Mini-Project: Farmer Input Purchase Analysis Project

Task: Using a relevant dataset of your choice containing farmer input purchase records, preprocess the dataset and implement two association rule mining algorithms to identify frequent combinations of inputs and propose recommendations for bundled input sales to increase efficiency.

In [1]:
```
#import libraries
import pandas as pd
import numpy as np
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, fpgrowth, associatio
import time
```

Since getting similar dataset as prescribed was difficult, we simulate our dataset.

In [2]:
```
# Create synthetic agricultural transactions
def create_dataset():
    data = []
    items = ['Maize', 'Wheat', 'Soybeans', 'DAP', 'Urea', 'CAN', 'H

    for _ in range(1000):
        transaction = []
        # Bias: Maize farmers almost always buy DAP and Herbicide
        if np.random.rand() < 0.4:
            transaction.extend(['Maize', 'DAP', 'Herbicide'])
        # Bias: Wheat farmers often buy Urea and Fungicide
        elif np.random.rand() < 0.3:
            transaction.extend(['Wheat', 'Urea', 'Fungicide'])
        # Add some random items
        num_extras = np.random.randint(1, 3)
        transaction.extend(np.random.choice(items, num_extras).toli
        data.append(list(set(transaction))) # unique items per tran
    return data

dataset = create_dataset()
```

Preprocessing the dataset

In [3]:
```
# one-hot encode the dataset
te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
#print dataset shape
print("Dataset shape:", df.shape)
```

```
# print first 5 rows of the dataset
print("First 5 rows of the dataset:")
print(df.head())
```

```
Dataset shape: (1000, 9)
First 5 rows of the dataset:
     CAN    DAP  Fungicide  Herbicide  Maize  Pesticide  Soybeans 
Urea  \
0  False   True      False       True   True      False      True  F
alse
1  False   True      False      False   True      False     False  F
alse
2  False   True      False       True   True      False     False   
True
3  False  False      False      False   True      False     False  F
alse
4  False  False       True       True  False      False     False   
True

   Wheat
0  False
1  False
2   True
3  False
4   True
```

In [4]:

```
# apply apriori algorithm and fpgrowth algorithm
# Algorithm 1: Apriori
start_time = time.time()
frequent_itemsets_apriori = apriori(df, min_support=0.1, use_colnam
apriori_time = time.time() - start_time
# print(frequent_itemsets_apriori)
rules_apriori = association_rules(frequent_itemsets_apriori, metric
# display top 10 rules and key columns
top_rules_apriori = rules_apriori.sort_values('confidence', ascendi
print("Top 10 Apriori Rules:")
print(top_rules_apriori[['antecedents', 'consequents', 'support', '
```

```
Top 10 Apriori Rules:
             antecedents  consequents  support  confidence      lift
12          (DAP, Maize)  (Herbicide)    0.411    0.985612  1.917532
13      (DAP, Herbicide)      (Maize)    0.411    0.985612  1.987120
14    (Maize, Herbicide)        (DAP)    0.411    0.983254  1.994429
18    (Fungicide, Wheat)       (Urea)    0.201    0.961722  3.082444
20         (Wheat, Urea)  (Fungicide)    0.201    0.957143  2.900433
19     (Fungicide, Urea)      (Wheat)    0.201    0.943662  2.868273
0                  (DAP)  (Herbicide)    0.417    0.845842  1.645607
2                  (DAP)      (Maize)    0.417    0.845842  1.705326
8                (Maize)  (Herbicide)    0.418    0.842742  1.639576
3                (Maize)        (DAP)    0.417    0.840726  1.705326
```

In [5]:

```
# Algorithm 2: FP-Growth
start_time = time.time()
frequent_itemsets_fp = fpgrowth(df, min_support=0.1, use_colnames=T
fpgrowth_time = time.time() - start_time

# Generate association rules from FP-Growth frequent itemsets
```

```
rules_fpgrowth = association_rules(frequent_itemsets_fp, metric="li
# Sorting by confidence to find the most 'certain' bundles
top_rules = rules_fpgrowth.sort_values('confidence', ascending=Fals

# Displaying key columns
print(top_rules[['antecedents', 'consequents', 'support', 'confiden

# print execution times
print(f"Apriori Time: {apriori_time:.4f}s")
print(f"FP-Growth Time: {fpgrowth_time:.4f}s")
```

```
              antecedents  consequents  support  confidence      lift
6            (DAP, Maize)  (Herbicide)    0.411    0.985612  1.917532
7        (DAP, Herbicide)      (Maize)    0.411    0.985612  1.987120
8      (Maize, Herbicide)        (DAP)    0.411    0.983254  1.994429
18     (Fungicide, Wheat)       (Urea)    0.201    0.961722  3.082444
20          (Wheat, Urea)  (Fungicide)    0.201    0.957143  2.900433
19      (Fungicide, Urea)      (Wheat)    0.201    0.943662  2.868273
2                   (DAP)      (Maize)    0.417    0.845842  1.705326
4                   (DAP)  (Herbicide)    0.417    0.845842  1.645607
0                 (Maize)  (Herbicide)    0.418    0.842742  1.639576
3                 (Maize)        (DAP)    0.417    0.840726  1.705326
Apriori Time: 0.0052s
FP-Growth Time: 0.0045s
```

Strategic Recommendations for Bundling

Here are the prooposed bundles from the outputs:

Bundle Name	Items Included	Recommendations
Planting bundle	Maize + DAP + Herbicide	Found in over 20% of transactions with > 97% confidence. Reduces stock-out risk for critical items.
Wheat Protection bundle	Wheat + Urea + Fungicide	Strong association (Lift > 3.0). Simplifies the supply chain for cereal farmers.
Top-Dressing bundle	DAP + Maize	High support (0.399) indicates farmers often return for pest control when buying nitrogen boosters.