Shipping Calculator AI Requirement Document

Objective

Build an automated shipping calculator that processes Amazon order reports (CSV files), aggregates multi-item orders, computes the shipping weight, and calculates the shipping cost per order based on product variants and destination state rates.

Data Overview

1. Amazon Order Report CSV

- **Rows:** Each row represents an order line item.
- **Order IDs:** If an order has multiple items, the same order ID appears in multiple rows.
- **Key columns needed:**
- `Order ID` (or equivalent): Unique identifier for each order.
- `Product Name` or `Product Variant`: Contains text such as "Pack of One" or "Pack of Two".
- `Quantity`: Number of this item in the order line (default 1 unless otherwise specified).
- `State`: Shipping state (destination for rate lookup).

2. Product Variants and Weights

- **Pack of One**: 500 grams (0.5 kg)
- **Pack of Two**: 2 x 500 grams = 1 kg

Shipping Rates Table

- Provide a mapping of states to per-kg shipping rates.
- Example: `{ "Karnataka": 50, "Tamil Nadu": 60, ... }`
- Optionally, set a default rate for states not in the list.

Shipping Calculation Logic

1. **Aggregate Orders:**
2. Group CSV rows by `Order ID`.

3. For each order, sum up the total quantity of each variant ("Pack of One", "Pack of Two").

4. **Calculate Total Weight:**

5. For each group (order):
    - Total weight = (`Pack of One` qty × 0.5kg) + (`Pack of Two` qty × 1kg)
6. If quantity column exists, multiply accordingly.

7. **Determine Destination State:**

8. Extract the state from the order's shipping address or state column.

9. **Apply Shipping Rate:**

10. Use the state's per-kg rate from the rates table.

11. Shipping cost = **ceil(Total Weight in kg) × Rate**

    - If the weight is fractional, always round up to the next full kg (e.g., 1.3kg → 2kg).

12. **Handle Special Rules:**

13. If there's a minimum shipping charge, apply it.

14. If any orders qualify for free shipping, skip calculation (flagged by another column or business rule).

15. **Final Output:**

16. For each order, output:
    - `Order ID`
    - `State`
    - `Total Weight (kg)`
    - `Shipping Rate (per kg)`
    - `Shipping Cost`
17. Provide summary: **Total shipping cost** for all orders.

Edge Cases

- **Multiple items (different variants) in one order:** Add up all weights before calculating shipping.
- **Unrecognized variants:** Either flag as error or skip (log for review).
- **Missing state or rate:** Use default rate or flag for manual review.
- **Quantity > 1:** Multiply pack weight accordingly.

Input/Output Format

- **Input:** CSV file (Amazon order report), and a shipping rate table (CSV, JSON, or dict).
- **Output:** CSV or table with calculated shipping per order, plus summary stats.

Sample Workflow

1. **Upload**: User uploads the Amazon order CSV and (optionally) a shipping rate table.
2. **Parse**: Read all rows, extract required columns.
3. **Aggregate**: Group by order, sum weights.
4. **Calculate**: Look up state rate, compute cost per order.
5. **Export**: Output result as CSV or downloadable table.

User Prompt for the AI Agent

You are an expert AI assistant for e-commerce logistics automation. Please build a “Shipping Calculator” workflow that takes as input an Amazon order report CSV (where each row represents an item, and orders with multiple items have repeated order IDs), and outputs the shipping cost per order based on the following requirements:

1. **Product Variants:**
    - Each product is sold as either "Pack of One" (500g, i.e., 0.5kg) or "Pack of Two" (1kg, i.e., 2 x 500g).
    - The product name or variant field will always contain either "Pack of One" or "Pack of Two".
2. **Order Aggregation:**
    - Aggregate all rows with the same Order ID.
    - For each order, sum the total number of “Pack of One” and “Pack of Two” items (multiply by quantity if the field exists).
3. **Weight Calculation:**
    - Total weight per order = (Number of “Pack of One” × 0.5kg) + (Number of “Pack of Two” × 1kg).
    - Always round up the total weight to the next whole kilogram (e.g., 1.1kg → 2kg).
4. **Shipping Rate Lookup:**
    - Shipping rates are defined per state (provide a mapping as input).
    - For each order, look up the shipping rate using the destination state.
    - Shipping cost per order = (rounded weight in kg) × (rate for the state).
5. **Output:**
    - For each order, output: Order ID, State, Total Weight (kg), Shipping Rate (per kg), Shipping Cost.
    - At the end, provide the total shipping cost for all orders.
    - Export results as a CSV.

6. **Edge Cases:**

    - If state is missing or not found, use a default rate (provided as input or specified in logic).
    - If product variant cannot be identified, flag for review and skip calculation for that row.
    - If an order qualifies for free shipping (optional, rule-based), output shipping cost as zero.

7. **Other Requirements:**

    - The solution should be easily reusable for monthly uploads.
    - The solution must handle 1000+ orders efficiently.
    - If possible, make it easy for the user to upload their own rate card.

Given a sample CSV and a rate card, implement this in Python (or another language, if requested), and provide the user with a script, notebook, or no-code workflow to process new files easily.

Example Table (for Output)

Order ID	State	Pack of One	Pack of Two	Total Weight (kg)	Rounded Weight (kg)	Rate per kg	Shipping Cost
123	KA	1	2	2.5	3	50	150

Attachments for AI

- Attach a sample Amazon order CSV
- Attach a state-wise shipping rate table (CSV, JSON, or as a Python dictionary)

If you have any custom rules (like minimum shipping, free shipping, or bundled orders), add them to the prompt as needed.