#Load Libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

from sklearn.cluster import KMeans

from sklearn.metrics import silhouette\_score

#Load Dataset

dataset = pd.read\_csv('./illnessstudy.csv')

dataset.head()

#Create x (we ignore the y variable)

x = dataset.drop('diagnosis', axis=1).to\_numpy()

#Scale the Data

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

data\_transformed=sc.fit\_transform(x)

#Determine optimum number of clusters

wcss = []

for i in range(1, 14):

kmeans = KMeans(n\_clusters = i, init = 'k-means++', max\_iter = 300, n\_init = 10,random\_state = 100)

kmeans.fit(data\_transformed)

wcss.append(kmeans.inertia\_)

#Plot Elbow Method

plt.plot(range(1, 14), wcss,marker='o')

plt.title('The elbow method')

plt.xlabel('Number of clusters')

plt.ylabel('WCSS') #within cluster sum of squares

plt.show()

#Create Silhouette Coefficients

scores= [0]

for n\_cluster in range(2, 12):

kmeans = KMeans(n\_clusters=n\_cluster).fit(x)

label = kmeans.labels\_

sil\_coeff = silhouette\_score(data\_transformed, label, metric='euclidean')

scores.append(sil\_coeff)

print('For n\_clusters= {}, The Silhouette Coefficient is {}'.format(n\_cluster, sil\_coeff))

#Plot Silhouette

plt.plot(range(1,12), scores, marker='o')

plt.title('The Silhouette method')

plt.xlabel('Number of clusters')

plt.ylabel('Sihouette Coefficient')

plt.show()