**PRACTICAL 4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name:** | Harsh Shah | **Semester:** | VII | **Division:** | 6 |
| **Roll No.:** | 21BCP359 | **Date:** | 13-08-24 | **Batch:** | G11 |
| **Aim:** | Understanding Feature Extraction in Datasets. | | | | |

**Question 1**

**Dataset:** iris.csv

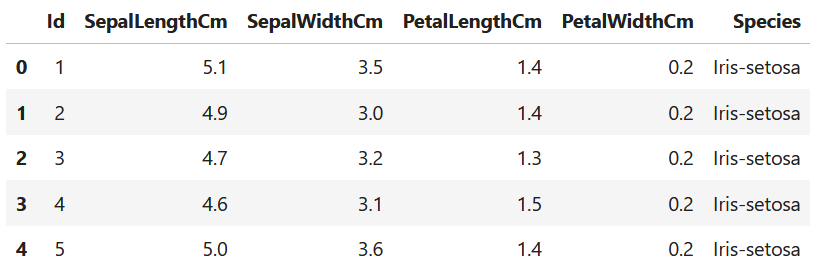
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

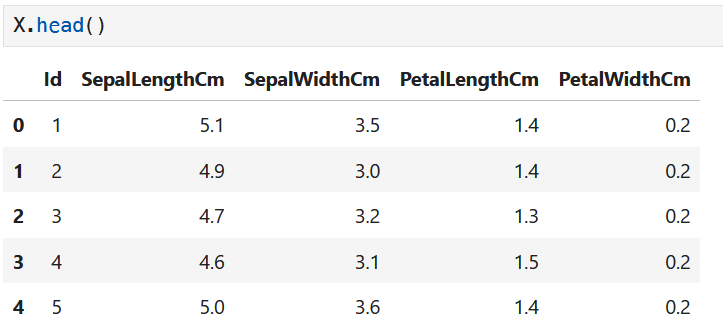
df = pd.read\_csv('./Iris.csv')

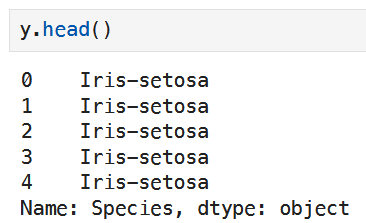


***#******Splitting Features and Target***

X = df.drop(['Species'], axis=1)

y = df['Species']



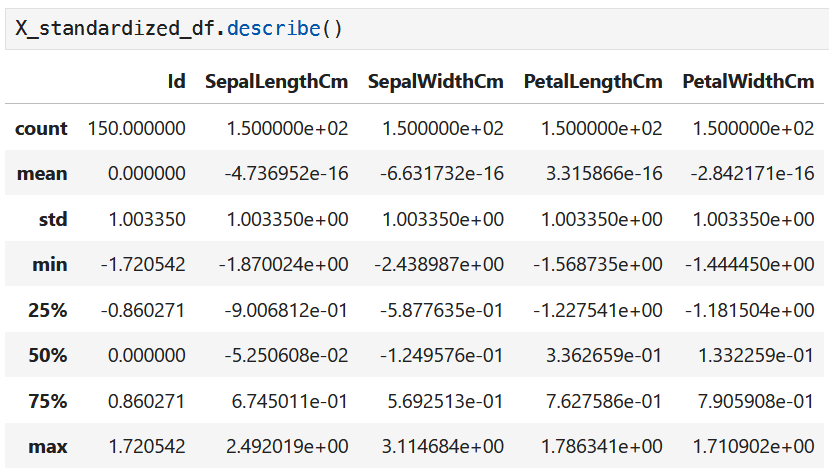
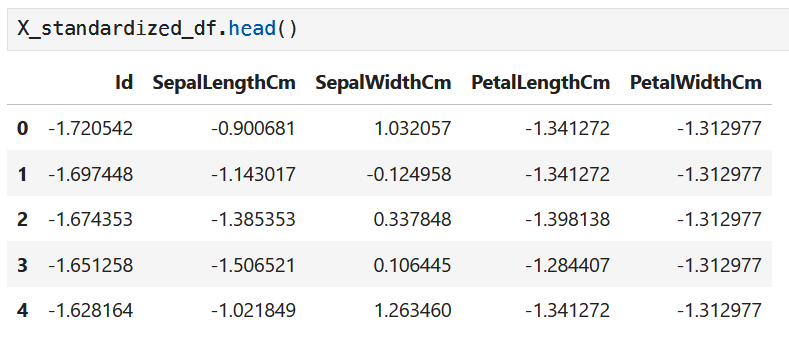


***# Standard Scaler***

scaler = StandardScaler()

X\_standardized = scaler.fit\_transform(X)

X\_standardized\_df = pd.DataFrame(X\_standardized, columns=X.columns)



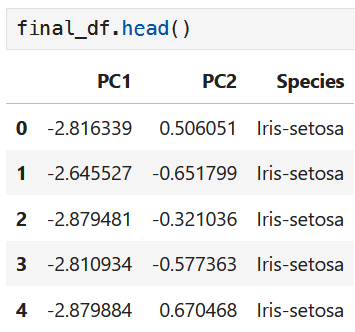
***# Principle Component Analysis***

pca = PCA(n\_components=2)

principal\_components = pca.fit\_transform(X\_standardized)

principal\_df = pd.DataFrame(principal\_components, columns=['PC1', 'PC2'])

final\_df = pd.concat([principal\_df, y], axis=1)



***# Plot***

plt.figure(figsize=(8, 6))

colors = ['red', 'green', 'blue']

species\_names = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']

for species, color in zip(species\_names, colors):

indices\_to\_keep = final\_df['Species'] == species

plt.scatter(final\_df.loc[indices\_to\_keep, 'PC1'],

final\_df.loc[indices\_to\_keep, 'PC2'],

c=color, s=50, label=species)

# Add labels and title

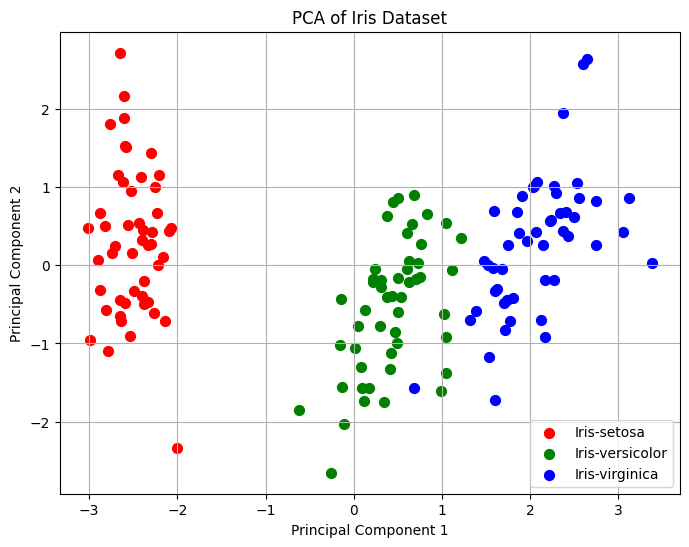
plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.title('PCA of Iris Dataset')

plt.legend()

plt.grid()



**Question 2**

**Dataset:** wine.csv

import pandas as pd

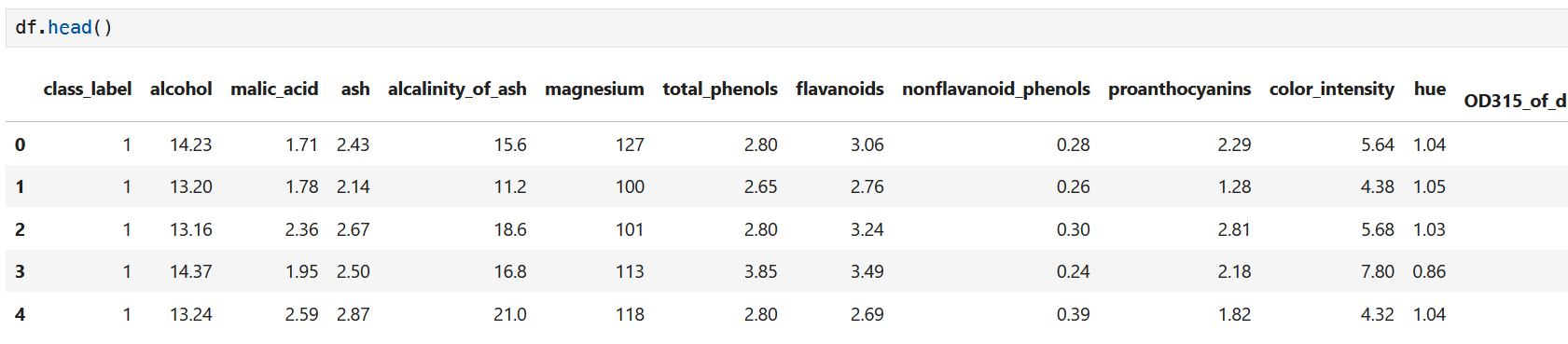
import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

df = pd.read\_csv('./wine\_data.csv')



X = df.drop(['class\_label'], axis=1)

y = df['class\_label']

***# Standardization***

scaler = StandardScaler()

X\_standardized = scaler.fit\_transform(X)

X\_standardized\_df = pd.DataFrame(X\_standardized, columns=X.columns)

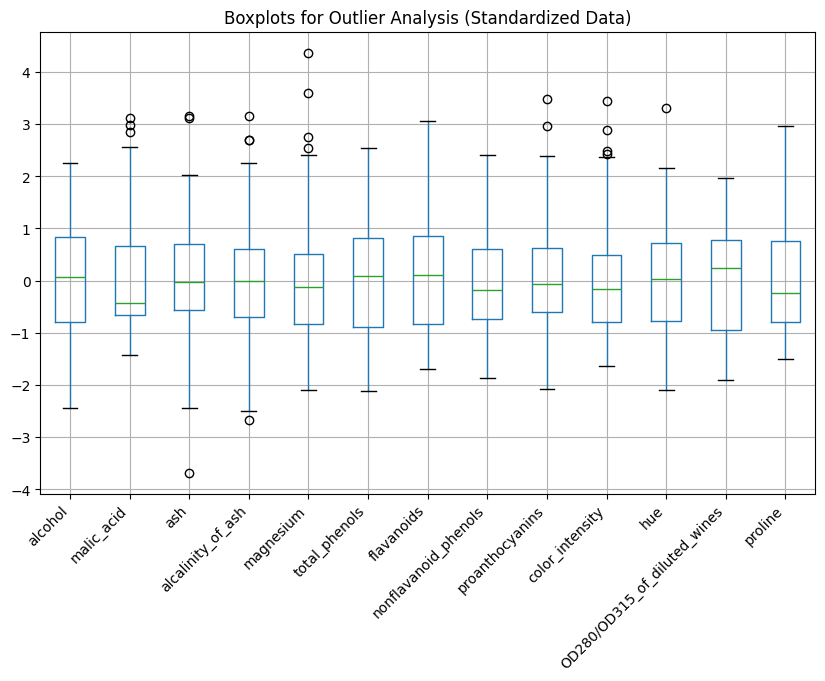
plt.figure(figsize=(10, 6))

X\_standardized\_df.boxplot()

plt.xticks(rotation=45, ha='right')

plt.title('Boxplots for Outlier Analysis (Standardized Data)')

plt.show()



***# Covariance Matrix***

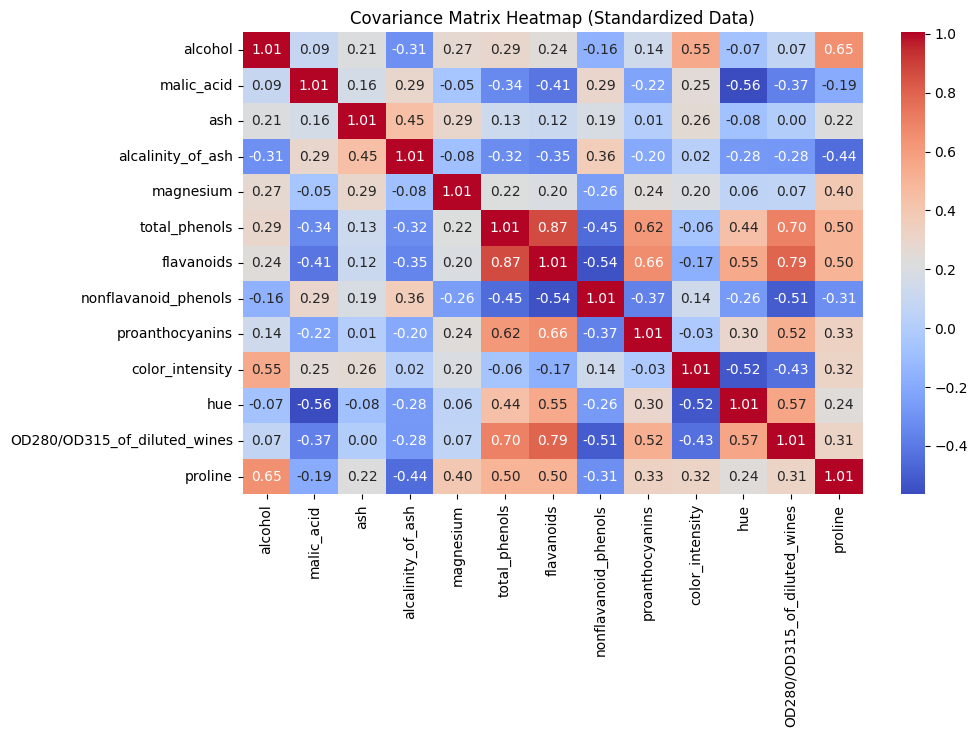
cov\_matrix\_standardized = pd.DataFrame(X\_standardized, columns=X.columns).cov()

plt.figure(figsize=(10, 6))

sns.heatmap(cov\_matrix\_standardized, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Covariance Matrix Heatmap (Standardized Data)')

plt.show()



***# PCA without specifying components***

pca = PCA(n\_components=None)

pca.fit(X\_standardized)

plt.figure(figsize=(8, 5))

plt.scatter(range(1, len(pca.explained\_variance\_ratio\_) + 1), pca.explained\_variance\_ratio\_, label='Variance Ratio', color='blue', alpha=0.6)

# plt.plot(range(1, len(pca.explained\_variance\_ratio\_) + 1), pca.explained\_variance\_ratio\_)

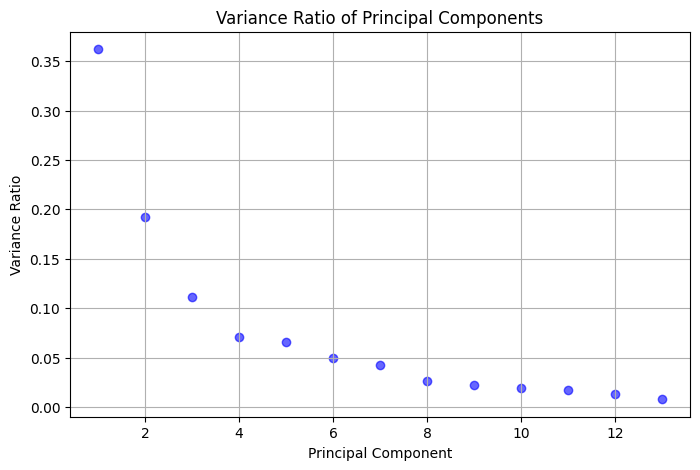
plt.xlabel('Principal Component')

plt.ylabel('Variance Ratio')

plt.title('Variance Ratio of Principal Components')

plt.grid()

plt.show()



***# PCA with 2 components***

pca\_2d = PCA(n\_components=2)

principal\_components = pca\_2d.fit\_transform(X\_standardized)

principal\_df = pd.DataFrame(data=principal\_components, columns=['PC1', 'PC2'])

final\_df = pd.concat([principal\_df, y.reset\_index(drop=True)], axis=1)

plt.figure(figsize=(10, 6))

colors = ['red', 'green', 'blue']

for label, color in zip(df['class\_label'].unique(), colors):

indices\_to\_keep = final\_df['class\_label'] == label

plt.scatter(final\_df.loc[indices\_to\_keep, 'PC1'],

final\_df.loc[indices\_to\_keep, 'PC2'],

c=color, s=50, label=label, alpha=0.6)

plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.title('PCA of Wine Dataset (2 Components)')

plt.legend()

plt.grid()

plt.show()

