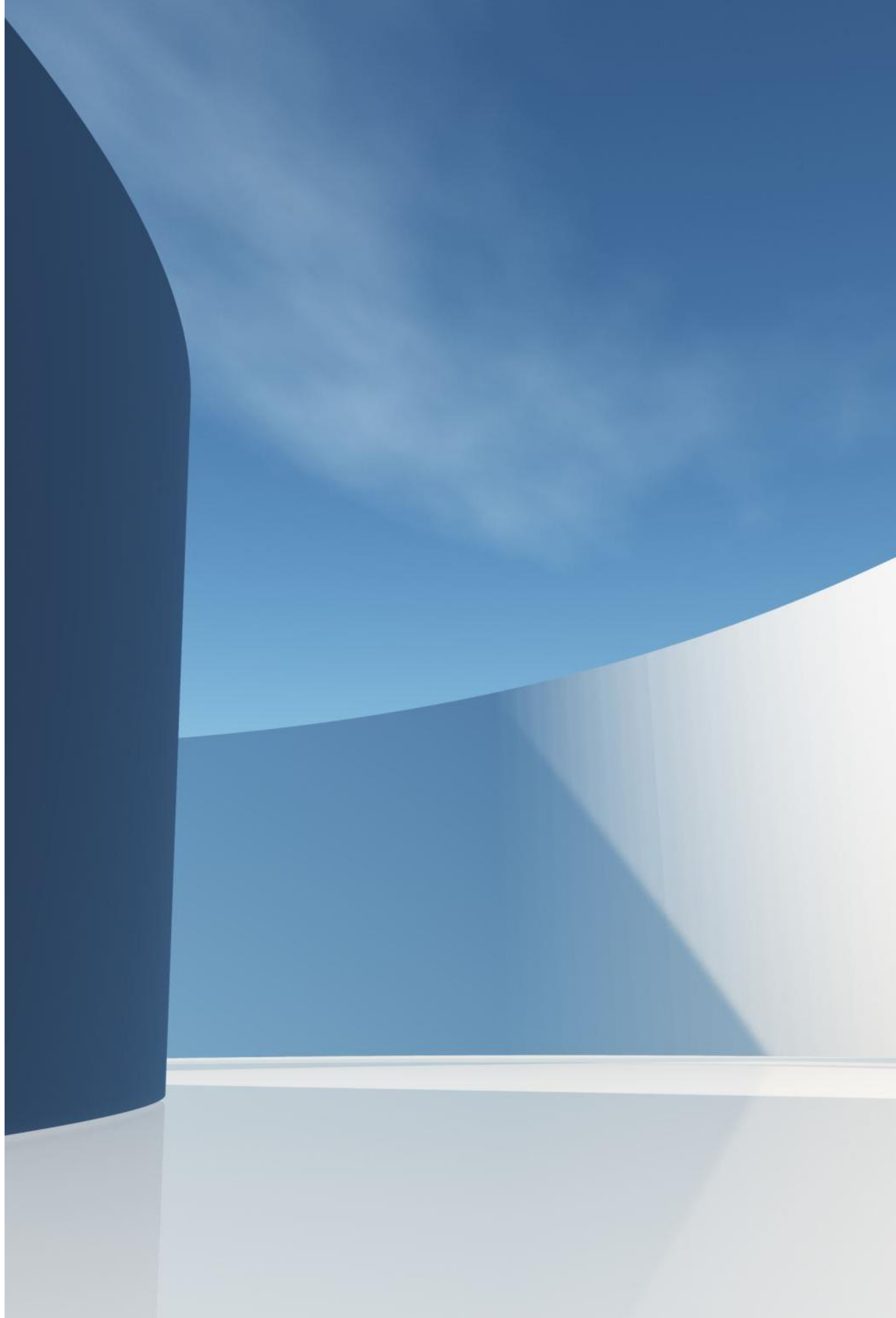
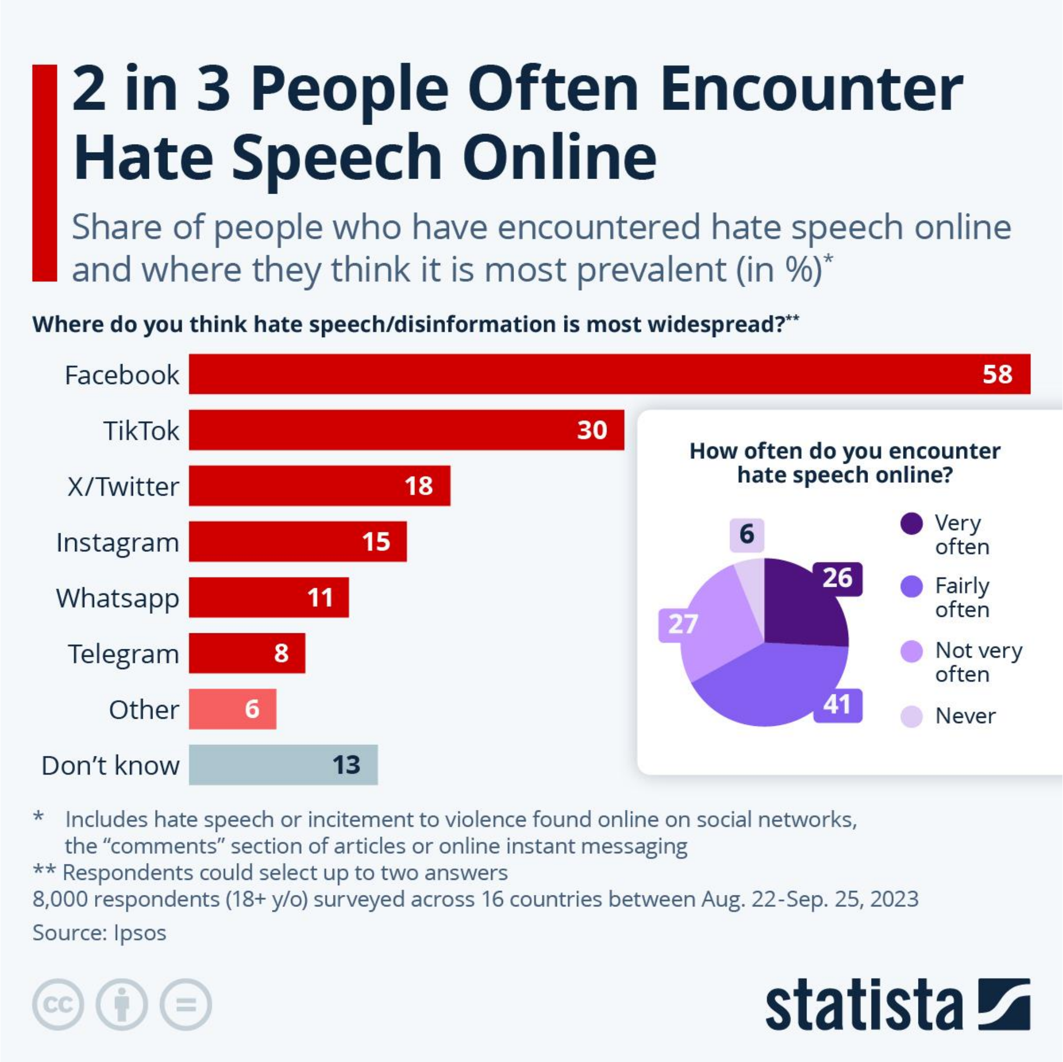
Progress Presentation

AI CONTENT MODERATION SYSTEM

LEELA SAI RAPARLA, NIKHIL VISHWANATH, RISHABH NAIR, PRINCE RUSWEKA RWABONGOYA



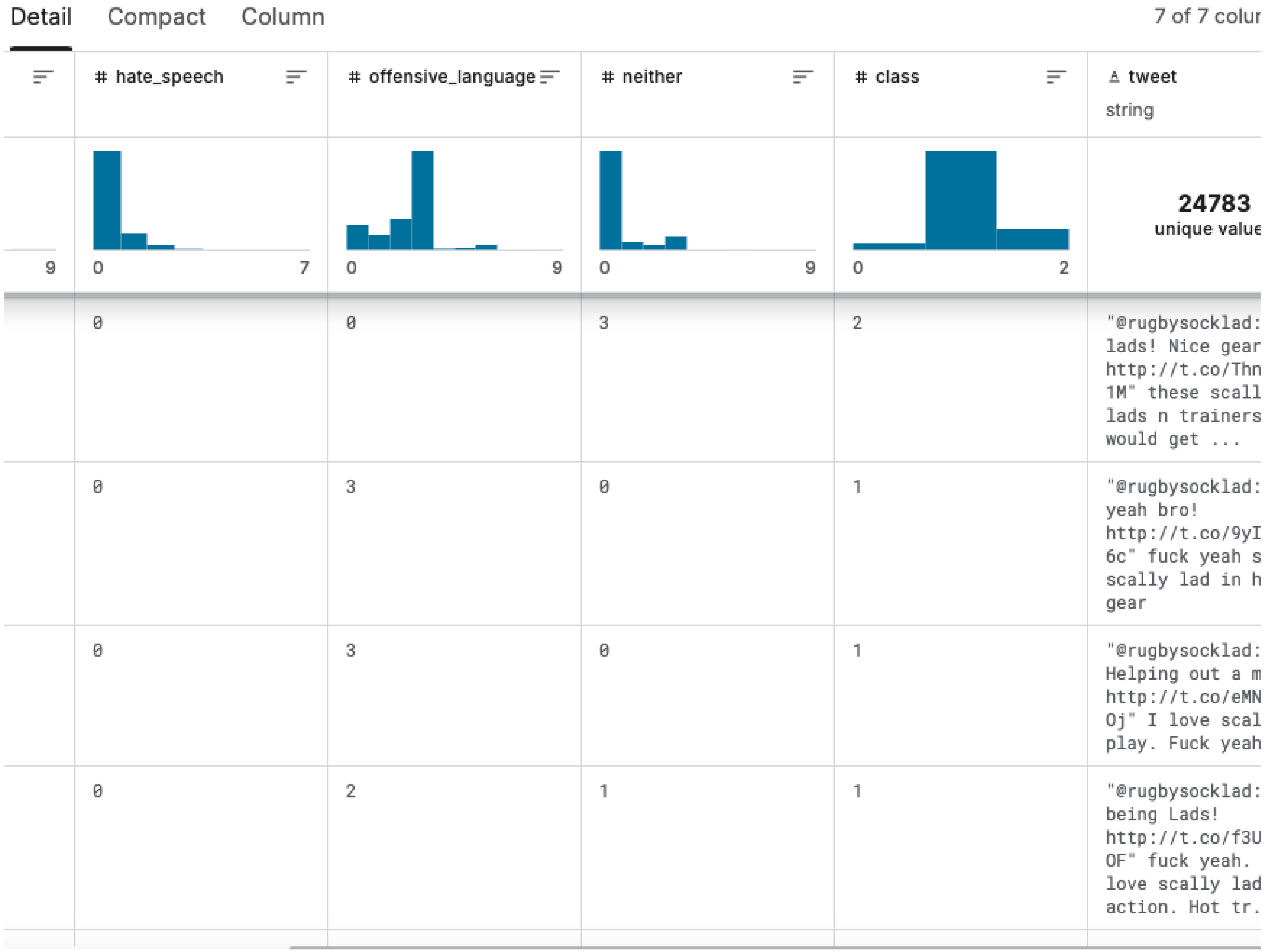


Motivation/ Objective

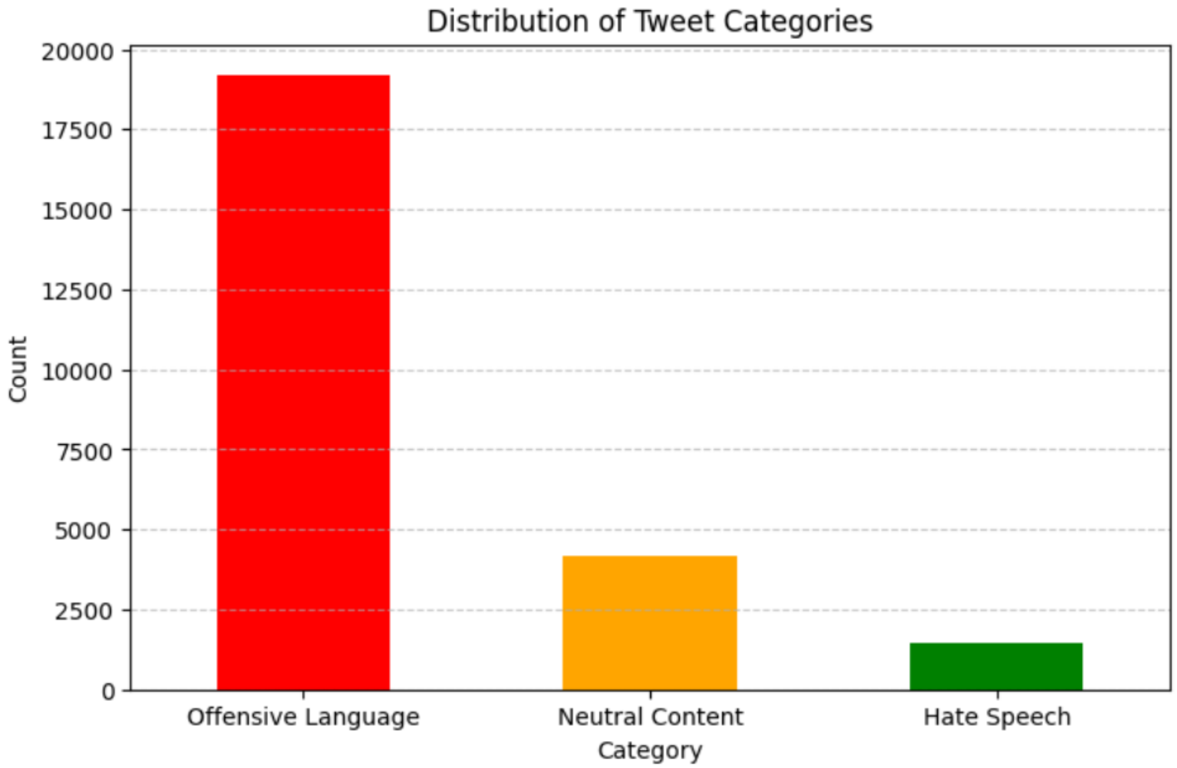
THE OBJECTIVE OF THIS PROJECT IS TO DEVELOP AN AI-POWERED CONTENT MODERATION SYSTEM THAT ACCURATELY CLASSIFIES TEXT INTO TWO CATEGORIES: *HATE SPEECH* AND *NOT HATE SPEECH*.

Kaggle Hate Speech Data Set

- 3 Categories
  - Hate Speech
    - Expresses hate towards a specific group
  - Offensive Language
    - Profane or abusive language
    - Does not target a particular group
  - Neutral Content
    - Non-offensive language



Data Visualizations



- The dataset is imbalanced, with **most tweets labeled as offensive language (class 1)**.
- Hate speech (class 0) is less frequent.
- Neutral content (class 2) is the least common category.

Data Preprocessing

Removing URLs, mentions, hashtags, special characters, and common stopwords.

```
import re

basic_stopwords = set([
    "a", "an", "the", "and", "or", "but", "if", "while", "with", "without", "about",
    "against", "between", "into", "through", "during", "before", "after", "above", "below",
    "to", "from", "up", "down", "in", "out", "on", "off", "over", "under", "again", "further",
    "then", "once", "here", "there", "when", "where", "why", "how", "all", "any", "both",
    "each", "few", "more", "most", "other", "some", "such", "no", "nor", "not", "only",
    "own", "same", "so", "than", "too", "very"
])

# Define a function to clean the tweet text without nltk stopwords
def clean_text(text):
    text = text.lower()  # Convert to lowercase
    text = re.sub(r'http\S+|www\S+', '', text)  # Remove URLs
    text = re.sub(r'@\w+', '', text)  # Remove mentions
    text = re.sub(r'#\w+', '', text)  # Remove hashtags
    text = re.sub(r'[^a-z\s]', '', text)  # Remove special characters and numbers
    text = ' '.join([word for word in text.split() if word not in basic_stopwords])  # Remove basic stopwords
    return text
```

AI Model: Naïve Bayes

```
df = pd.DataFrame(data)
print(df.head(5))
```

```
   Unnamed: 0  count  hate_speech  offensive_language  neither  class  \
0           0      3            0                   0        3      2
1           1      3            0                   3        0      1
2           2      3            0                   3        0      1
3           3      3            0                   2        1      1
4           4      6            0                   6        0      1

                                               tweet
0  !!! RT @mayasolovely: As a woman you shouldn't...
1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3  !!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
4  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...
```

```
model = MultinomialNB()
model.fit(X_train_vectors, y_train)
```

```
▾ MultinomialNB
MultinomialNB()
```

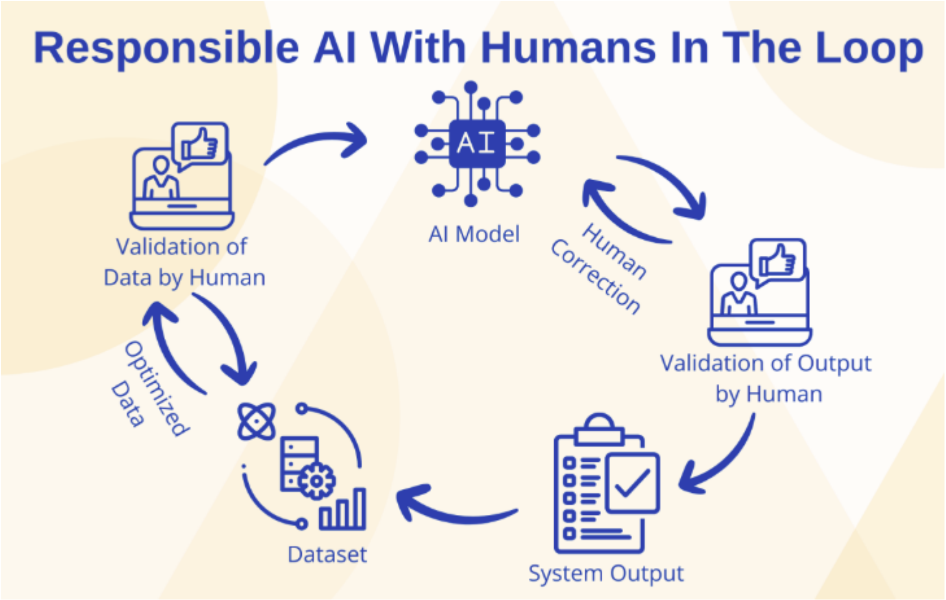
```
y_pred = model.predict(X_test_vectors)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100}%")
```

```
Accuracy: 84.94956287827841%
```

https://www.geeksforgeeks.org/multinomial-naive-bayes/

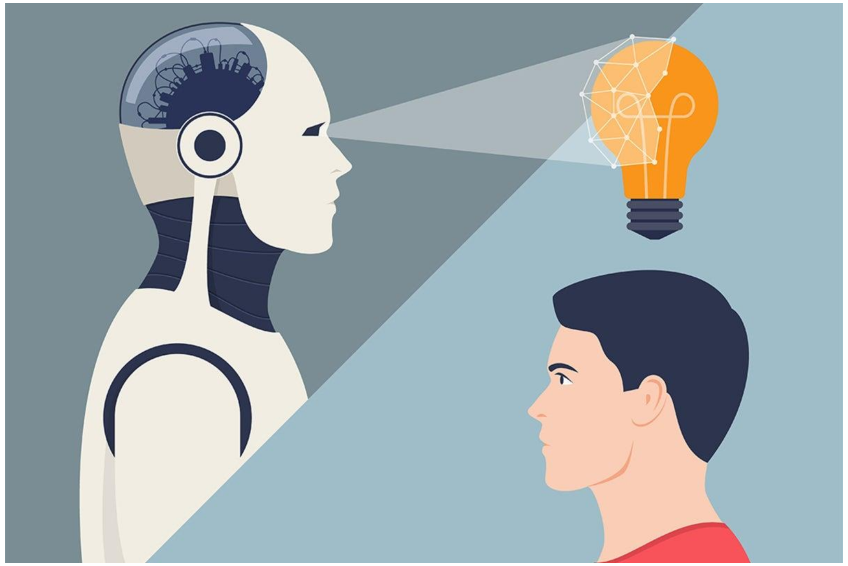
Ethical Implications

- Bias detection
  - Potential disproportionate flagging in language of certain communities/groups
    - Mitigation strategy: bias audit (evaluation of the data set to identify biases) + threshold adjustments (modifying classification threshold based on attributes like race/gender)
  - False positive/negatives + incorrect flagging of non-hateful tweets
    - Lead to false censorship of non-hateful tweets / hateful tweets can bypass moderation (disguised language)
      - Mitigation strategy: human-in-the loop review (human involvement in the AI system to accurately review AI content flags/decisions)
- Reflection: Accuracy vs. Inclusivity
  - Strict moderation → less hate speech, over censorship
  - Lenient moderation → free speech, risk of harmful content bypassing



https://www.titanml.co/glossary/human-in-the-loop

Potential Improvements to Model



https://www.scientificamerican.com/article/humans-absorb-bias-from-ai-and-keep-it-after-they-stop-using-the-algorithm/

- Diverse & Balanced Data – Ensure training data represents different dialects, cultures, and contexts.
- Context-Aware Features – Use n-grams and word embeddings to capture meaning beyond single words.
- Hybrid Model Approach – Combine Naïve Bayes with deep learning (e.g., BERT, LSTMs) for better accuracy.
- Bias Detection & Fairness Checks – Regularly audit model predictions to ensure fair treatment across groups.
- Human-AI Collaboration – Use AI to flag content, but let human moderators review complex cases.

Plan to Finish Project

- Data Collection & Preprocessing – Choose a dataset, clean the text, and convert it into numerical features using CountVectorizer or TfidfVectorizer.
- Model Training & Evaluation – Train a Naïve Bayes, split the data, and evaluate using accuracy, precision, recall, and F1-score.
- Bias & Ethical Analysis – Examine whether the model disproportionately flags certain groups, analyze dataset fairness, and propose improvements.
- Visualization & Reporting – Create performance visualizations and summarize findings in a report
- Presentation Preparation – Develop slides with key insights on model performance, bias, and ethical considerations for the final presentation

Thank You

Any Questions?