Purchase Order Processing Automation - Summary Report

1. Overall Approach

The objective was to automate the processing and analysis of purchase order data received through emails. The pipeline was built to extract relevant information, structure the data, detect duplicates, and generate insights. The steps are as follows:

1. Extract text from email content.
2. Extract structured information from the text.
3. Match and align product names with item codes.
4. Detect duplicate orders using both exact and fuzzy matching.
5. Generate reports and dashboards with statistics and visualizations.

2. Approaches Tried

Initially, regular expressions (regex) were used to extract structured information such as quantity, product name, and item code. Regex works well when the input format is highly consistent. For example:
- Pattern: `r'(\d+) units of ([\w\s]+) \(Item Code: ([A-Z]{2}-\d{4})\)'`
This pattern can extract 3 groups: quantity, product name, and item code.

However, regex-based extraction faced several limitations:
- It is sensitive to even small format variations.
- It breaks when product names span multiple words or are inconsistent.
- It lacks context-awareness, leading to false positives or missed extractions.

3. Working Methodology of spaCy

To overcome the limitations of regex, spaCy NLP was adopted for more flexible and context-aware extraction. The spaCy pipeline includes the following:

- Tokenization: Text is split into tokens (words, punctuation, numbers).
- Named Entity Recognition (NER): Recognizes numeric quantities and potential product names.
- Custom Rule Matching: spaCy's Matcher was used to define rules like:
  e.g., a number followed by 'units' followed by a noun phrase for product name.
- Contextual Grouping: Entities from the same sentence are grouped to form a complete record (quantity, product name, item code).
- Flexibility: spaCy handles variations in sentence structure, synonyms, and order of elements better than regex.

4. Key Challenges and How They Were Addressed

• **Challenge 1:** Email formats varied widely, making it hard for regular expressions to consistently extract relevant data.
→ **Solution:** spaCy's NLP pipeline allowed for semantic understanding and more adaptive pattern matching, even when word order changed.

• **Challenge 2:** Product names and item codes were often mismatched or shuffled.
→ **Solution:** Matching based on initials (e.g., 'Hydraulic Pump' → 'HP') was used to align product names with their item codes when standard matching failed.

• **Challenge 3:** Duplicate orders were submitted multiple times with slight variations, such as whitespace or reordered items or sometimes the order units were written twice consecutively.
→ **Solution:** Used a two-tiered strategy—exact matching for identical rows and fuzzy matching (using token sort ratio and partial ratio from fuzzywuzzy) on key fields like product name, quantity, and customer name. → Explained more detailed in the report.

• **Challenge 4:** Extracted data lacked consistency in naming conventions and spelling errors.
→ **Solution:** Applied fuzzy clustering logic and basic normalization (lowercasing, trimming, spelling corrections) to improve comparability.

5. Suggestions for Further Improvements

- Integrate with a live email API to auto-fetch and process incoming orders in real-time.
- Enable feedback and manual review workflows for flagged duplicates to improve accuracy.
- Introduce multilingual and locale-aware NLP models for global customer handling.
- Include OCR (Optical Character Recognition) for processing scanned image-based purchase orders.
- Build a web-based dashboard for monitoring trends, reviewing flagged orders, and exporting structured reports.
- Create a Chatbot which can read all the emails and can answer the questions asked by the users with visuals and graphs.