```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.cluster import KMeans, DBSCAN
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

# Load dataset
data = pd.read_csv('crop.csv')

# Preprocessing
data = data.drop(['Yield', 'Crop_Year'], axis=1)
data.dropna(subset=['Production'], inplace=True)

# Label encoding
le = LabelEncoder()
data['State'] = le.fit_transform(data['State'])
data['District'] = le.fit_transform(data['District'])
data['Crop'] = le.fit_transform(data['Crop'])
data['Season'] = le.fit_transform(data['Season'])

# Clustering with K-means algorithm
kmeans = KMeans(n_clusters=3, random_state=0)
kmeans.fit(data)
data['kmeans_cluster'] = kmeans.labels_

# Clustering with DBSCAN algorithm
dbscan = DBSCAN(eps=0.5, min_samples=5)
dbscan.fit(data)
data['dbscan_cluster'] = dbscan.labels_

# Association analysis with Apriori algorithm
one_hot_encoded_data = pd.get_dummies(data, columns=['State', 'District', 'Crop',
'Season'])
one_hot_encoded_data = one_hot_encoded_data.apply(lambda x: np.where(x > 0, 1, 0))
one_hot_encoded_data = one_hot_encoded_data.astype(int)
frequent_itemsets = apriori(one_hot_encoded_data, min_support=0.1,
use_colnames=True)
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)

# Print the results
print("K-means clustering:")
print(data['kmeans_cluster'].value_counts())
print("\nDBSCAN clustering:")
print(data['dbscan_cluster'].value_counts())
print("\nAssociation rules:")
print(rules.head())
```