Final Report: VeritasVigil — The Truth Watchman

Custom Tokenizer Design

Goals:

- Handle informal text commonly found in online news content.
- Normalize elongated/repeated characters (e.g., sooo → so + REPEAT:3).
- Process emoticons, contractions, and punctuation.
- Lowercase and split into meaningful tokens

A manual dictionary is used to expand common contractions:

```
"don't" → "do not", "it's" → "it is", "i'm" → "i am"
```

Rule-Based POS Tagger

POS	Rule
VERB	Words ending in -ing, -ed
NOUN	Words ending in -tion, -ity, -ment, -ness
ADJ	Words ending in -ous, -ful, -able, -ive

Eg: running -> stemmed to run

.

Custom Lemmatizer

Goals:

- Reduce words to base form considering POS context
- Avoid stemming errors like "better" → "bett"

Methodology:

- VERB:
    - Remove `-ing, -ed` endings (e.g., running → run)
- NOUN:
    - Remove `-ness`

Feature Extraction

Methods Used:

1. **Bag-of-Words (BoW)**:
    - Frequency count of tokens
    - Captures term occurrence
2. **TF-IDF (Term Frequency-Inverse Document Frequency)**:
    - Weighs terms that are frequent in a document but rare in corpus
    - Highlights unique keywords

Classification

Models Trained:

1. **Multinomial Naive Bayes (NB)**:
    - Fast, probabilistic model for text data

2. **Linear Support Vector Machine (SVM)**:
    - Margin-based classifier for high-dimensional spaces

Impact Analysis

Repeated-Character Normalization

- Helps model understand exaggeration and sentiment often present in fake news headlines or user-like stories.
- Improves vocabulary normalization by collapsing redundant variants (e.g., "goooood", "goood", etc.)

POS-Guided Lemmatization

- Prevents over-truncation compared to stemmers.
- Enhances semantic clarity by converting words only when contextually appropriate.