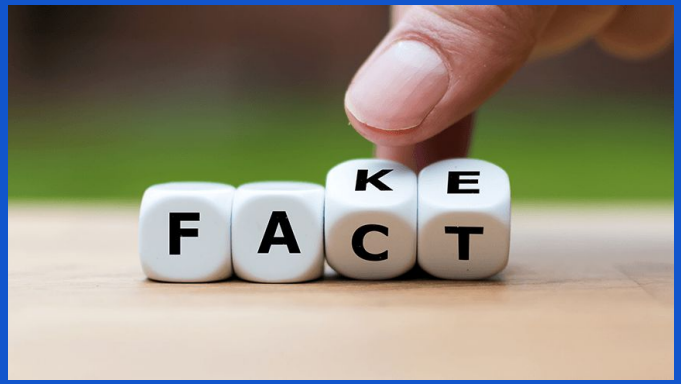
CSE 573 Project Group 16: Fake Reviews Detection Demo

By:

- Visaj Nirav Shah
- Mohil Devan Khimani
- Hastin Himanshubhai Modi
- Shivam Milind Bodiwala
- Shivani Shailesh Nandani
- Rahil Ashish Shah



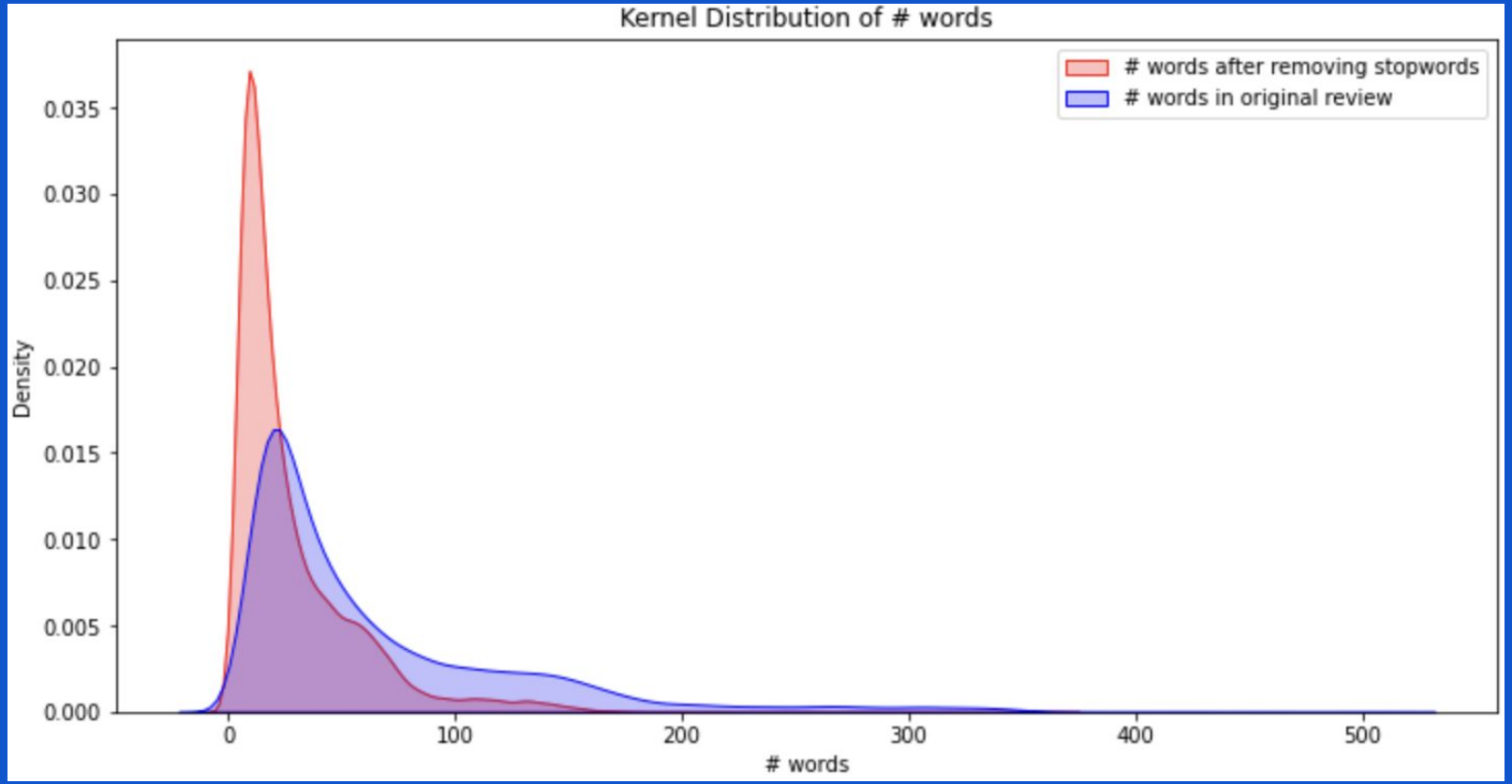
Introduction

- Online reviews are crucial in customers' decision-making process but have led to an increase in fake reviews.
- Fake reviews are a challenge for online platforms and consumers as these fake reviews, generated by fake accounts, bots, or paid individuals, can mislead customers.
- Various techniques have been developed to detect fake reviews, but more advanced methods are needed.
- This project compares and analyzes traditional and modern fake review detection techniques.
- The algorithms will be tested on two types of datasets – GPT-2 generated and back translation generated.
- The aim is to identify the most effective techniques for detecting fake reviews.
- The proposed system will use machine learning and NLP to analyze reviews and achieve high accuracy, speed, and scalability.
- The results could help improve online review systems and aid consumer decision-making.

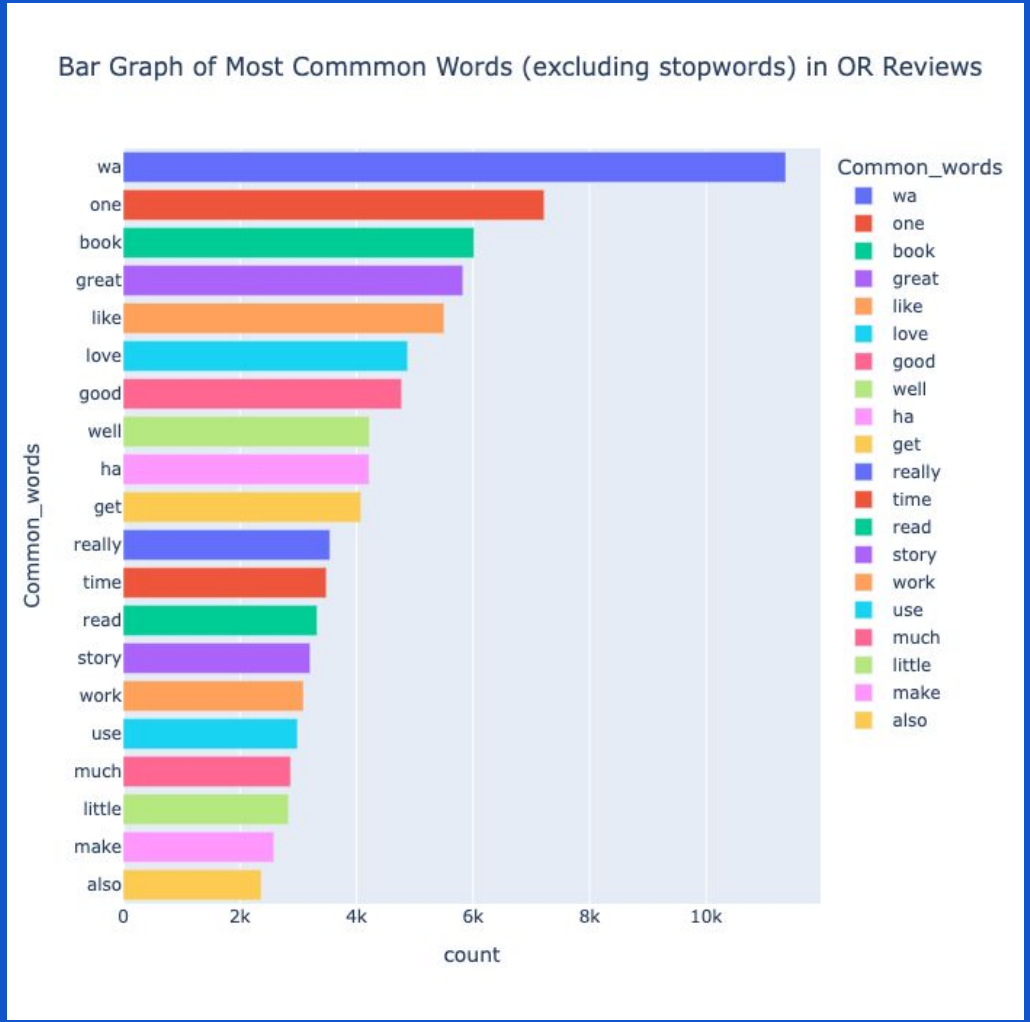
Dataset

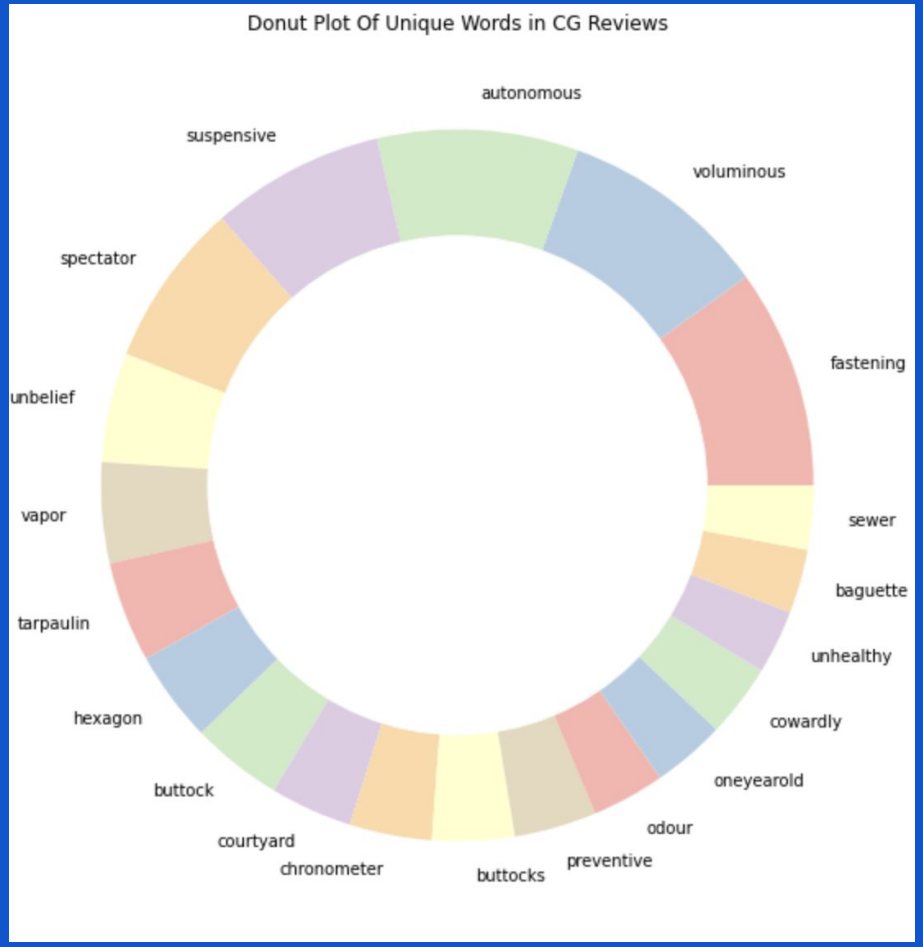
- The dataset used is the Amazon Review Data (2018) dataset.
- The dataset has been reduced to extract k-core subsets with each category having ~4000 samples each.
- The dataset includes ~40,000 samples for model training and testing.
- The dataset consists of text data in the form of product reviews.
- The dataset includes features such as product category, product rating, review text, and class label.
- The authors used stratified sampling to generate reviews from each selected product category.
- The dataset includes reviews from 10 product categories, each with 4,000 samples (2,000 real and 2,000 fake).

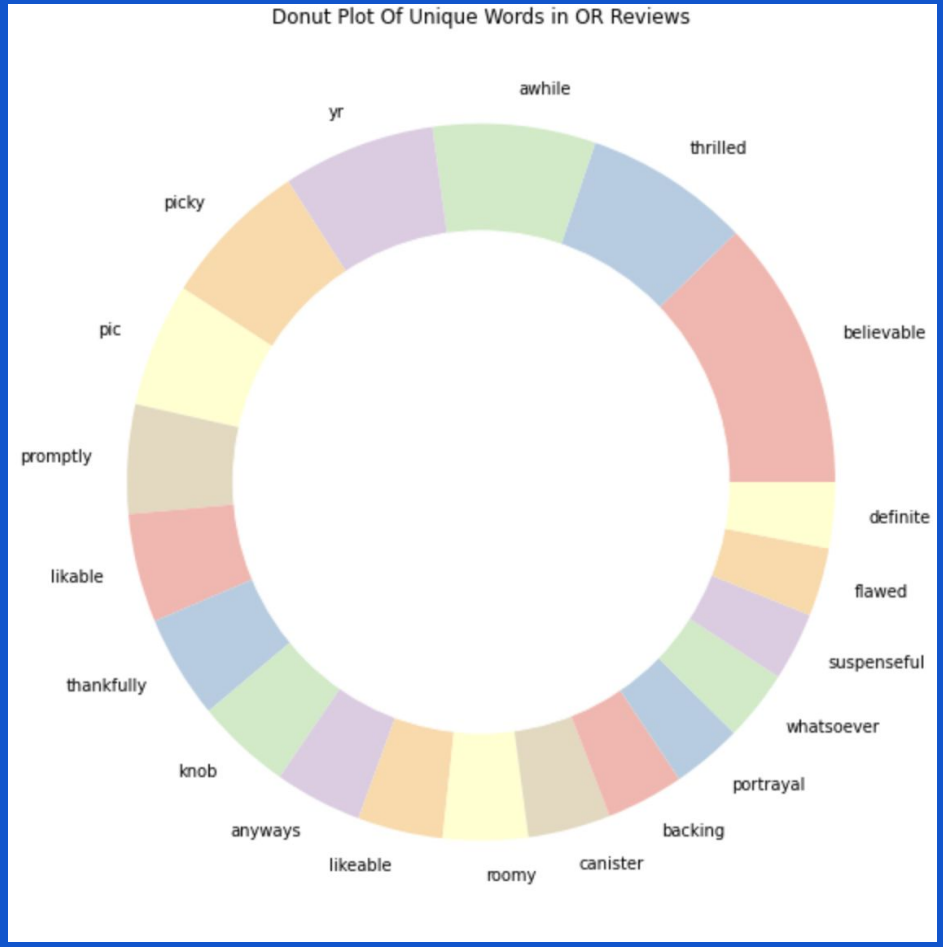
Data Visualizations


Kernel Distribution of # words
words after removing stopwords
words in original review
Density
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
100
200
300
400
500
words

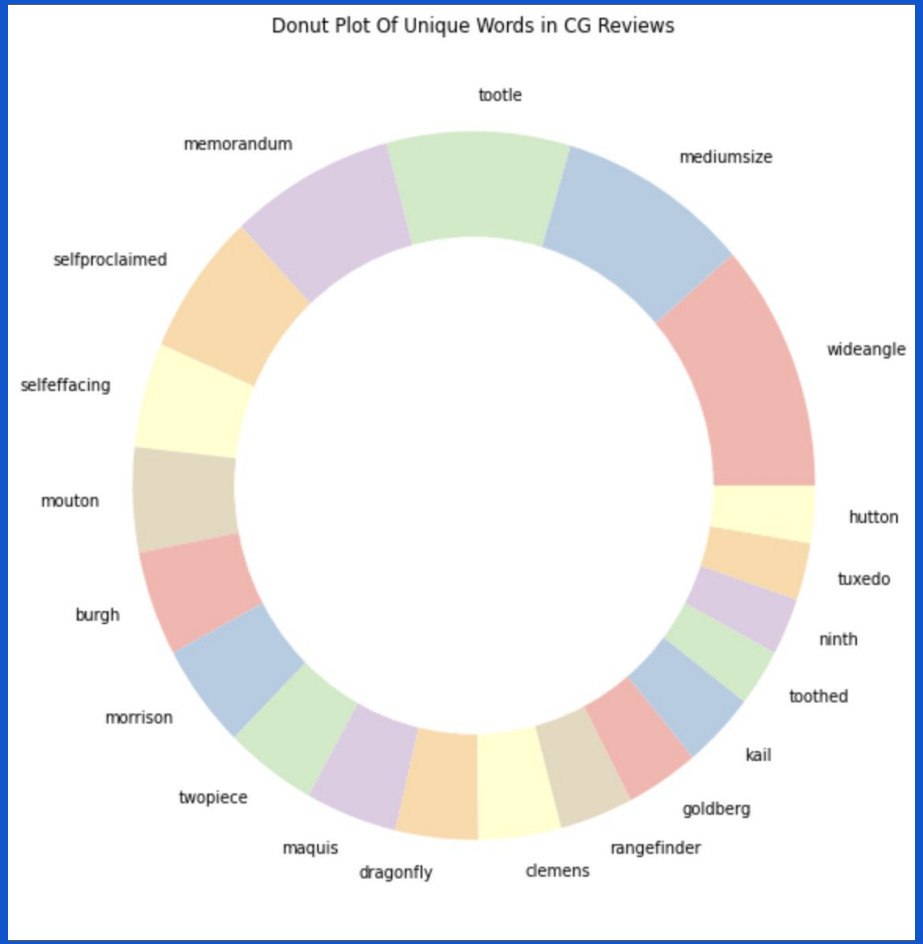
Tree of Most Common Words (excluding stopwords) in CG Reviews

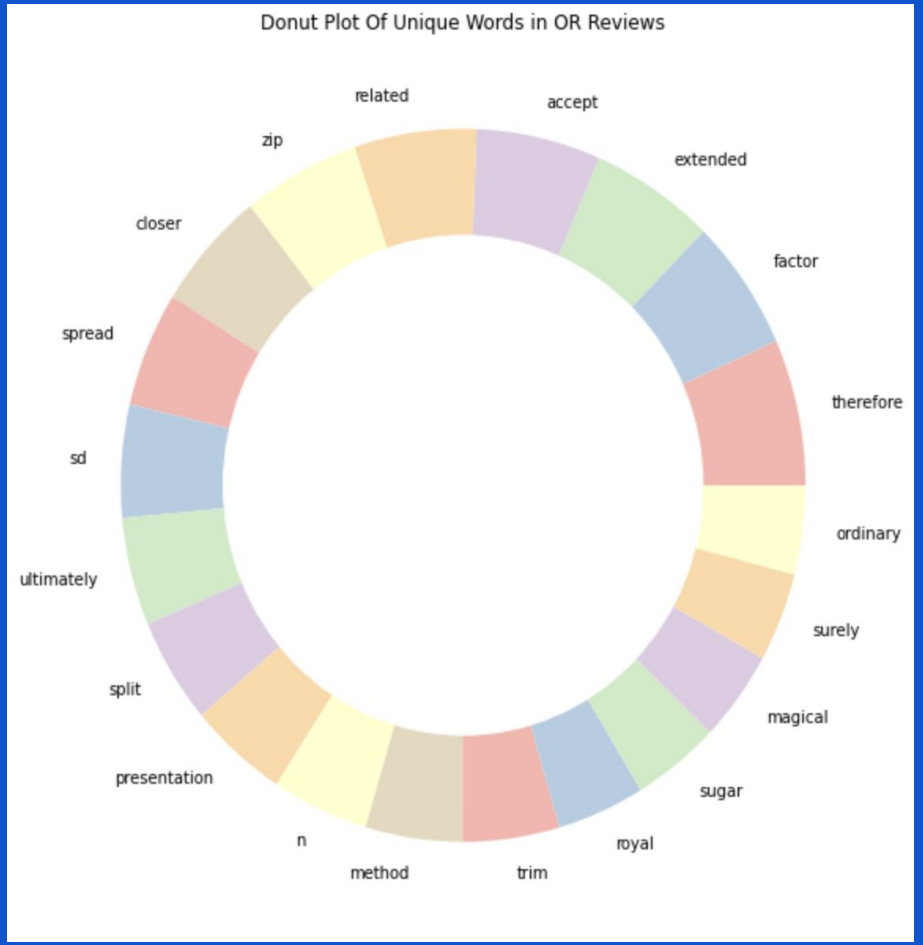

Bar Graph of Most Commmon Words (excluding stopwords) in OR Reviews
Common_words
wa
one
book
great
like
love
good
well
ha
get
really
time
read
story
work
use
much
little
make
also
0
2k
4k
6k
8k
10k
count
Common_words
wa
one
book
great
like
love
good
well
ha
get
really
time
read
story
work
use
much
little
make
also





Based on New Data (Fake using back translation)





Based on Original Data (Fake using GPT-2)

Data Preprocessing (For traditional models)

1. Data cleaning: Removed any special characters, extra spaces, numbers or punctuation marks that are not relevant to the analysis.
2. Vectorization: Used the tfidf vectorizer in sklearn to for the text data.
3. One hot encoding: Used the one hot encoder in sklearn for categorical variables

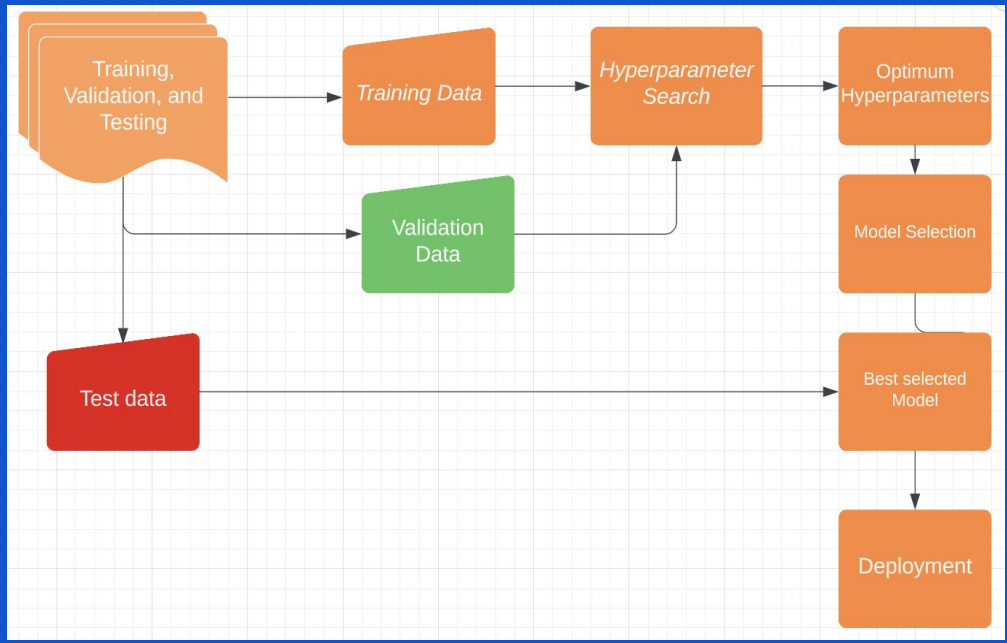
This data is finally passed to the RandomizedSearchCV function of sklearn to find the best hyperparameters for the respective models based on the cross-validation results.

Data Preprocessing (For advanced models)

1. Data cleaning: Removal of any special characters, extra spaces, numbers or punctuation marks that are not relevant to the analysis.
2. Tokenization: Tokenization is the process of breaking up a text into individual words or tokens.
3. Vocabulary building: After the text has been tokenized, a vocabulary is built from the unique words in the text.
4. Sequencing: Once the vocabulary has been built, the text is converted into sequences of integers.
5. Padding: In order to ensure that all sequences are of the same length, padding is added to the sequences.
6. Conversion to numpy arrays: The preprocessed data and labels are converted to numpy arrays so that they can be used for modeling.

Algorithms and Architectures

- The baseline model is built using Logistic Regression machine learning algorithm.
- The other models used for the classification include K-Nearest neighbours, Support-vector classifier, Naive Bayes, Tree-based classifiers, and deep-learning models. All the models have been trained on 6 datasets - 2 datasets (new and old) with each being trained on 3 combinations of data.
- The deep learning models used for classification are BERT, CNN, and CNN with attention



Evaluation Plan

- Evaluate metrics relevant to classification tasks using Scikit-learn
  - Accuracy
  - Precision
  - Recall
  - F1-score
- Focus on Recall
  - “Not fake” detected as “fake” is acceptable upto some extent, but the other way around is not
- Try out a variety of combinations of models and features
  - For example,
  - Remove features with low impact to speed up inferences
- Deploy the best model to create a web application
  - Accepts user input (a review)
  - Output is a probability measure – how likely it is fake

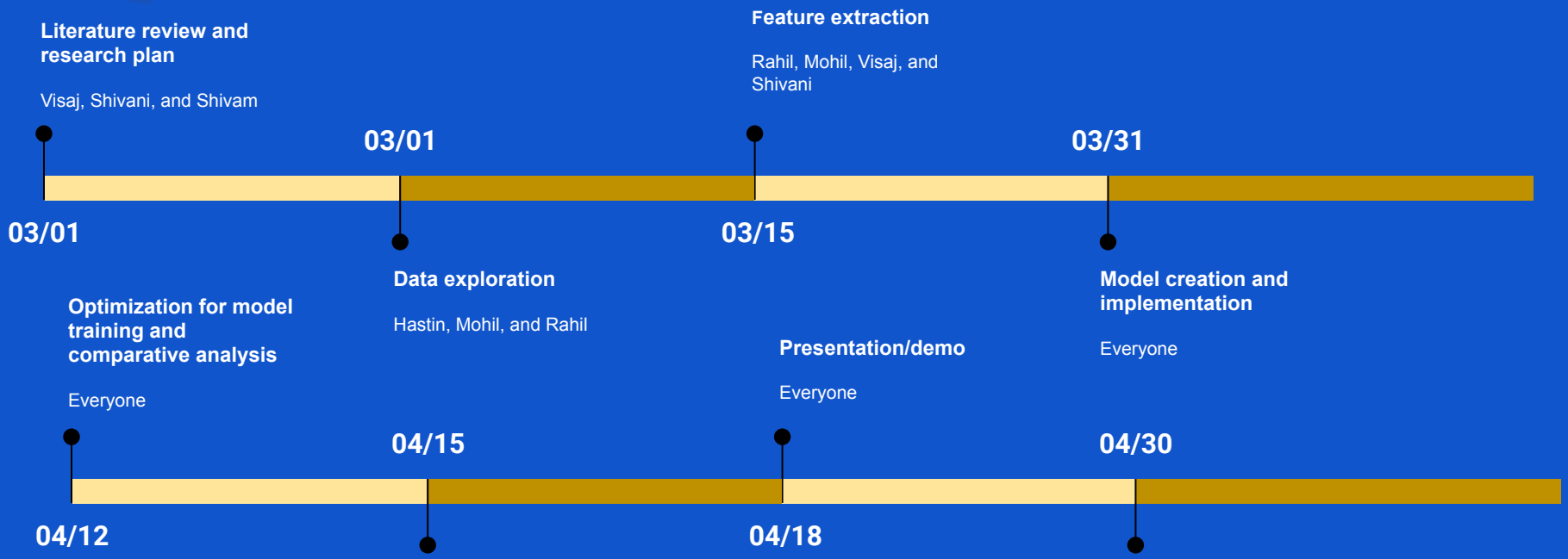
Results (Traditional models - Using sentiment instead of rating as a feature)

Algorithm	Accuracy % (old)	Recall % (old)	Accuracy % (new)	Recall % (new)
Logistic Regression	**92.1**	**92.5**	**77.1**	**80.9**
K-nearest neighbors	77.6	90	54.4	49.6
Support Vector Classifier	88.7	89.8	71.2	76.4
Naive Bayes	87.1	87.4	73.1	76.3
Decision Tree	80.8	74.7	63.8	80.1
Random Forests	86.4	85.6	67.8	75
AdaBoost	85.9	84.8	71.7	**81**
XGBoost	91.2	90.2	74.2	80

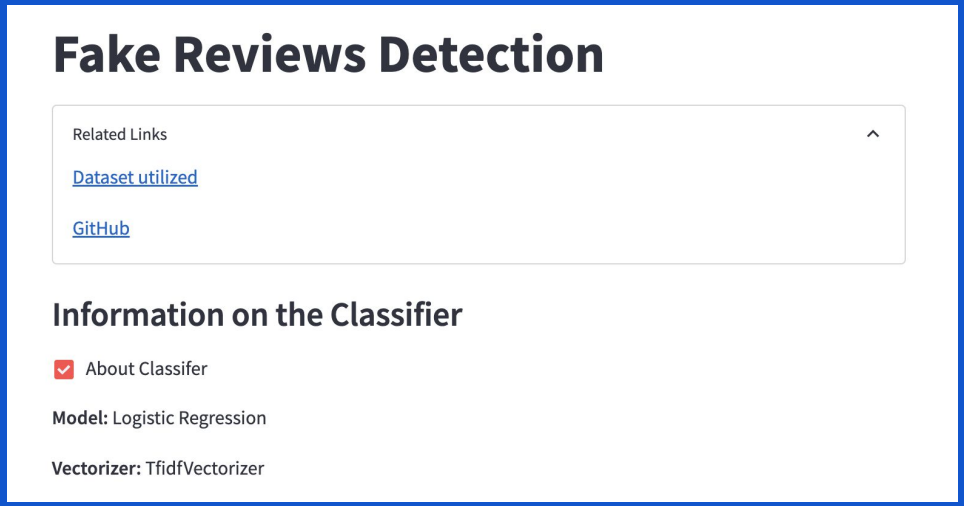
Results (Advanced models - Using sentiment instead of rating as a feature)

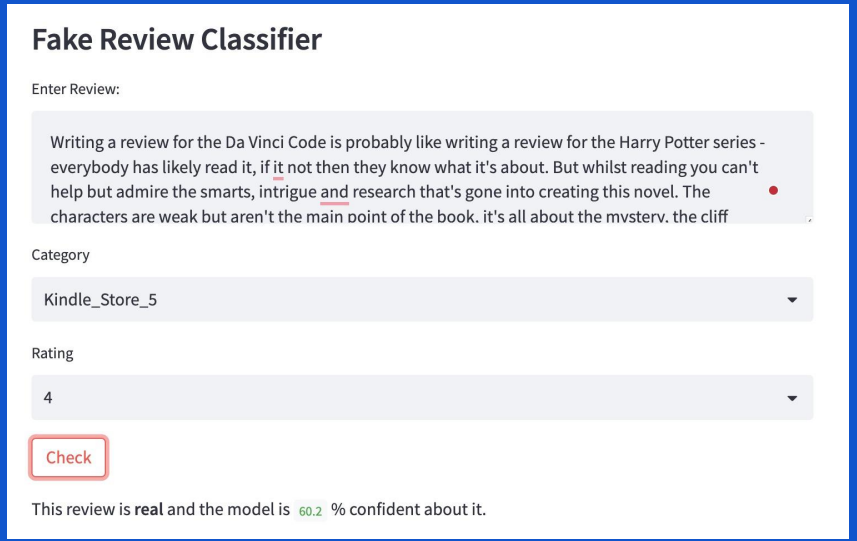
Algorithm	Accuracy % (old)	Recall % (old)	Accuracy % (new)	Recall % (new)
BERT	89.4	91.8	65.8	67.4
CNN	93.2	93.2	66.0	66.1
CNN + Attention	**94.7**	**95.5**	**75.9**	**83.6**
BiLSTM	93.3	93.0	74.0	76.7

Timelines and division of work



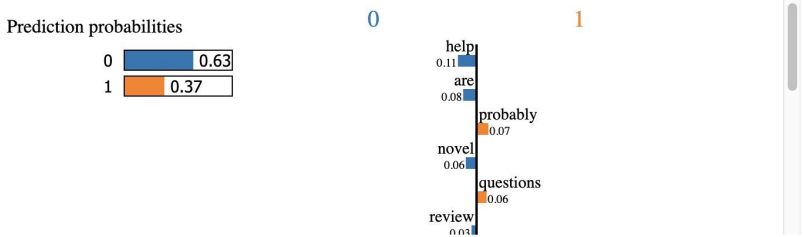
Project Demo



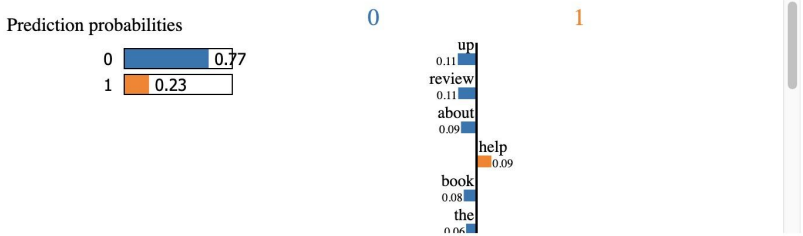


Project Demo

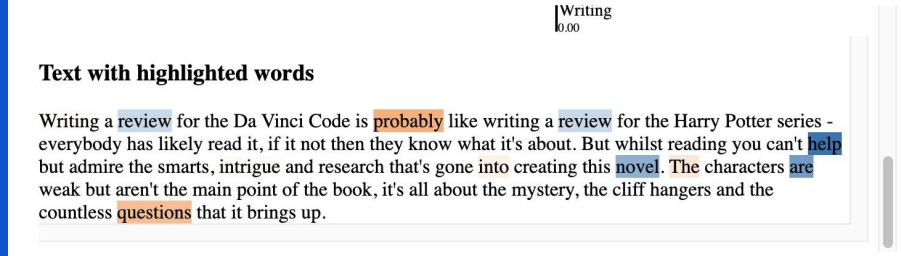
The explanation of the model output by keeping stopwords.



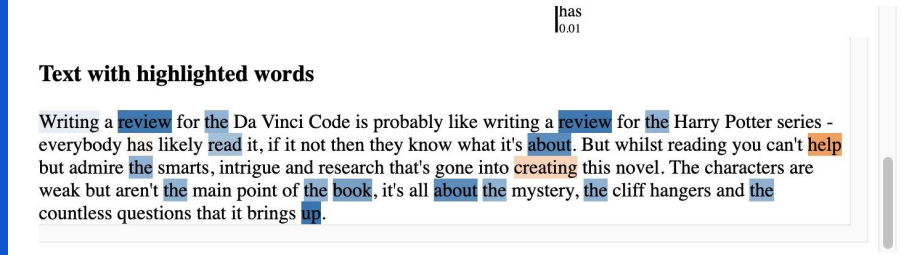
The explanation of the model output by removing stopwords.



The explanation of the model output by keeping stopwords.

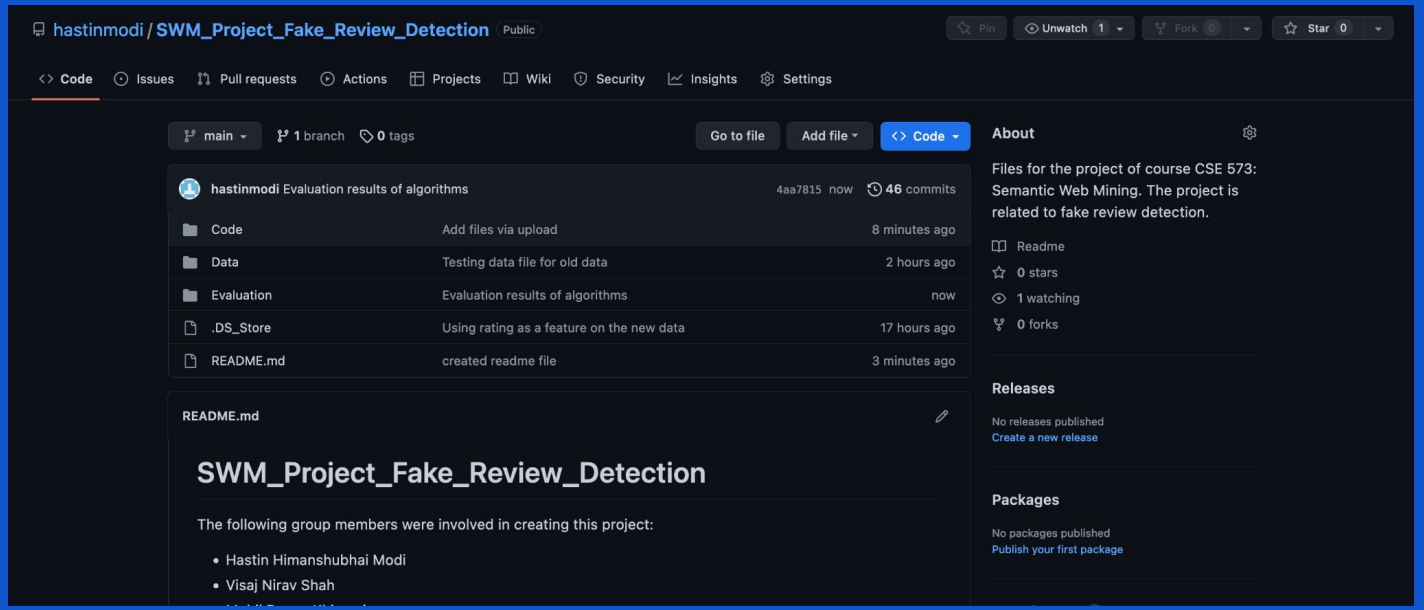


The explanation of the model output by removing stopwords.



GitHub

The project files are uploaded on GitHub here - https://github.com/hastinmodi/SWM_Project_Fake_Review_Detection



Thank you.