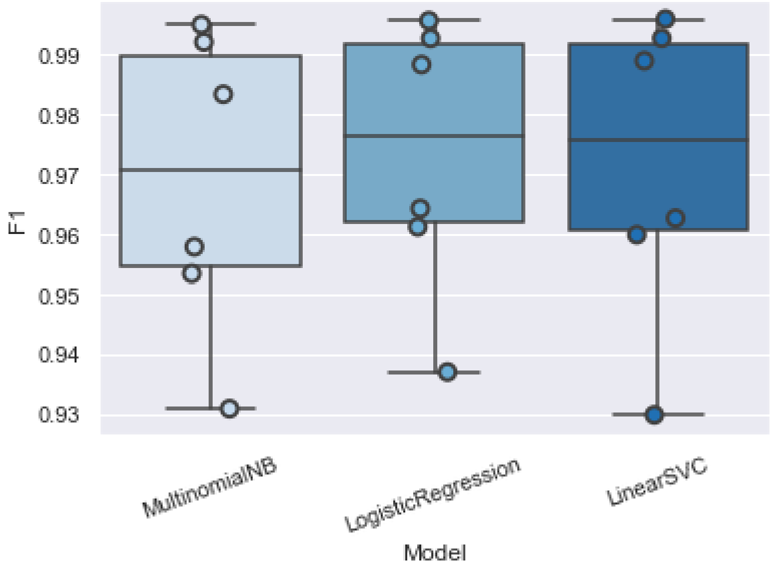
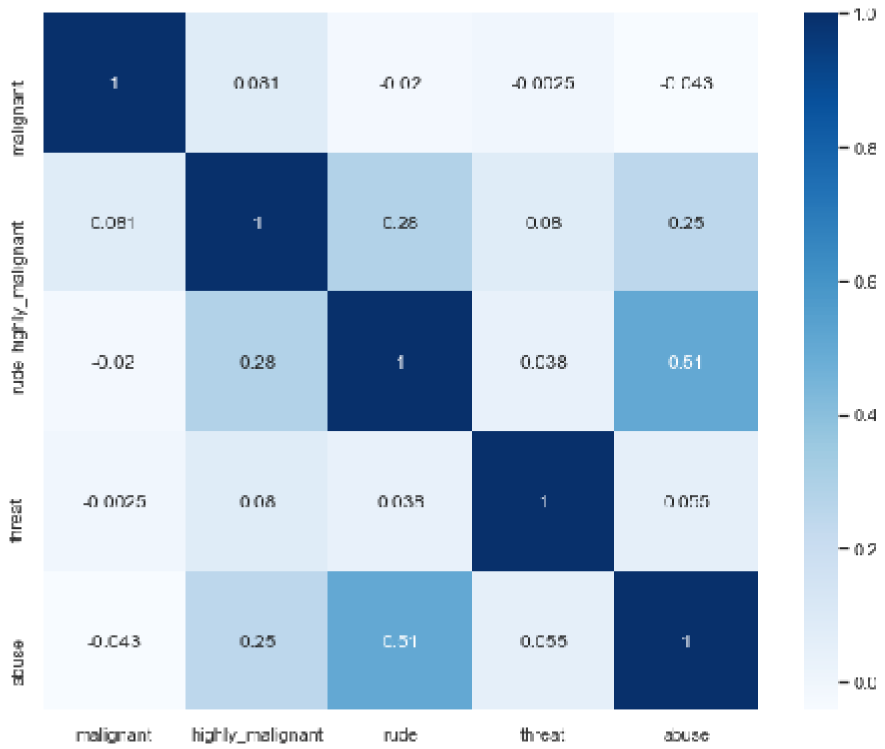


Figure 1bar graph for visualising distribution of classes within each label

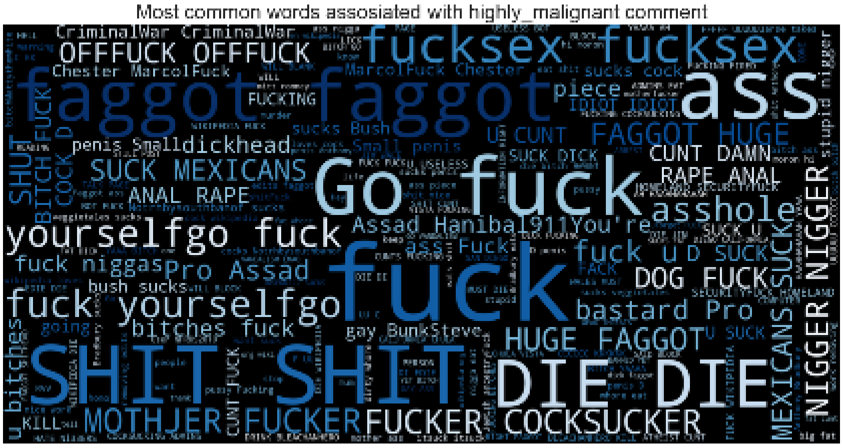






- As seen in the cross-correlation matrix, there is a high chance of rude comments to be abusing.

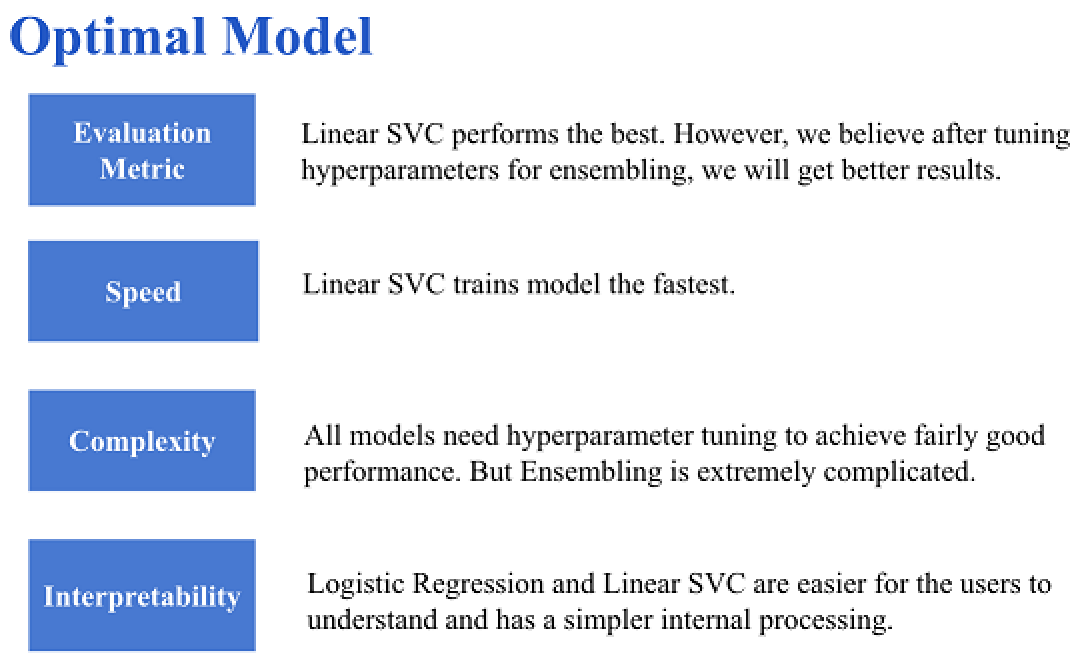
  In order to get an idea of what are the words that contribute the most to different labels, we write a function to generate word clouds. The function takes in a parameter label (i.e., toxic, insult, threat, etc)



class to visualize the most common words contributing to the class:highly_malignant

- Interpretation of the Results

  In terms of evaluation metric, Linear SVC performs the best. But we believe after tuning hyperparameters for ensembling, we will get better results. Besides, Linear SVC trains model the fastest. Refering to interpretability, Linear SVC is also easier for the users to understand and has a simpler internal processing. Therefore, we choose Linear SVC as our optimal model.



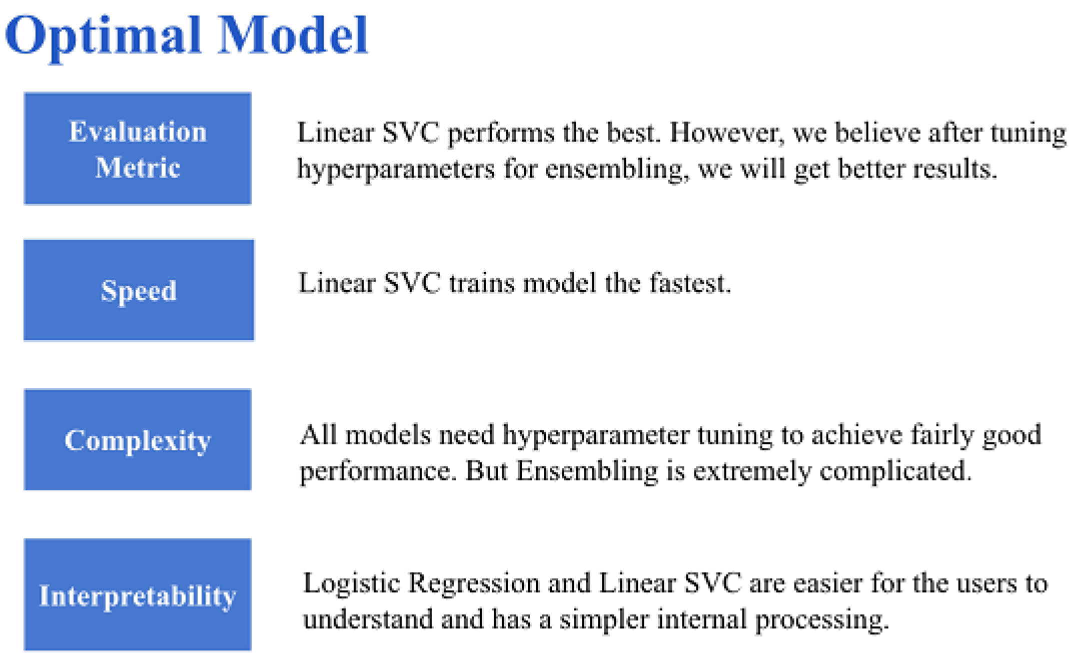
Optimal Model

Evaluation Metric	Linear SVC performs the best. However, we believe after tuning hyperparameters for ensembling, we will get better results.
Speed	Linear SVC trains model the fastest.
Complexity	All models need hyperparameter tuning to achieve fairly good performance. But Ensembling is extremely complicated.
Interpretability	Logistic Regression and Linear SVC are easier for the users to understand and has a simpler internal processing.

CONCLUSION

- Key Findings and Conclusions of the Study

It was a key point to note that almost all models generated by boosting methods had low values of hamming losses compared to earlier models



Optimal Model

Evaluation Metric

Linear SVC performs the best. However, we believe after tuning hyperparameters for ensembling, we will get better results.

Speed

Linear SVC trains model the fastest.

Complexity

All models need hyperparameter tuning to achieve fairly good performance. But Ensembling is extremely complicated.

Interpretability

Logistic Regression and Linear SVC are easier for the users to understand and has a simpler internal processing.

- Learning Outcomes of the Study in respect of Data Science

  Understood tokenizing, lemmatizing application of F1score Recall and models used for NLP classification

- Limitations of this work and Scope for Future Work

- Try more ways of vectorizing text data.
- Go deeper on feature engineering : Spelling corrector, Sentiment scores, n-grams, etc.
- Advanced models (e.g., lightgbm).
- Advanced Ensemble model (e.g., stacking).
- Deep learning model (e.g., LSTM).
- Advanced hyperparameter tuning techniques (e.g., Bayesian Optimization).