Cyberbullying Detection on Social Networks Using Hybrid RNN-LSTM Model
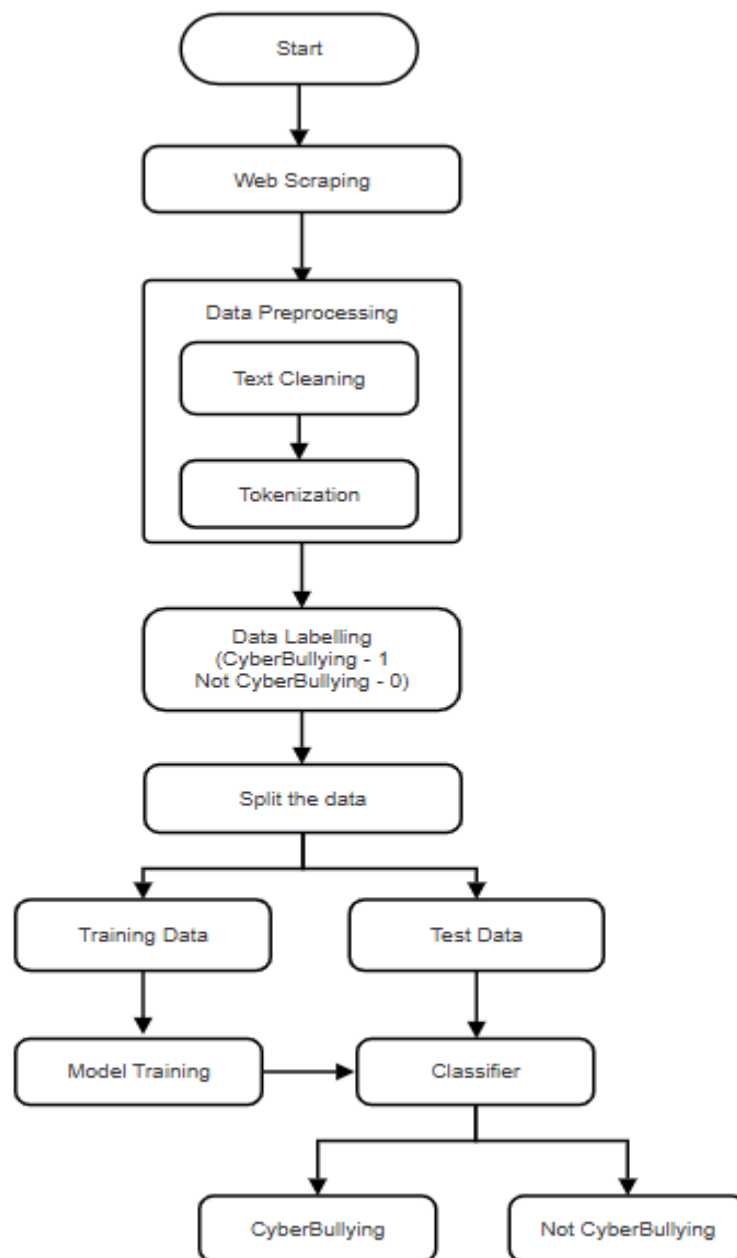
Flow Chart:

1. Web Scraping:

Web scraping is a technique used to extract large amounts of data from websites in an automated fashion, and it is particularly useful for collecting user-generated content such as comments from platforms like Reddit and YouTube. Using Python libraries such as PRAW (Python Reddit API Wrapper) and Google's YouTube Data API, we can efficiently gather comments from specific posts or videos. For example, I have collected comments from the Reddit posts "Can't fix stupid" (https://www.reddit.com/r/SweatyPalms/comments/1fi07jp/cant_fix_stupid/) and "Donald Trump side angle from his rally" (https://www.reddit.com/r/pics/comments/1g4rtnh/donald_trump_side_angle_from_his_rally_in/) by authenticating the Reddit API, as well as from YouTube videos such as the live stream https://www.youtube.com/live/qqG96G8YdcE?si=07X1Iw1q2F12v2Vw and the video "SX-BT-Ua07g" (https://youtu.be/SX-BT-Ua07g?si=tEmhnGZK_y-zc3P_) using the YouTube Data API. These fetched comments are saved in a structured format like CSV for further analysis.

2. Data Preprocessing:

Data preprocessing is a crucial step in preparing raw data for analysis or machine learning tasks, as it cleans and transforms the data into a more usable form. The process typically involves handling inconsistencies, removing irrelevant parts of the data, and transforming it into a format that is better suited for further analysis.

In our project, We preprocessed Reddit and YouTube comments from a dataset aimed at detecting cyberbullying. The steps we followed include:

- ✓ **Lowercasing:** We converted all comments to lowercase to ensure uniformity, making comparisons between words easier and avoiding case-sensitivity issues.

- ✓ **URL Removal:** Using regular expressions, we removed any URLs from the comments, as they are not relevant for content-based analysis.
- ✓ **Stopword Removal:** We removed common English stopwords like "the", "and", and "is" using NLTK's stopword list to reduce noise in the data and focus on more meaningful words.
- ✓ **Tokenization:** We split the comments into individual words (tokens) using NLTK's word_tokenize, preparing the text for more granular analysis.
- ✓ **Lemmatization:** Using WordNetLemmatizer, We reduced words to their root form (e.g., "running" becomes "run"), which helps standardize different forms of the same word.
- ✓ **Saving the Cleaned Data:** Finally, we saved the preprocessed comments to a CSV file, making them ready for further analysis such as text classification for cyberbullying detection.
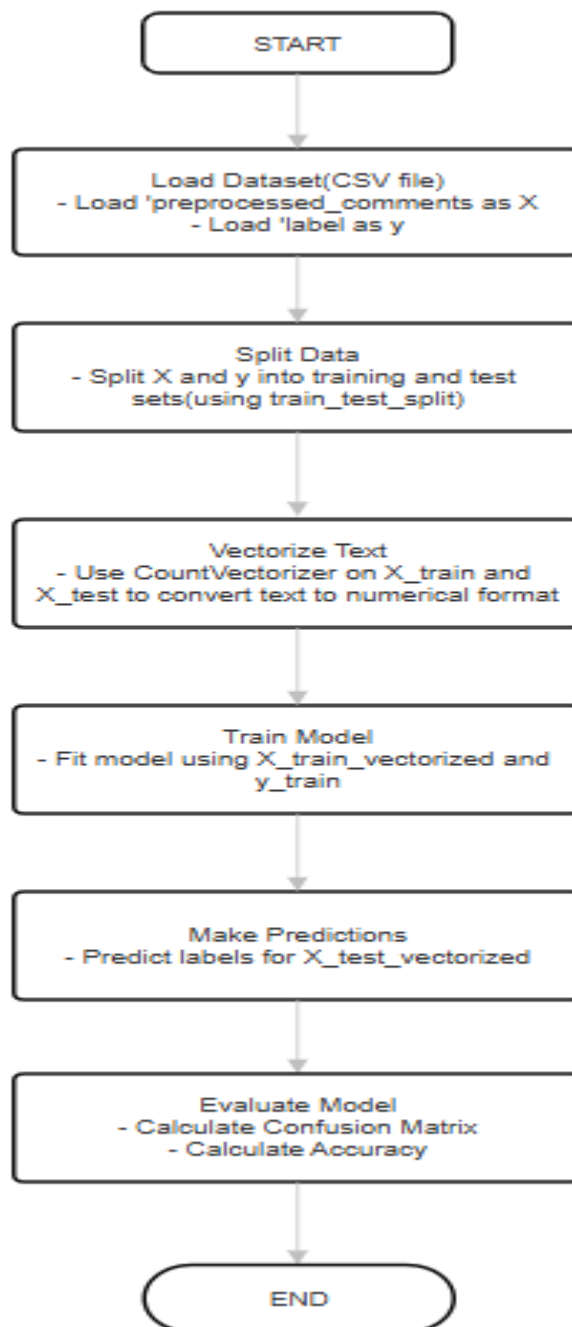
	A	preprocessed_comments
1	preprocessed_comments	
2	unsteady biden subdued trump debate analyzed	
3	trump president future change mean bad	
4	joe biden pls win	
5	economy tank thanks biden harris	
6	trump win	
7	would love guy make grumpier old men sequel lol fighting golf haha trump 2024	
8	swear half time biden asleep	
9	long live trump	
10	325 million american chose clown	
11	im gon na miss trump	
12	oligarch defend interest oligarch ordinary people	
13	donald trump care others himslef devil person bad complains lot problem united state ameerica	
14	neither mentioned anything rehab center people family need want help get never stop coming resource	
15	seen swing know swing let act like child	
16	donald trump love indonesia	
17	blame democrat saying acceptable	
18	dervish please help help help	
19	snl much much better	
20	plzzzz vote trump 2024 want save country	
21	dear american people sake humanity urge beg please please vote president government likely bad rest world please big favour stop voting forever whole world thank	
22	ungracious shake hand	
23	thumbnail insane	
24	way fan biden fully support trump medical professional watching biden mental state decline break heart	
25	greatest democracy world	
26	dementia joe	
27	biden always mediocre buffoon really seal deal dismal career fleecing taxpayer	
28	biden visiting angel help	
29	trump looked worse biden debating kamala pussed 2nd debate	

3. Data Labelling:

Data labeling is an essential process where data points are tagged with labels to train models for specific tasks. In our project, the goal was to label Reddit and YouTube comments to distinguish between cyberbullying and non-cyberbullying content. We manually reviewed each comment and assigned a binary label: 1 for comments that exhibited signs of cyberbullying, such as harassment, insults, or aggressive language, and 0 for comments that were neutral or non-harmful. This labeling process is critical because it provides the training data required for a model to learn how to classify new, unseen comments accurately. The manual review ensured that context and tone were considered, which can be difficult for automatic labeling techniques to handle effectively.

preprocessed_comments	label
unsteady biden subdued trump debate analyzed	0
trump president future change mean bad	0
joe biden pls win	0
economy tank thanks biden harris	1
trump win	0
would love guy make grumpier old men sequel lol fighting golf haha trump 2024	0
swear half time biden asleep	1
long live trump	0
325 million american chose clown	1
im gon na miss trump	0
oligarch defend interest oligarch ordinary people	0
donald trump care others himslef devil person bad complains lot problem united state ameerica	1
neither mentioned anything rehab center people family need want help get never stop coming resource	0
seen swing know swing let act like child	1
donald trump love indonesia	0
blame democrat saying acceptable	0
dervish please help help help	0
snl much much better	0
plzzzz vote trump 2024 want save country	0
dear american people sake humanity urge beg please please vote president government likely bad rest world please big favour stop voting forever whole world thank	0
ungracious shake hand	1
thumbnail insane	0
way fan biden fully support trump medical professional watching biden mental state decline break heart	1
greatest democracy world	0
dementia joe	1
biden always mediocre buffoon really seal deal dismal career fleecing taxpayer	1
biden visiting angel help	0
trump looked worse biden debating kamala pussed 2nd debate	1
idiocracy maga documentary	1
noticed go read comment higher cost grocery go	0
trump november 2024	0
trump look	0
whole 250m citizen incl abroad remote area also vote	0
second orange mooshes drawer	1
much changed month	0
ever see debate one candidate swallow tongue instant classic	1
shitted self	1
fight fight fight	0
brandon speak	1
period period period biden said unknowingly	1
trump sharp	0
biden one foot banana peal foot grave	1
trump sh pant disgusting	1
woke coma first video saw checking current president would 100 think trump	0
come back oct 24 watch show	0
guy true soon comment can not see comment see video	0
biden become rock	0
dimencha anyone	1
joe dear headlight go far enough wow border patrol endorse	0

MILESTONE-2

```
                    ┌──────────────────────┐
                    │        START         │
                    └──────────────────────┘
                               │
                               ▼
          ┌──────────────────────────────────────┐
          │        Load Dataset(CSV file)         │
          │ - Load 'preprocessed_comments as X    │
          │          - Load 'label as y           │
          └──────────────────────────────────────┘
                               │
                               ▼
          ┌──────────────────────────────────────┐
          │              Split Data               │
          │ - Split X and y into training and test│
          │        sets(using train_test_split)   │
          └──────────────────────────────────────┘
                               │
                               ▼
          ┌──────────────────────────────────────┐
          │            Vectorize Text             │
          │  - Use CountVectorizer on X_train and │
          │ X_test to convert text to numerical   │
          │               format                  │
          └──────────────────────────────────────┘
                               │
                               ▼
          ┌──────────────────────────────────────┐
          │             Train Model               │
          │ - Fit model using X_train_vectorized  │
          │            and y_train                │
          └──────────────────────────────────────┘
                               │
                               ▼
          ┌──────────────────────────────────────┐
          │           Make Predictions            │
          │ - Predict labels for X_test_vectorized│
          └──────────────────────────────────────┘
                               │
                               ▼
          ┌──────────────────────────────────────┐
          │            Evaluate Model             │
          │  - Calculate Confusion Matrix         │
          │       - Calculate Accuracy            │
          └──────────────────────────────────────┘
                               │
                               ▼
                    ┌──────────────────────┐
                    │         END          │
                    └──────────────────────┘
```

1. **Load Dataset (CSV file)**

- **Description**: Load the labeled dataset from a CSV file.

- **Details**:

    o Extract the 'preprocessed_comments' column as XXX, containing the text data.

    o Extract the 'label' column as yyy, containing the target labels (cyberbullying vs. non-cyberbullying).

2. **Split Data**

- **Description**: Split the dataset into training and test sets to train and evaluate the model separately.

- **Details**:

    o Use train_test_split to divide XXX and yyy into training (for model learning) and test sets (for evaluation).

    o Set a test size and a random seed for reproducibility.

3. **Vectorize Text**

- **Description**: Transform text data into numerical format for model input.

- **Details**:

    o Use CountVectorizer to convert text in X_{train} and X_{test} into matrices of token counts, which serve as feature vectors for each comment.

4. **Train Model**

- **Description**: Train models like logistic regression, random forests on the vectorized training data.

- **Details**:

    o Fit the model using X_train_vectorized and y_train

    o This allows the model to learn patterns in the data that indicate cyberbullying.

5. **Make Predictions**

- **Description**: Use the trained model to make predictions on the test data.

- **Details**:

    o Apply the model to X_test_vectorized and predict the label.

6. **Evaluate Model**

- **Description**: Measure the model's performance on the test set.

- **Details**:

    o Calculate the confusion matrix to assess classification performance in terms of true/false positives and negatives.

    o Compute accuracy to determine the overall percentage of correct predictions.
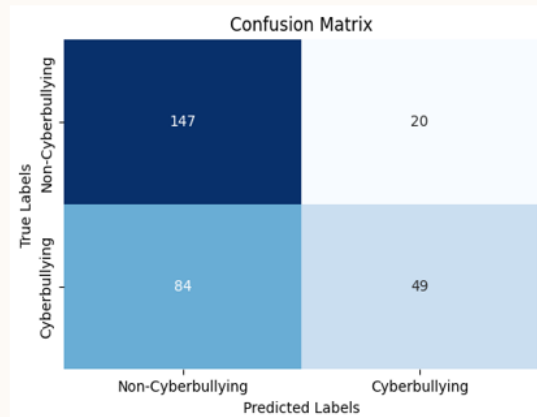
ANALYSIS:

Accuracy of different models at various test sizes

Model / Test size	20%	25%	30%	35%
Logistic Regression	65	68	70	69
Random Forest	60	63	65	67
Naïve Bayes	59	60	64	63

CONFUSION MATRIX:
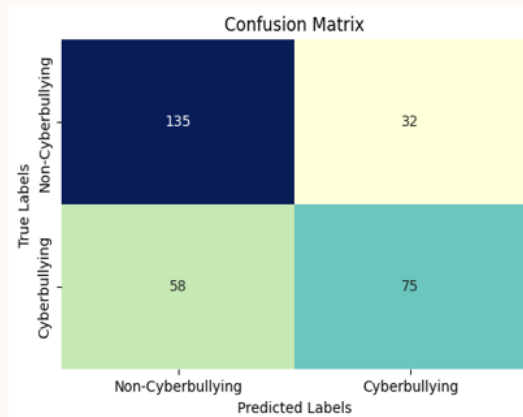
Confusion matrix for Logistic Regression

Confusion matrix for Random forest

ERRORS OCCURRED DURING THE MODEL TRAINING PHASE:

```
PS C:\cyber_bullying> python logistic.py
Traceback (most recent call last):
  File "C:\cyber_bullying\logistic.py", line 2, in <module>
    from sklearn.model_selection import train_test_split
ModuleNotFoundError: No module named 'sklearn'
```

Reason: did not install scikit-learn package
Solution:
pip install pandas scikit-learn

```
® PS C:\cyber_bullying> python logistic.py
 Traceback (most recent call last):
   File "C:\Users\hi\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\indexes\base.py", line 3805, in get_loc
     return self._engine.get_loc(casted_key)
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
   File "index.pyx", line 167, in pandas._libs.index.IndexEngine.get_loc
   File "index.pyx", line 196, in pandas._libs.index.IndexEngine.get_loc
   File "pandas\\_libs\\hashtable_class_helper.pxi", line 7081, in pandas._libs.hashtable.PyObjectHashTable.get_item
   File "pandas\\_libs\\hashtable_class_helper.pxi", line 7089, in pandas._libs.hashtable.PyObjectHashTable.get_item
 KeyError: 'preprocessed_comment'

 The above exception was the direct cause of the following exception:

 Traceback (most recent call last):
   File "C:\cyber_bullying\logistic.py", line 11, in <module>
     X = df['preprocessed_comment']
         ~~^^^^^^^^^^^^^^^^^^^^^^^^^
   File "C:\Users\hi\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\frame.py", line 4102, in __getitem__
     indexer = self.columns.get_loc(key)
               ^^^^^^^^^^^^^^^^^^^^^^^^^^
   File "C:\Users\hi\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\indexes\base.py", line 3812, in get_loc
     raise KeyError(key) from err
 KeyError: 'preprocessed_comment'
```

Reason: did not give the correct column name (given preprocessed_comment for preprocessed_comments)

```
® PS C:\cyber_bullying> python logistic.py
 Traceback (most recent call last):
   File "C:\cyber_bullying\logistic.py", line 27, in <module>
     model.fit(X_train, y_train)
   File "C:\Users\hi\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.py", line 1473, in wrapper
     return fit_method(estimator, *args, **kwargs)
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
   File "C:\Users\hi\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\linear_model\_logistic.py", line 1223, in fit
     X, y = self._validate_data(
            ^^^^^^^^^^^^^^^^^^^^^
   File "C:\Users\hi\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.py", line 650, in _validate_data
     X, y = check_X_y(X, y, **check_params)
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
   File "C:\Users\hi\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\utils\validation.py", line 1301, in check_X_y
     X = check_array(
         ^^^^^^^^^^^^
   File "C:\Users\hi\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\utils\validation.py", line 1012, in check_array
     array = _asarray_with_order(array, order=order, dtype=dtype, xp=xp)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
   File "C:\Users\hi\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\utils\_array_api.py", line 745, in _asarray_with
_order
     array = numpy.asarray(array, order=order, dtype=dtype)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
   File "C:\Users\hi\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\series.py", line 1031, in __array__
     arr = np.asarray(values, dtype=dtype)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 ValueError: could not convert string to float: 'scholar wow big number'
```

Reason: did not perform vectorization i.e, converting text to numerical format

Team Members:

D. HariChandana

D. Sudheer

K. Raghunadha Rao

T. Prashanth