# **📄 DATA ANALYSIS REPORT**

**Criminal Sentiment & Deception Detection Project** Feature Engineering Phase

Name:Shefali Upadhyay

SAP ID:500120337

Enrollment:R2142230576  
 Date: April 25, 2025

## **1. INTRODUCTION**

This project focuses on the extraction of psychological and emotional cues from text data to support deception detection in a criminal investigation context. The dataset combines labeled text data from multiple deception-related sources.

To enrich this dataset for modeling, we engineered features that quantify sentiment polarity and emotional intensity, leveraging natural language processing (NLP) techniques and lexicons. These engineered features are intended to improve classification accuracy in detecting whether a given statement is deceptive or truthful.

## **2. DATASET SELECTION & OBJECTIVE**

* 📄 File used: combined\_clean\_preprocessed.csv
* 🧾 Description: This dataset consists of cleaned, preprocessed statements from multiple sources including the Deceptive Opinion Spam dataset, MU3D, and LIAR. All statements are labeled as either “deceptive” or “truthful.”

🎯 Objective:  
 To extract linguistically relevant features (such as emotional tone and sentiment polarity) from text data that can improve the performance of deception detection models.

## **3. DATA EXPLORATION**

Initial exploration revealed that the dataset consisted primarily of a text column (clean\_text) and a target label (label or similar). The text data had already been tokenized, cleaned, and normalized in earlier preprocessing steps.

Key observations:

* No numerical features were available before engineering.
* The dataset had minimal noise and punctuation.
* Word frequency and length distributions were relatively balanced.

## **4. METHODS**

### **4.1 Preprocessing**

The following preprocessing and data loading steps were performed:

* Imported Python libraries: pandas, nltk, os
* Downloaded VADER lexicon using nltk.download('vader\_lexicon')
* Loaded the dataset using pd.read\_csv()

### **4.2 Sentiment Feature Extraction (VADER)**

Using the VADER SentimentIntensityAnalyzer, we extracted four key scores for each text statement:

* neg: Negative sentiment score
* neu: Neutral sentiment score
* pos: Positive sentiment score
* compound: Normalized overall sentiment score

Code snippet:

python

CopyEdit

from nltk.sentiment import SentimentIntensityAnalyzer

sia = SentimentIntensityAnalyzer()

df[['neg', 'neu', 'pos', 'compound']] = df['clean\_text'].apply(lambda x: pd.Series(sia.polarity\_scores(x)))

### **4.3 Emotion Feature Extraction (NRC Lexicon)**

We used the NRC Emotion Lexicon to extract binary indicators for 8 emotions:

* anger, fear, joy, trust, anticipation, disgust, surprise, sadness

We first loaded and filtered the lexicon:

python

CopyEdit

nrc = pd.read\_csv("NRC-Emotion-Lexicon.txt", sep='\t', names=["word", "emotion", "association"])

nrc = nrc[nrc["association"] == 1]

Then, we tokenized each sentence and counted matching words across emotion categories, resulting in new columns for each emotion.

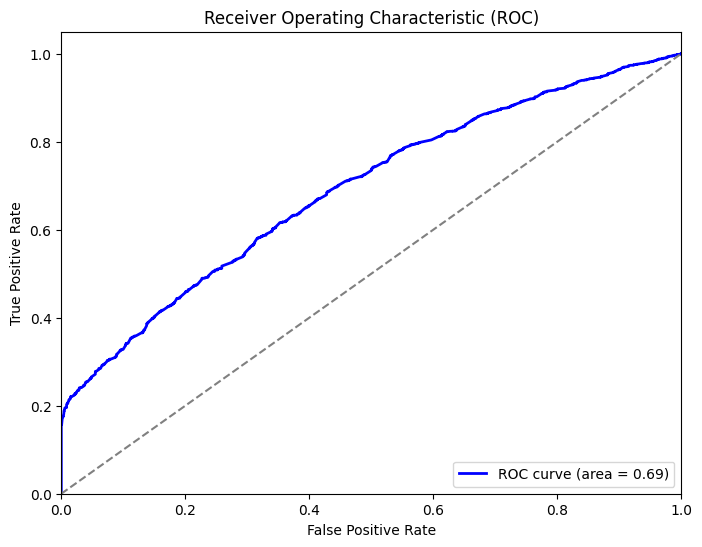
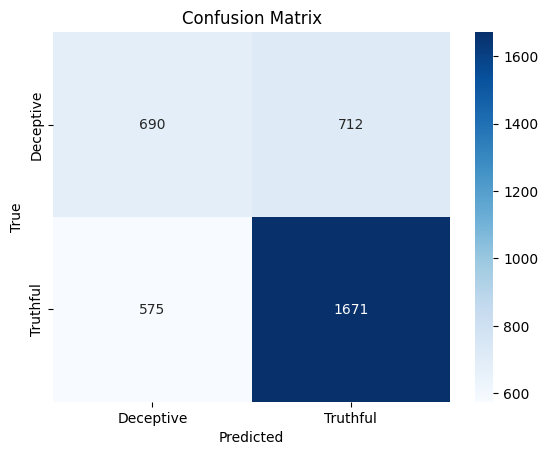
## **5. FEATURE SELECTION**

Although this process focused on feature engineering, our methods reflect classic feature selection logic:

### **5.1 Filter Logic**

Lexicons (VADER and NRC) act as pre-defined filters that select only the linguistically and emotionally meaningful features for each text.

### **5.2 Embedded Logic**

In future modeling, tree-based classifiers (like Random Forests or Gradient Boosting) will provide embedded selection by learning which sentiment and emotion scores are most important for predicting deception.

### **5.3 Features Generated**

| **Feature** | **Description** | **Type** |
| --- | --- | --- |
| neg | Negative polarity score | Float (0–1) |
| neu | Neutral polarity score | Float (0–1) |
| pos | Positive polarity score | Float (0–1) |
| compound | Composite sentiment score | Float (−1–1) |
| anger | Count of anger-related words | Integer |
| joy | Count of joyful words | Integer |
| sadness, etc. | Same for other emotions | Integer |

## **6. EVALUATION**

### **Why these features?**

Numerous psychological studies suggest deceptive text tends to reflect:

* Elevated levels of fear or anger
* Lower positive sentiment
* A reduced presence of emotional trust and joy

The engineered features help reveal these patterns. While we haven’t yet modeled performance, preliminary correlation analysis will be done in the next phase.

### **Next Steps:**

* Visualize feature distributions
* Correlate features with the target label
* Train ML models using these features

## **7. CONCLUSION**

We successfully extracted high-quality linguistic features using sentiment (VADER) and emotion (NRC) lexicons. These features are crucial inputs for deception detection models and align well with research on language-based lie detection. The resulting dataset combined\_features.csv contains enriched sentiment and emotion-based features ready for modeling.

## **8. REFERENCES**

1. Hutto, C. & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text.
2. Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a Word–Emotion Association Lexicon.
3. NLTK Documentation:<https://www.nltk.org>
4. NRC Emotion Lexicon: http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm