Label Preparation and Final Dataset Creation (Part 4)

Objective

This notebook handles the final piece of the puzzle: the "Labels" (Target Variables). In Part 3, we built the features (inputs). Now, we need to structure the crop yield data so the model knows what to predict.

Key Steps:

1. **Load Data:** Import the features from Part 3 and the raw yield data.
2. **Clean & Pivot:** Convert the yield data from "long" format (rows) to "wide" format (columns).
3. **Merge:** Combine the Features (X) and Labels (Y) into one final dataset.
4. **Save:** Export the ready-to-use dataset as `XY_v2.parquet`.

In [1]:
```
import pandas as pd
import numpy as np
```

1. Load Features and Raw Labels

We import our feature set (`x_features_v2.parquet`) and the raw crop yield data (`label_yield.parquet`). The raw yield data is currently in a "long" format, which means different crops are stacked in rows.

In [2]:
```
# Load Features created in Part 3
X = pd.read_parquet('Parquet/x_features_v2.parquet')

# Load Raw Yield Data
label_yield = pd.read_parquet('Parquet/label_yield.parquet')
```

2. Standardize Crop Names

Machine learning models prefer clean, consistent names. We process the `item` column to remove special characters and spaces, converting names like "Maize (corn)" to `maize_corn`. This prevents errors during column creation.

In [3]:
```
label_yield['item'] = label_yield['item'].str.replace(r'[^0-9a-zA-Z ]', '', rege
label_yield['item'] = label_yield['item'].str.replace(" ", "_").str.lower()

# Display sample to verify cleaning
label_yield.head()
```

Out[3]:

	area	item	year	label
0	Afghanistan	maize_corn	1970-12-31	1475.7
1	Afghanistan	maize_corn	1971-12-31	1340.0
2	Afghanistan	maize_corn	1972-12-31	1565.2
3	Afghanistan	maize_corn	1973-12-31	1617.0
4	Afghanistan	maize_corn	1974-12-31	1617.0

3. Create Target Columns (Pivoting)

We need a separate column for each crop so we can predict them individually.

- **Current Format (Long):** One row per crop per year.
- **New Format (Wide):** One row per year, with columns like `Y_rice`, `Y_wheat`, `Y_maize`.

We use the `pivot_table` function to transform the data structure.

In [4]:
```
# Extract Year as integer
label_yield['year'] = pd.to_datetime(label_yield['year']).dt.year

# Pivot the table
Y = label_yield.pivot_table(
    index=['year','area'],  # Unique identifier for row
    columns='item',         # Create columns for each crop
    values='label'          # The Yield value
).reset_index()
```

4. Rename Columns

To avoid confusion between our *features* (inputs) and our *targets* (outputs), we add a prefix `Y_` to all the new crop columns. For example, the target for Rice becomes `Y_rice`.

In [5]:
```
# Dynamically generate column list
current_cols = Y.columns.tolist()

# Identify crop columns (those that represent items)
crop_cols = [c for c in current_cols if c not in ['year', 'area']]

# Create new column mapping
new_col_names = ['year', 'area'] + [f'Y_{c}' for c in crop_cols]
Y.columns = new_col_names

# Display structure
Y.head()
```

Out[5]:

	year	area	Y_bananas	Y_barley	Y_cassava_fresh	Y_cucumbers_and_g
0	1970	Afghanistan	NaN	1174.6	NaN	
1	1970	Albania	NaN	1077.8	NaN	
2	1970	Algeria	NaN	668.5	NaN	
3	1970	Angola	10000.0	NaN	3555.6	
4	1970	Antigua_and_Barbuda	1500.0	NaN	4000.0	

5. Merge Features and Labels

Now we combine everything. We perform an **inner join** between our Feature table (X) and our Label table (Y) based on `year` and `area`. This ensures every row in our final dataset has both input features and a target value to learn from.

In [6]:
```
XY = X.merge(Y, on=['year', 'area'], how='inner')

# Output the shape to verify merge
print(f"Final dataset shape: {XY.shape}")
```

```
Final dataset shape: (6631, 81)
```

6. Save Final Dataset

We export the merged dataframe to `XY_v2.parquet`. This file contains everything needed to train the model in the next step.

In [7]:
```
# Save to Parquet
XY.to_parquet('Parquet/XY_v2.parquet')
```

In [8]:
```
# Prevent pandas from hiding columns
pd.set_option('display.max_columns', None)

# Show first 20 rows for Thailand
XY[XY['area'] == 'Thailand'].head(20)
```

Out[8]:

	year	area	avg_yield_maize_corn_1y	avg_yield_maize_corn_3y	avg_yield_maize
5818	1982	Thailand	2353.8	2197.466667	
5819	1983	Thailand	2298.8	2293.633333	
5820	1984	Thailand	2267.4	2306.666667	
5821	1985	Thailand	2430.5	2332.233333	
5822	1986	Thailand	2571.9	2423.266667	
5823	1987	Thailand	2373.8	2458.733333	
5824	1988	Thailand	2048.6	2331.433333	
5825	1989	Thailand	2617.6	2346.666667	
5826	1990	Thailand	2569.0	2411.733333	
5827	1991	Thailand	2409.0	2531.866667	
5828	1992	Thailand	2711.7	2563.233333	
5829	1993	Thailand	2970.9	2697.200000	
5830	1994	Thailand	2733.3	2805.300000	
5831	1995	Thailand	2934.4	2879.533333	
5832	1996	Thailand	3288.4	2985.366667	
5833	1997	Thailand	3447.8	3223.533333	
5834	1998	Thailand	3198.2	3311.466667	
5835	1999	Thailand	3344.8	3330.266667	
5836	2000	Thailand	3552.6	3365.200000	
5837	2001	Thailand	3671.5	3522.966667	