DATASET TAKEN – BASKET.CSV(500 data rows)

CODE:

import pandas as pd

import numpy as np

from mlxtend.frequent\_patterns import apriori, association\_rules

import matplotlib.pyplot as plt

from sklearn.tree import DecisionTreeClassifier,plot\_tree

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

#import seaborn as sns

data = pd.read\_csv('basket.csv')

data=data.head(500)

data=data.fillna('None',inplace=False)

items=data['0'].unique()

itemset = set(items)

encoded\_vals = []

for index, row in data.iterrows():

rowset = set(row)

labels = {}

uncommons = list(itemset - rowset)

commons = list(itemset.intersection(rowset))

for uc in uncommons:

labels[uc] = 0

for com in commons:

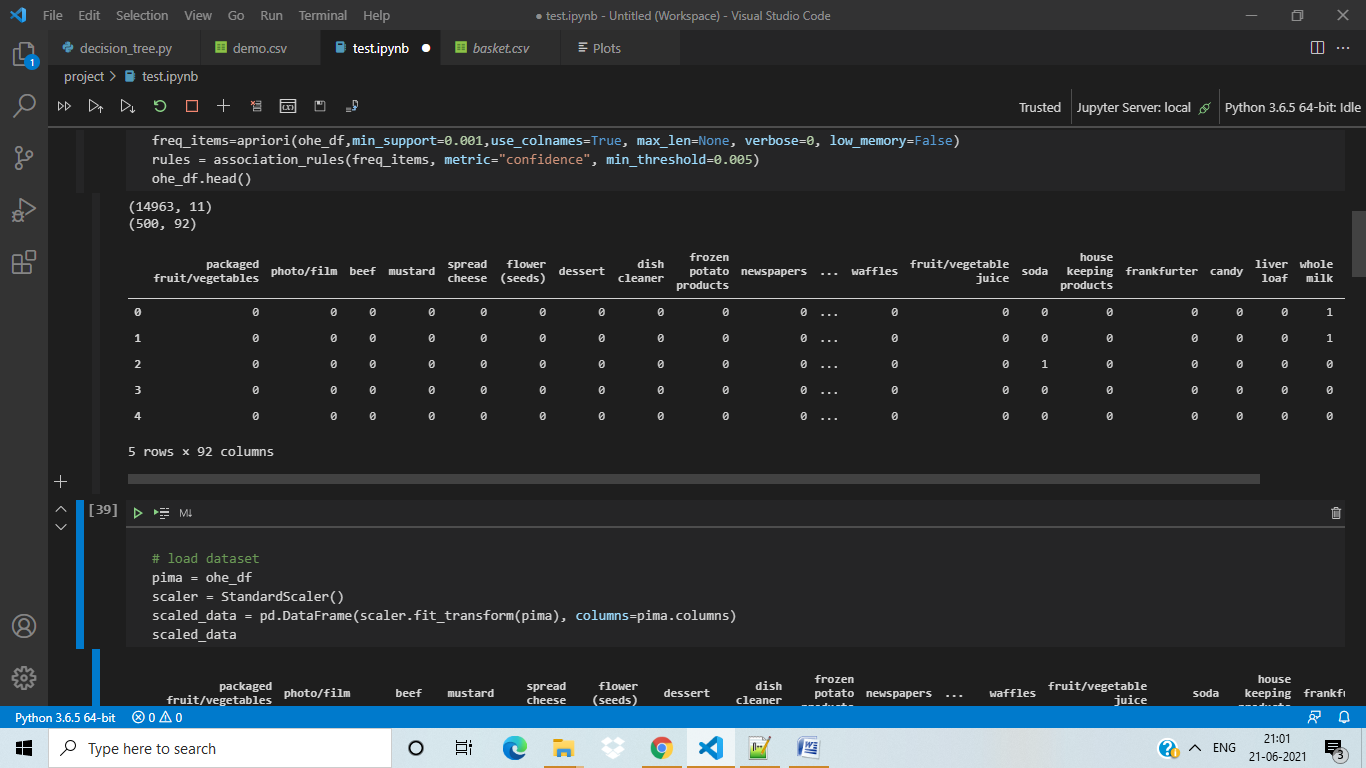
labels[com] = 1

encoded\_vals.append(labels)

ohe\_df = pd.DataFrame(encoded\_vals)

ohe\_df.head()

OUTPUT



----------------------------------------------------------------------

#scaling and pca analysis

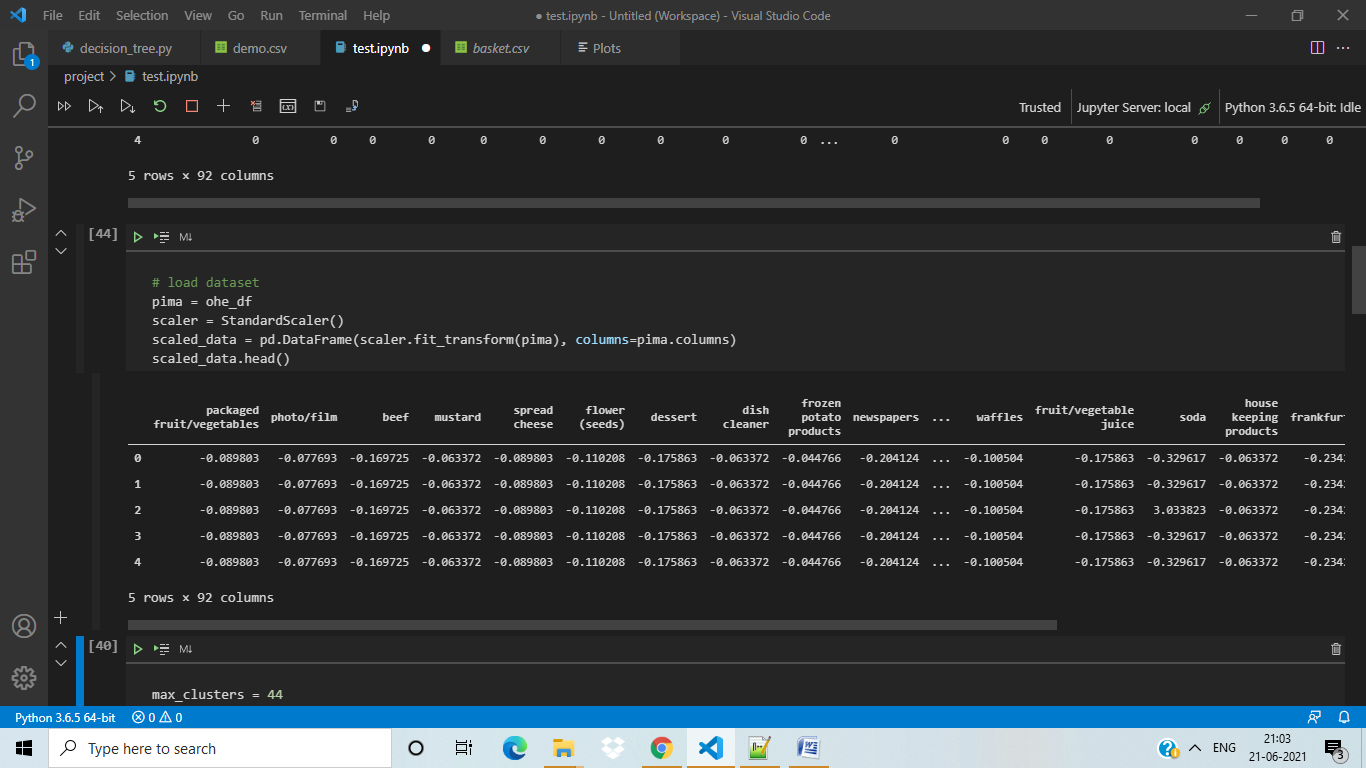
pima = ohe\_df

scaler = StandardScaler()

scaled\_data = pd.DataFrame(scaler.fit\_transform(pima), columns=pima.columns)

scaled\_data.head()

OUTPUT



---------------------------------------------------------------------------

#clustering

kmeans = KMeans(n\_clusters=7, n\_init=10)

kmeans.fit(scaled\_data)

clusters = kmeans.predict(scaled\_data)

pca = PCA(n\_components=8)

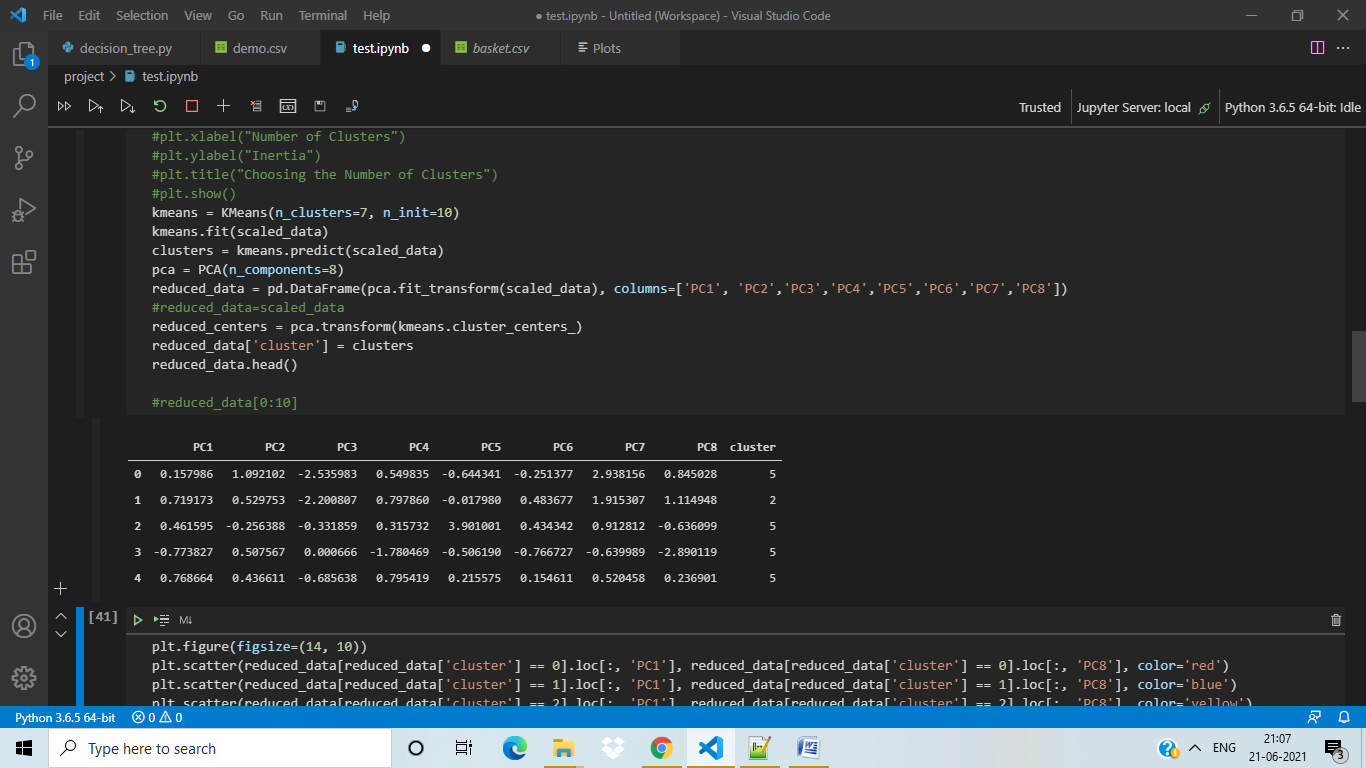
reduced\_data = pd.DataFrame(pca.fit\_transform(scaled\_data), columns=['PC1', 'PC2','PC3','PC4','PC5','PC6','PC7','PC8'])

reduced\_centers = pca.transform(kmeans.cluster\_centers\_)

reduced\_data['cluster'] = clusters

reduced\_data.head()

OUTPUT OF PCA ANALYSIS



plt.figure(figsize=(14, 10))

plt.scatter(reduced\_data[reduced\_data['cluster'] == 0].loc[:, 'PC1'], reduced\_data[reduced\_data['cluster'] == 0].loc[:, 'PC8'], color='red')

plt.scatter(reduced\_data[reduced\_data['cluster'] == 1].loc[:, 'PC1'], reduced\_data[reduced\_data['cluster'] == 1].loc[:, 'PC8'], color='blue')

plt.scatter(reduced\_data[reduced\_data['cluster'] == 2].loc[:, 'PC1'], reduced\_data[reduced\_data['cluster'] == 2].loc[:, 'PC8'], color='yellow')

plt.scatter(reduced\_data[reduced\_data['cluster'] == 3].loc[:, 'PC1'], reduced\_data[reduced\_data['cluster'] == 3].loc[:, 'PC8'], color='orange')

plt.scatter(reduced\_data[reduced\_data['cluster'] == 4].loc[:, 'PC1'], reduced\_data[reduced\_data['cluster'] == 4].loc[:, 'PC8'], color='cyan')

plt.scatter(reduced\_data[reduced\_data['cluster'] == 5].loc[:, 'PC1'], reduced\_data[reduced\_data['cluster'] == 5].loc[:, 'PC8'], color='magenta')

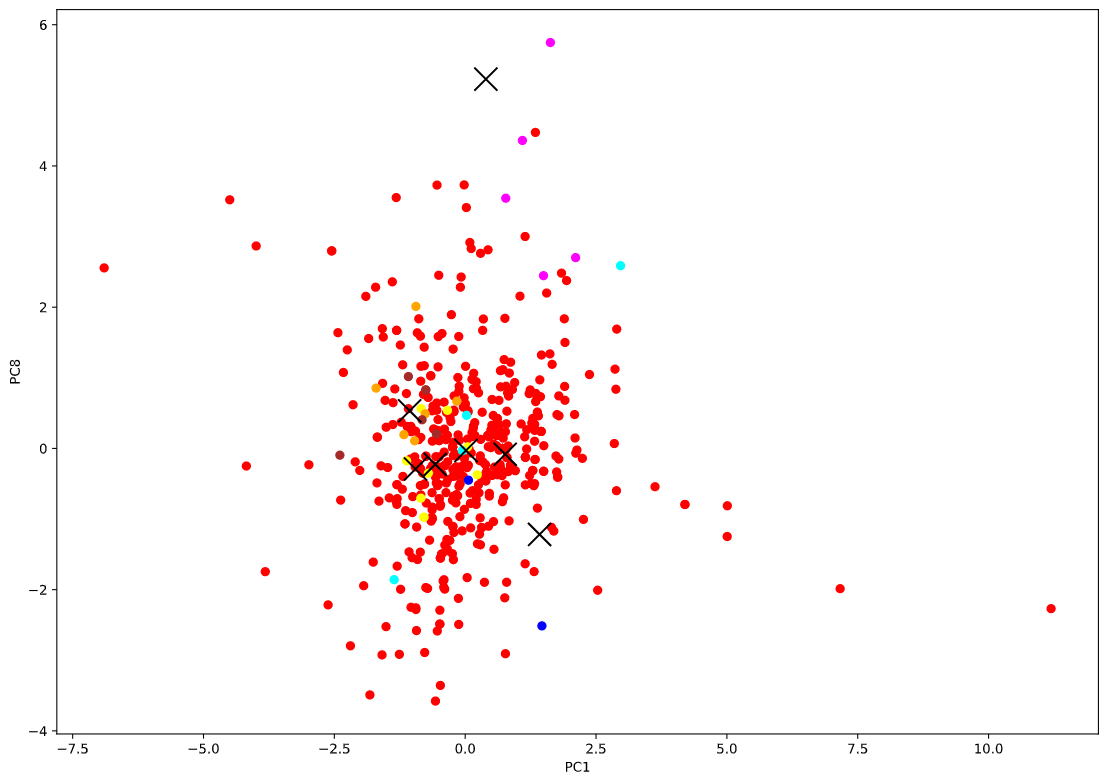
plt.scatter(reduced\_data[reduced\_data['cluster'] == 6].loc[:, 'PC1'], reduced\_data[reduced\_data['cluster'] == 6].loc[:, 'PC2'], color='brown')

plt.scatter(reduced\_centers[:, 0], reduced\_centers[:, 1], color='black', marker='x', s=300)

plt.xlabel("PC1")

plt.ylabel("PC8")

plt.show()



----------------------------------------------------

#Decision Tree

import sklearn

from sklearn import tree

X=reduced\_data.loc[:,reduced\_data.columns!='cluster']

y = reduced\_data.cluster

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=1)

clf = DecisionTreeClassifier()

clf = clf.fit(X\_train,y\_train)

y\_pred = clf.predict(X\_test)

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

metrics.confusion\_matrix(y\_test,y\_pred)

text\_representation = tree.export\_text(clf)

print(text\_representation)

#\_ = tree.plot\_tree(clf)

plt=tree.plot\_tree(clf)

plt

OUPUT

Accuracy: 0.9266666666666666

