Principal Component Analysis

# Checking data

import numpy as np  
import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
from sklearn.datasets import load\_breast\_cancer  
  
cancer = load\_breast\_cancer()  
print(type(cancer))  
print(cancer.keys())  
print(cancer["DESCR"])  
print()  
  
df = pd.DataFrame(cancer["data"],columns=cancer["feature\_names"])  
print(df.head())  
print()

# Performing standard scaling

from sklearn.preprocessing import StandardScaler  
scaler = StandardScaler()  
scaler.fit(df)  
scaled\_df = scaler.transform(df)

# Performing PCA to reduce dimensions

from sklearn.decomposition import PCA  
pca = PCA(n\_components=2)  
pca.fit(scaled\_df)  
x\_pca = pca.transform(scaled\_df)  
print(x\_pca.shape)

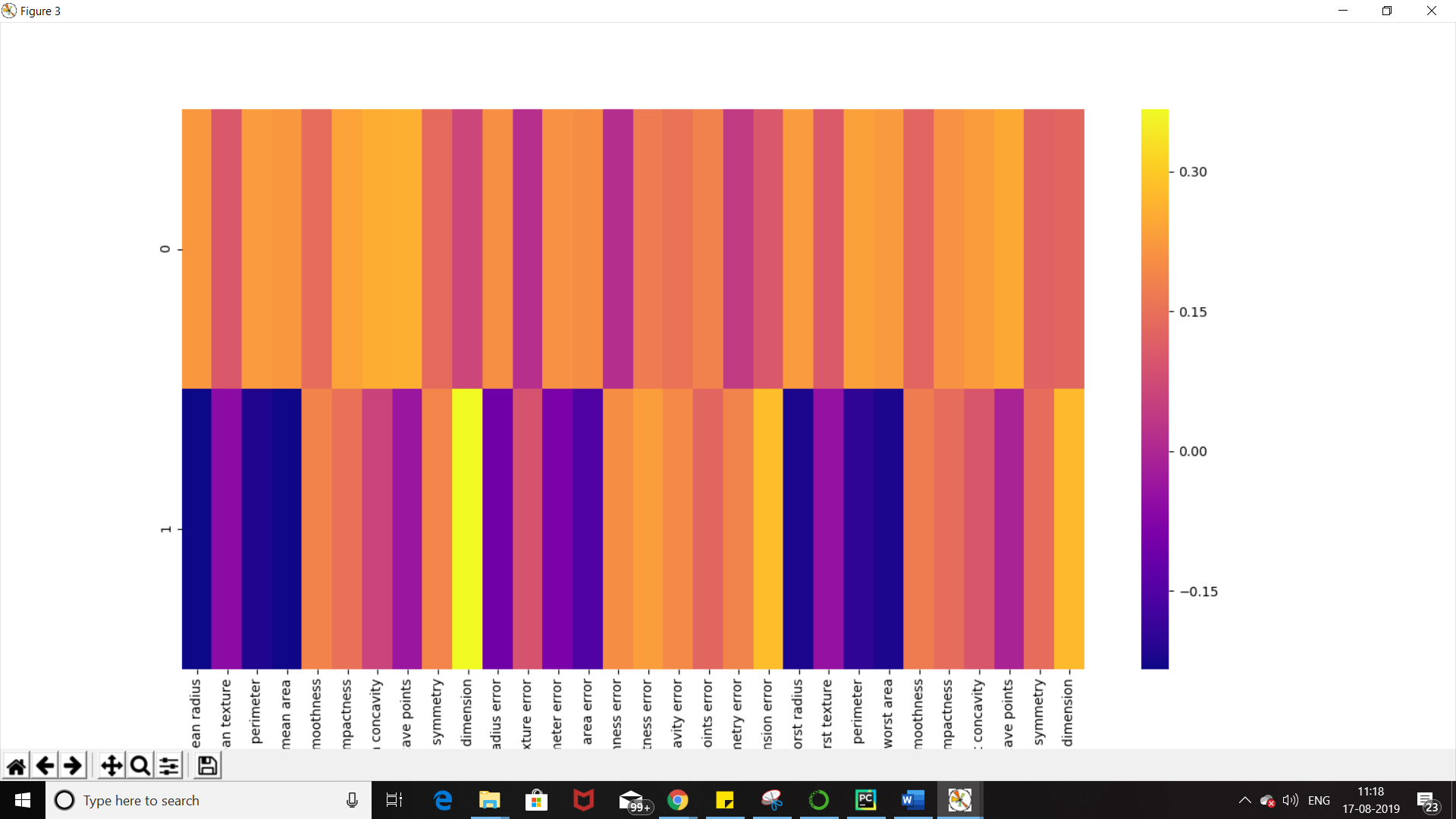
# Trying to plot data now with two components

plt.figure(num=1)  
plt.scatter(x\_pca[:,0],x\_pca[:,1])  
plt.xlabel("First Principal Component")  
plt.ylabel("Second Principal Component")  
  
plt.figure(num=2)  
plt.scatter(x\_pca[:,0],x\_pca[:,1],c=cancer["target"],cmap="plasma")  
plt.xlabel("First Principal Component")  
plt.ylabel("Second Principal Component")

# Checking co-relation between components and original features

print(pca.components\_)  
print()  
df\_comp = pd.DataFrame(pca.components\_,columns=cancer["feature\_names"])  
print(df\_comp.head())  
print()  
plt.figure(num=3,figsize=(12,6))  
sns.heatmap(df\_comp,cmap="plasma")  
  
plt.show()

We used heatmap to show co-relation between components which we got from PCA and our original features. Using colour scheme, we can find out the co-relation.



Once we have got out principal components, we can use these new components in our normal machine learning algorithm such as logistic regression or SVM.