Group name: Solitude Ensemble

Group member's details

Name:	Parthkumar Sutariya
Email ID:	parthsutariya31@gmail.com
Country:	India
College:	BSBI, Berlin, Germany
Specialization:	Data Analyst

Problem Description:

- Hate speech detection aims to identify and classify text messages containing hateful, abusive, or discriminatory language.
- The proliferation of hate speech on social media platforms like Twitter poses significant challenges for maintaining a safe and inclusive online environment.
- By developing a hate speech detection system using NLP techniques, we can mitigate the harmful effects of hate speech, promote online civility, and protect vulnerable communities from harassment and discrimination.

Data Understanding:

- The Twitter dataset consists of text data extracted from tweets posted on the platform.
- Each tweet is labeled as either containing hate speech, offensive language, or being non-offensive.
- Data exploration reveals the distribution of different classes within the dataset, potential biases, and trends in the types of hate speech present.

Project Lifecycle:

- **Type of Data:** The Twitter dataset typically includes text content, user information, timestamps, and other metadata associated with each tweet.

- **Data Problems:** Common issues in the data may include missing values (NA values), outliers, class imbalance (skewed distribution of classes), and noisy or uninformative text.
- **Approaches to Address Problems:**
    - Handling NA Values: NA values in text data can be handled by imputation techniques such as filling missing values with placeholders or using advanced imputation algorithms.
    - Outlier Detection: Outliers in text data may not be as straightforward to detect as in numerical data. However, extreme values or anomalies can be identified through text analysis techniques such as frequency analysis, sentiment analysis, or clustering.
    - Class Imbalance: Techniques for handling class imbalance include resampling methods (e.g., oversampling minority classes, under sampling majority classes), using weighted loss functions during model training, or employing ensemble methods that give more weight to minority classes.
    - Text Preprocessing: Preprocessing steps like tokenization, stop word removal, stemming/lemmatization, and punctuation removal can help standardize and clean the text data, making it more suitable for NLP tasks.