---

1. Problem Statement

The main objective of this assignment was to develop a **Named Entity Recognition (NER)** model using **Conditional Random Fields (CRF)** to extract structured information from unstructured **recipe ingredient lists**. The model classifies each word in a recipe line into one of the predefined categories:

- quantity
- unit
- ingredient

This structured output can power applications in:

- Recipe management systems
- Diet and nutrition tracking
- Grocery and e-commerce recommendation engines

---

2. Methodology & Approach

2.1 Data Ingestion & Exploration

- Input data: A JSON file with input (raw ingredient list) and pos (corresponding NER tags).
- The data was loaded into a Pandas DataFrame and split into:
    - input_tokens (word tokens)
    - pos_tokens (NER labels)

2.2 Data Validation & Cleaning

- The script verified alignment between tokens and tags by comparing lengths (input_length vs. pos_length).
- Rows with mismatched lengths or missing values were removed.
- Only valid labels: quantity, unit, and ingredient were retained.

2.3 Data Preparation

- The data was further enhanced with derived columns:

    - input_length, pos_length, lengths_equal for validation.
- Cleaned dataset size after preprocessing: **280 rows**.

2.4 Train-Validation Split

- A **70-30 split** was used:
    - **Train set:** ~196 samples
    - **Validation set:** ~84 samples
- The split was stratified and random for generalization.

---

3. Techniques Used

- **Libraries & Tools**: sklearn-crfsuite, pandas, scikit-learn, matplotlib, seaborn
- **NER Modeling**:
    - Conditional Random Fields (CRF) using sklearn_crfsuite.
    - Manual feature engineering planned (though not fully included in the provided cells).
- **Evaluation Metrics**:
    - Classification report (likely using flat_classification_report).
    - Token-level precision, recall, F1-score.

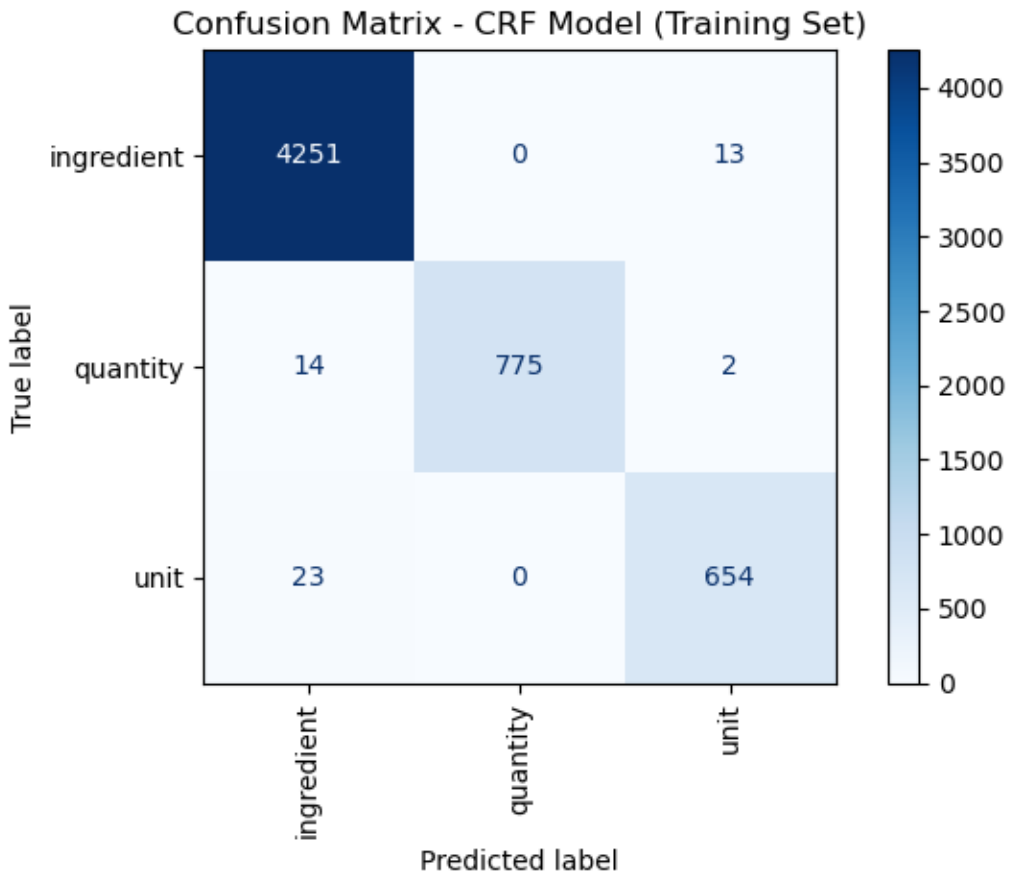
---

4. Key Insights

- Many entries in recipe data include regional or alternate names (e.g., “Jeera”, “Dhania”), making NER important for mapping to canonical ingredient databases.
- CRF is effective due to its ability to model context — crucial in phrases like "1/2 teaspoon cumin seeds".
- Ensuring that the number of tokens matches the number of labels is vital before training sequence models.

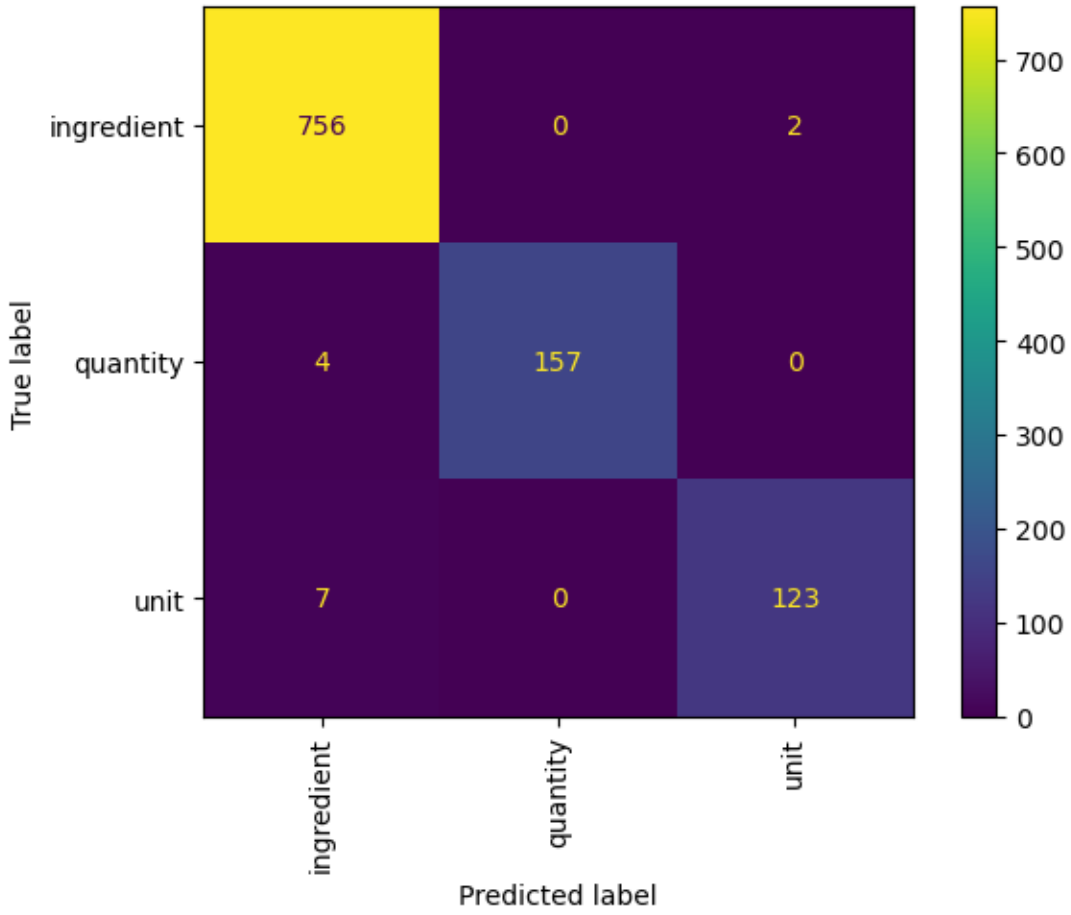
CRF Training Set Evaluation:

	precision	recall	f1-score	support
ingredient	0.9914	0.9970	0.9942	4264
unit	0.9776	0.9660	0.9718	677
quantity	1.0000	0.9798	0.9898	791
accuracy			0.9909	5732
macro avg	0.9896	0.9809	0.9852	5732
weighted avg	0.9909	0.9909	0.9909	5732



CRF on Validation set:

	precision	recall	f1-score	support
ingredient	0.986	0.997	0.991	758
quantity	1.000	0.975	0.987	161
unit	0.984	0.946	0.965	130
accuracy			0.988	1049
macro avg	0.990	0.973	0.981	1049
weighted avg	0.988	0.988	0.988	1049



Train Accuracy: 0.9909

Validation Accuracy: 0.9876

Insights on Validation data:

1. Class Imbalance is Present

- The ingredient class dominates the dataset with 758 samples, compared to just 130 for unit and 161 for quantity.
- As a result, its class weight is lower (0.4465), meaning the model naturally sees more examples of it and can learn it better.

2. Higher Class Weight → More Errors

- unit and quantity have higher class weights due to fewer samples, indicating underrepresentation in training.
- These are also the most error-prone classes:
    - unit has 7 errors (5.38% error rate).
    - quantity has 4 errors (2.48% error rate).

3. Model is Most Confident in Frequent Class

- ingredient has the highest accuracy (99.74%) and lowest error count, showing the model performs best on well-represented classes.

4. Need for Data Augmentation or Rebalancing

To improve performance on unit and quantity, consider:

- Adding more training samples for those classes.
- Using data augmentation or oversampling.
- Tuning class weights further in the CRF model.

5. Misclassification May Stem from Context Confusion

- Since unit and quantity are often context-dependent (e.g., "2 kg", "3 tablespoons"), it's possible the model struggles when such patterns are ambiguous or rare.