```
import gdown
file_id = '1MCmntA3BOquC7BWxLcYDItohDgTPRKx8'
url = f'https://drive.google.com/uc?id={file_id}'
output = 'Crop_Data_Final.csv'
gdown.download(url, output, quiet=False)
```

```
Downloading...
From: https://drive.google.com/uc?id=1MCmntA3BOquC7BWxLcYDItohDgTPRKx8
To: /content/Crop_Data_Final.csv
100%|██████████| 214k/214k [00:00<00:00, 48.6MB/s]
```

```
num_itemsets = frequent_itemsets['itemsets'].apply(len).value_counts().to_dict()
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.5, num_itemsets
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-3cf694829650> in <cell line: 0>()
----> 1 num_itemsets = frequent_itemsets['itemsets'].apply(len).value_counts().to_dict()
      2 rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.5, num_itemsets=num_itemsets)

NameError: name 'frequent_itemsets' is not defined
```

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
from sklearn.preprocessing import KBinsDiscretizer

# Load dataset
file_path = '/content/Crop_Data_Final.csv'
data = pd.read_csv(file_path)

# Districts to process
districts = data['Dist Name'].unique()

# Binarization thresholds
binarize_columns = [
    'RICE YIELD (Kg per ha)', 'WHEAT YIELD (Kg per ha)', 'SORGHUM YIELD (Kg per ha)',
    'PEARL MILLET YIELD (Kg per ha)', 'MAIZE YIELD (Kg per ha)', 'CHICKPEA YIELD (Kg per ha)',
    'PIGEONPEA YIELD (Kg per ha)', 'GROUNDNUT YIELD (Kg per ha)', 'SESAMUM YIELD (Kg per ha)',
    'OILSEEDS YIELD (Kg per ha)', 'SUGARCANE YIELD (Kg per ha)', 'COTTON YIELD (Kg per ha)',
    'Precipitation (mm)', 'Annual Rainfall'
]

# Function to discretize continuous variables
def discretize_data(df, columns):
    discretizer = KBinsDiscretizer(n_bins=2, encode='ordinal', strategy='uniform')
    df = df.copy()  # Prevent SettingWithCopyWarning
    for col in columns:
        if col in df.columns:
            df[col + '_Binned'] = discretizer.fit_transform(df[[col]]).astype(int)
    return df

# Function to generate transactions for Apriori
def generate_transactions(df):
    transactions = pd.DataFrame()
    for col in df.columns:
        if col.endswith('_Binned'):
            transactions[col + '_High'] = (df[col] == 1).astype(int)
```

```
            transactions[col + '_Low'] = (df[col] == 0).astype(int)
    return transactions

# Function to process each district
def process_district(district):
    # Filter district-specific data and discretize
    district_data = data[data['Dist Name'] == district].copy()
    district_data = discretize_data(district_data, binarize_columns)

    # Generate transactions
    transactions = generate_transactions(district_data)

    # Run Apriori algorithm
    frequent_itemsets = apriori(transactions, min_support=0.2, use_colnames=True)
    num_itemsets = frequent_itemsets['itemsets'].apply(len).value_counts().to_dict()
    rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.5, num_item
    # rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.0)

    # Add district name to results
    rules['District'] = district
    return rules

# Process all districts and collect results
all_rules = []
for district in districts:
    try:
        district_rules = process_district(district)
        all_rules.append(district_rules)
    except Exception as e:
        print(f"Error processing district {district}: {e}")

# Combine results for all districts
all_rules_df = pd.concat(all_rules, ignore_index=True)

# Save results to a CSV file
output_path = '/content/district_apriori_rules.csv'
all_rules_df.to_csv(output_path, index=False)
print(f"Apriori results saved to {output_path}")
```

```
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Error processing district Ahmednagar: unsupported operand type(s) for -: 'dict' and 'int'
Error processing district Akola: unsupported operand type(s) for -: 'dict' and 'int'
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/sklearn/preprocessing/_discretization.py:262: UserWarning: Feature 0 is constant and will b
  warnings.warn(
Error processing district Amarawati: unsupported operand type(s) for -: 'dict' and 'int'
Error processing district Aurangabad: unsupported operand type(s) for -: 'dict' and 'int'
Error processing district Beed: unsupported operand type(s) for -: 'dict' and 'int'
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/sklearn/preprocessing/_discretization.py:262: UserWarning: Feature 0 is constant and will b
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Error processing district Bhandara: unsupported operand type(s) for -: 'dict' and 'int'
Error processing district Buldhana: unsupported operand type(s) for -: 'dict' and 'int'
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Error processing district Chandrapur: unsupported operand type(s) for -: 'dict' and 'int'
Error processing district Dhule: unsupported operand type(s) for -: 'dict' and 'int'
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Error processing district Jalgaon: unsupported operand type(s) for -: 'dict' and 'int'
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Error processing district Kolhapur: unsupported operand type(s) for -: 'dict' and 'int'
Error processing district Nagpur: unsupported operand type(s) for -: 'dict' and 'int'
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Error processing district Nanded: unsupported operand type(s) for -: 'dict' and 'int'
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Error processing district Nasik: unsupported operand type(s) for -: 'dict' and 'int'
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Error processing district Osmanabad: unsupported operand type(s) for -: 'dict' and 'int'
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Error processing district Parbhani: unsupported operand type(s) for -: 'dict' and 'int'
Error processing district Pune: unsupported operand type(s) for -: 'dict' and 'int'
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Error processing district Sangli: unsupported operand type(s) for -: 'dict' and 'int'
Error processing district Satara: unsupported operand type(s) for -: 'dict' and 'int'
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Error processing district Solapur: unsupported operand type(s) for -: 'dict' and 'int'
Error processing district Yeotmal: unsupported operand type(s) for -: 'dict' and 'int'
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-3-ffa94f026c43> in <cell line: 0>()
     66
     67 # Combine results for all districts
---> 68 all_rules_df = pd.concat(all_rules, ignore_index=True)
     69
     70 # Save results to a CSV file

                                 2 frames
/usr/local/lib/python3.11/dist-packages/pandas/core/reshape/concat.py in _clean_keys_and_objs(self, objs, keys)
    505
    506         if len(objs_list) == 0:
--> 507             raise ValueError("No objects to concatenate")
    508
    509         if keys is None:
```

```
    ValueError: No objects to concatenate
```

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
from sklearn.preprocessing import KBinsDiscretizer

# Load dataset
file_path = '/content/Crop_Data_Final.csv'
data = pd.read_csv(file_path)

# Districts to process
districts = data['Dist Name'].unique()

# Binarization thresholds
binarize_columns = [
    'RICE YIELD (Kg per ha)', 'WHEAT YIELD (Kg per ha)', 'SORGHUM YIELD (Kg per ha)',
    'PEARL MILLET YIELD (Kg per ha)', 'MAIZE YIELD (Kg per ha)', 'CHICKPEA YIELD (Kg per ha)',
    'PIGEONPEA YIELD (Kg per ha)', 'GROUNDNUT YIELD (Kg per ha)', 'SESAMUM YIELD (Kg per ha)',
    'OILSEEDS YIELD (Kg per ha)', 'SUGARCANE YIELD (Kg per ha)', 'COTTON YIELD (Kg per ha)',
    'Precipitation (mm)', 'Annual Rainfall'
]

# Function to discretize continuous variables
def discretize_data(df, columns):
    discretizer = KBinsDiscretizer(n_bins=2, encode='ordinal', strategy='uniform')
    df = df.copy()  # Prevent SettingWithCopyWarning
    for col in columns:
        if col in df.columns:
            df[col + '_Binned'] = discretizer.fit_transform(df[[col]]).astype(int)
    return df

# Function to generate transactions for Apriori
def generate_transactions(df):
    transactions = pd.DataFrame()
    for col in df.columns:
        if col.endswith('_Binned'):
            transactions[col + '_High'] = (df[col] == 1).astype(int)
            transactions[col + '_Low'] = (df[col] == 0).astype(int)
    return transactions

# Function to process each district
def process_district(district):
    # Filter district-specific data and discretize
    district_data = data[data['Dist Name'] == district].copy()
    district_data = discretize_data(district_data, binarize_columns)

    # Generate transactions
    transactions = generate_transactions(district_data)

    # Run Apriori algorithm
    frequent_itemsets = apriori(transactions, min_support=0.2, use_colnames=True)
    num_itemsets = frequent_itemsets['itemsets'].apply(len).value_counts().to_dict()
    rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.5)
```

```
    # Add district name to results
    rules['District'] = district

    # Save rules for this district to its own CSV
    safe_district_name = district.replace(' ', '_').replace('/', '_')
    output_path = f'/content/{safe_district_name}_apriori_rules.csv'
    rules.to_csv(output_path, index=False)
    print(f"Saved rules for district: {district} to {output_path}")

    return rules

# Process all districts and collect results
all_rules = []
for district in districts:
    try:
        district_rules = process_district(district)
        all_rules.append(district_rules)
    except Exception as e:
        print(f"Error processing district {district}: {e}")

# Combine results for all districts (optional)
all_rules_df = pd.concat(all_rules, ignore_index=True)

# Save combined results to a CSV file
output_path = '/content/district_apriori_rules_combined.csv'
all_rules_df.to_csv(output_path, index=False)
print(f"Combined Apriori results saved to {output_path}")
```

```
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Ahmednagar to /content/Ahmednagar_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Akola to /content/Akola_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Amarawati to /content/Amarawati_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/sklearn/preprocessing/_discretization.py:262: UserWarning: Feature 0 is constant and will b
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/association_rules.py:186: RuntimeWarning: invalid value encounter
  cert_metric = np.where(certainty_denom == 0, 0, certainty_num / certainty_denom)
Saved rules for district: Aurangabad to /content/Aurangabad_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Beed to /content/Beed_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/sklearn/preprocessing/_discretization.py:262: UserWarning: Feature 0 is constant and will b
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/association_rules.py:186: RuntimeWarning: invalid value encounter
  cert_metric = np.where(certainty_denom == 0, 0, certainty_num / certainty_denom)
Saved rules for district: Bhandara to /content/Bhandara_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Buldhana to /content/Buldhana_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Chandrapur to /content/Chandrapur_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Dhule to /content/Dhule_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Jalgaon to /content/Jalgaon_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Kolhapur to /content/Kolhapur_apriori_rules.csv
```

```
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Nagpur to /content/Nagpur_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Nanded to /content/Nanded_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Nasik to /content/Nasik_apriori_rules.csv
/usr/local/lib/python3.11/dist-packages/mlxtend/frequent_patterns/fpcommon.py:161: DeprecationWarning: DataFrames with non-bool typ
  warnings.warn(
Saved rules for district: Osmanabad to /content/Osmanabad_apriori_rules.csv
```