APPLICATION OF CLUSTERING MODELS

Dataset Link https://drive.google.com/file/d/1pP0Rr83ri0voscgr95-YnVCBv6BYV22w/view Hint: Problem 1: There are various stocks for which we have collected a data set, which all stocks are apparently similar in performance

Problem 2: How many Unique patterns that exist in the historical stock data set, based on fluctuations in price.

Problem 3: Identify which all stocks are moving together and which all stocks are different from each other.

NOTE: The solution shared through Github should contain the source code used and the screenshot of the output.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

from sklearn.cluster import KMeans

from sklearn import metrics

from scipy.spatial.distance import cdist

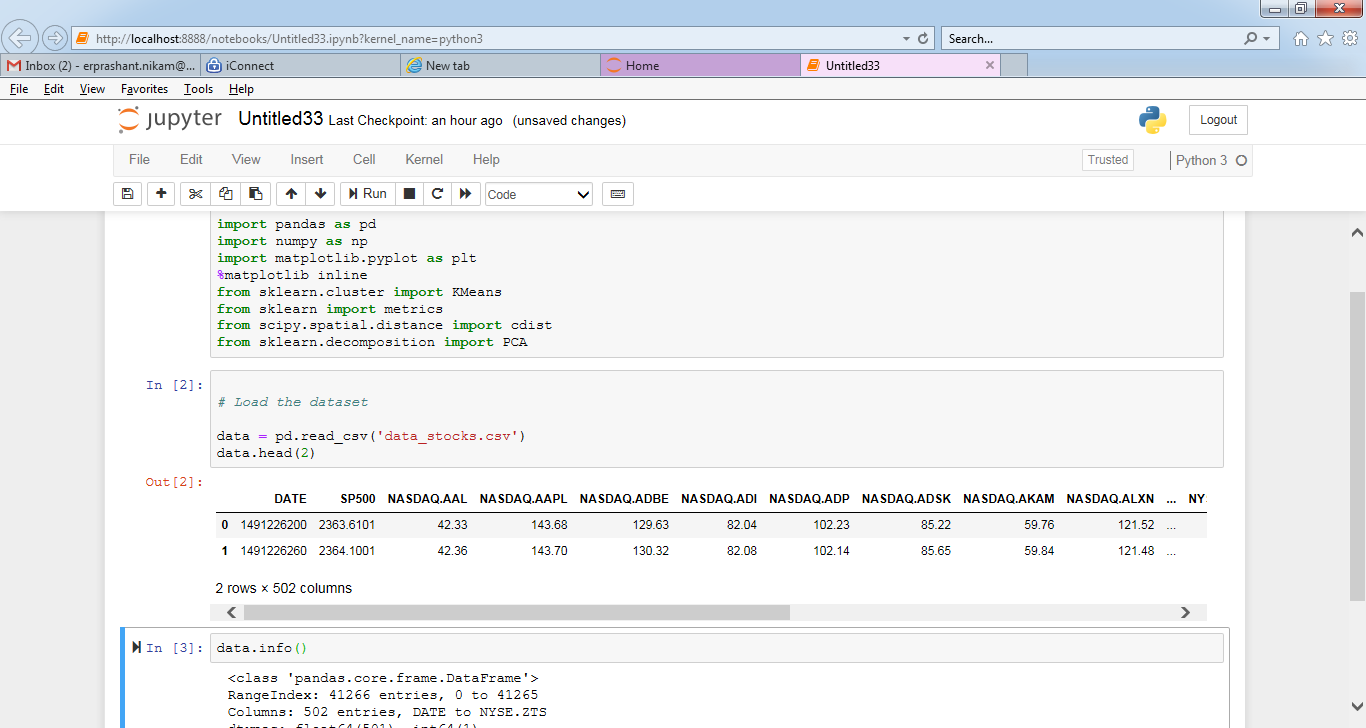
from sklearn.decomposition import PCA

# Load the dataset

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

data.head(2)

data.info()



data.info()

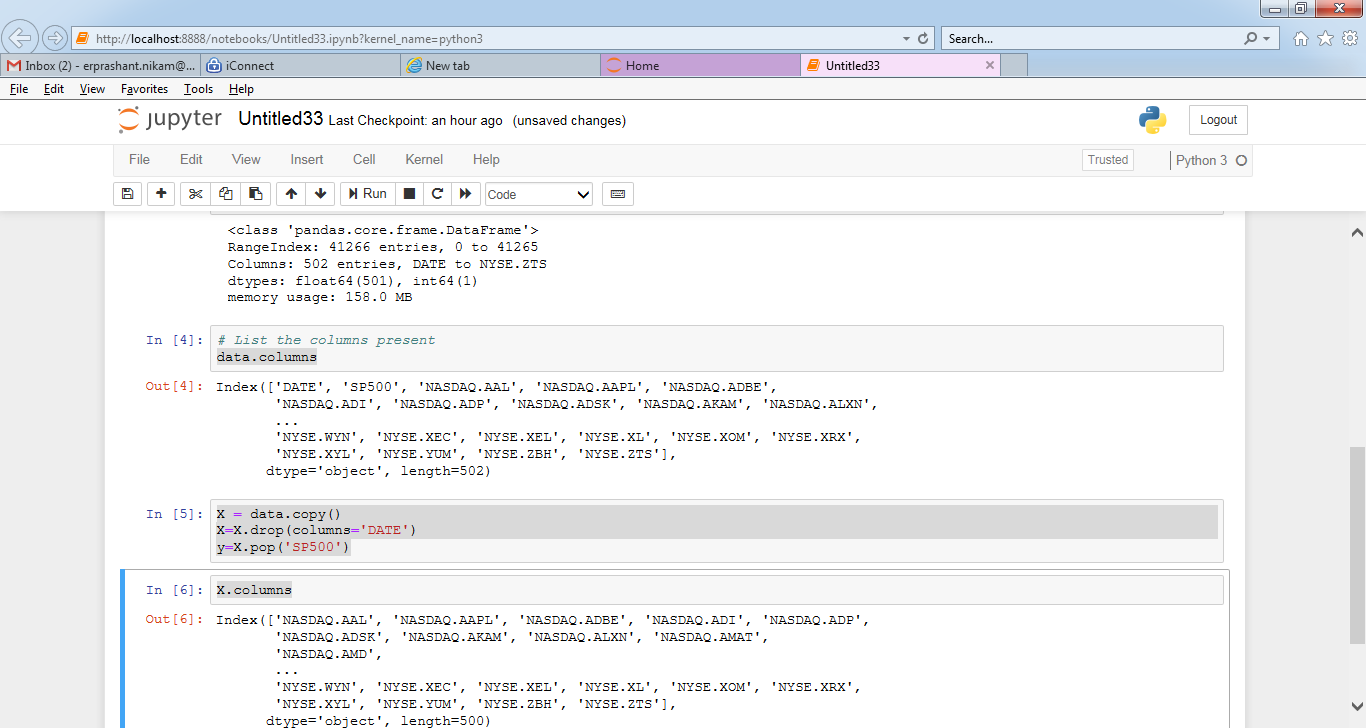
data.columns

X = data.copy()

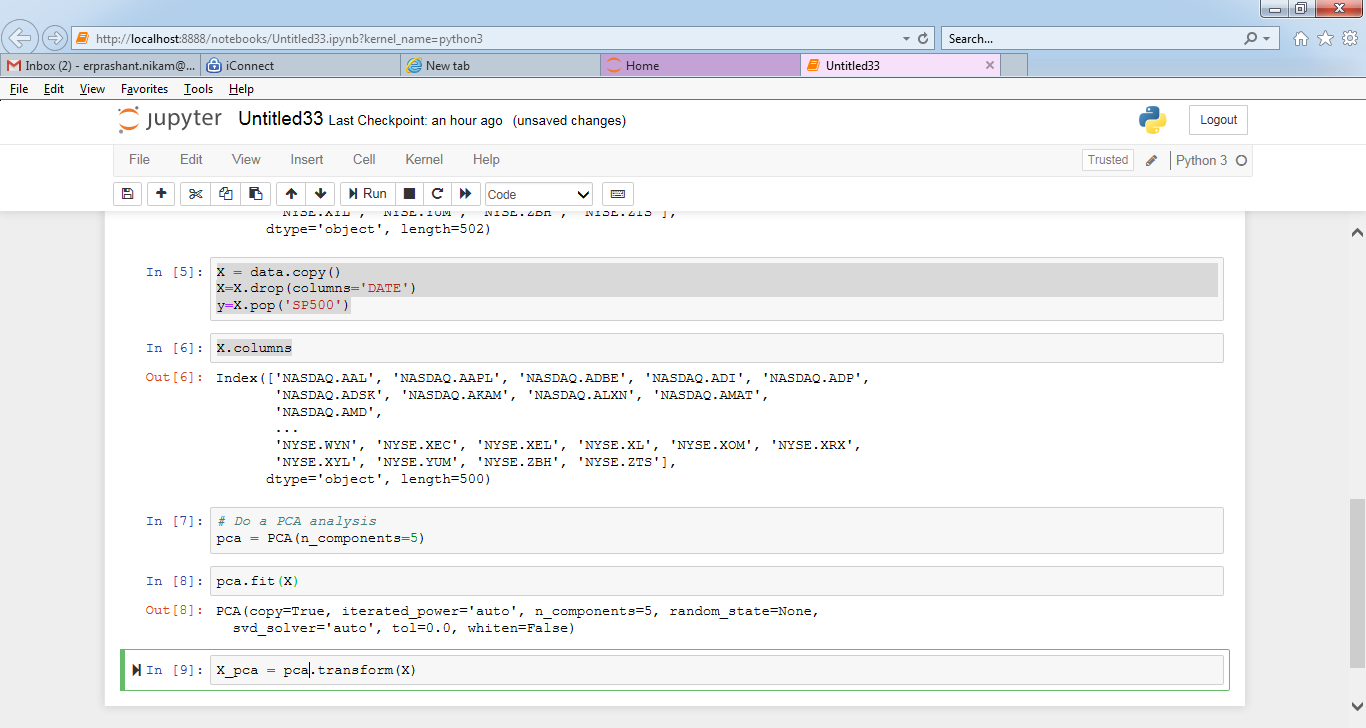
X=X.drop(columns='DATE')

y=X.pop('SP500')

X.columns



pca = PCA(n\_components=5)



pca.fit(X)

X\_pca = pca.transform(X)

*# Plot the PCA components*

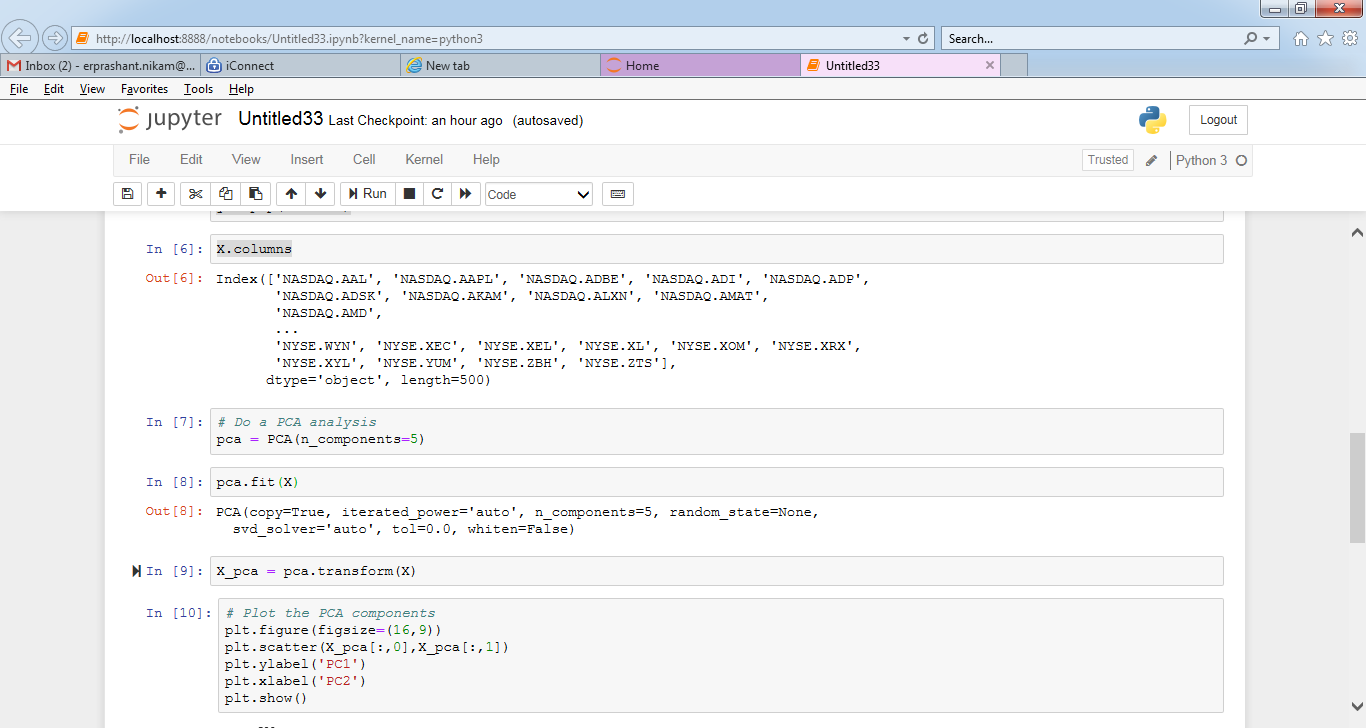
plt.figure(figsize=(16,9))

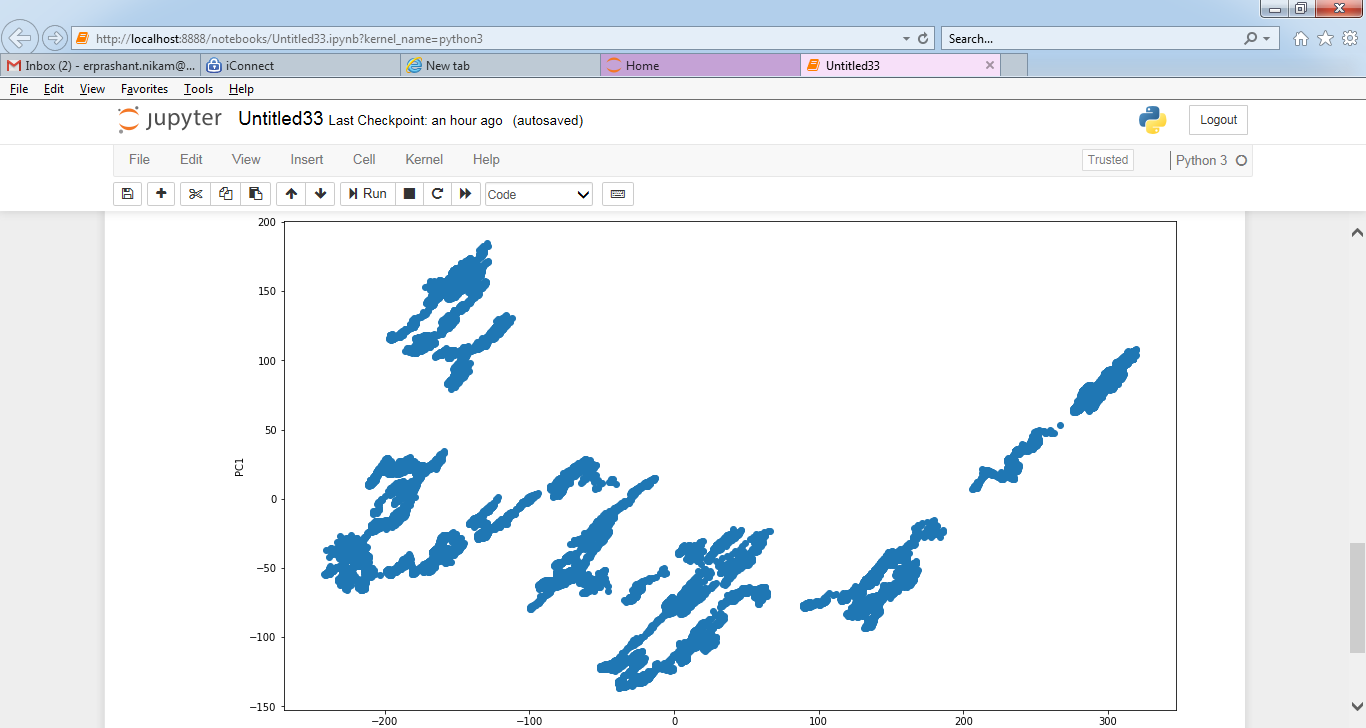
plt.scatter(X\_pca[:,0],X\_pca[:,1])

plt.ylabel('PC1')

plt.xlabel('PC2')

plt.show()





pca.explained\_variance\_

*# Find the number of clusters*

distortions = []

K = range(1,20)

**for** k **in** K:

kmeanModel = KMeans(n\_clusters=k).fit(X)

kmeanModel.fit(X)

distortions.append(sum(np.min(cdist(X, kmeanModel.cluster\_centers\_, 'euclidean'), axis=1)) / X.shape[0])

*# Plot the elbow*

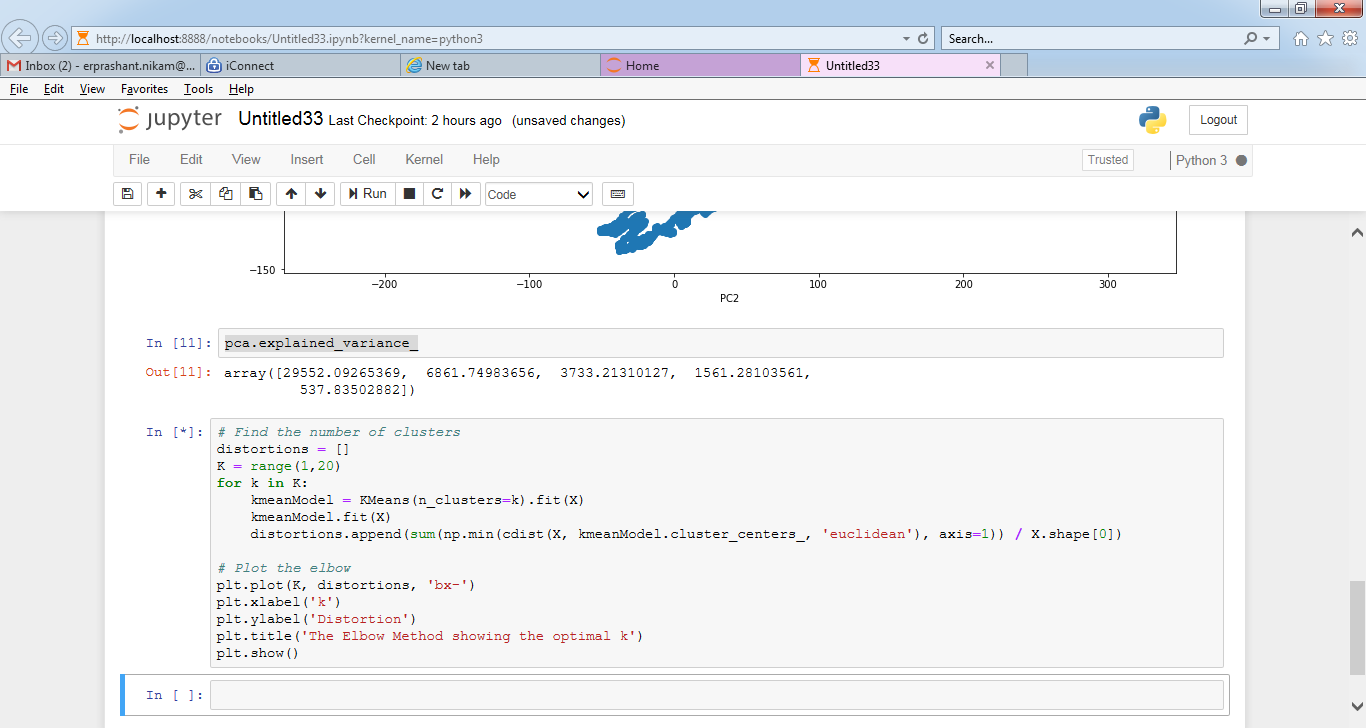
plt.plot(K, distortions, 'bx-')

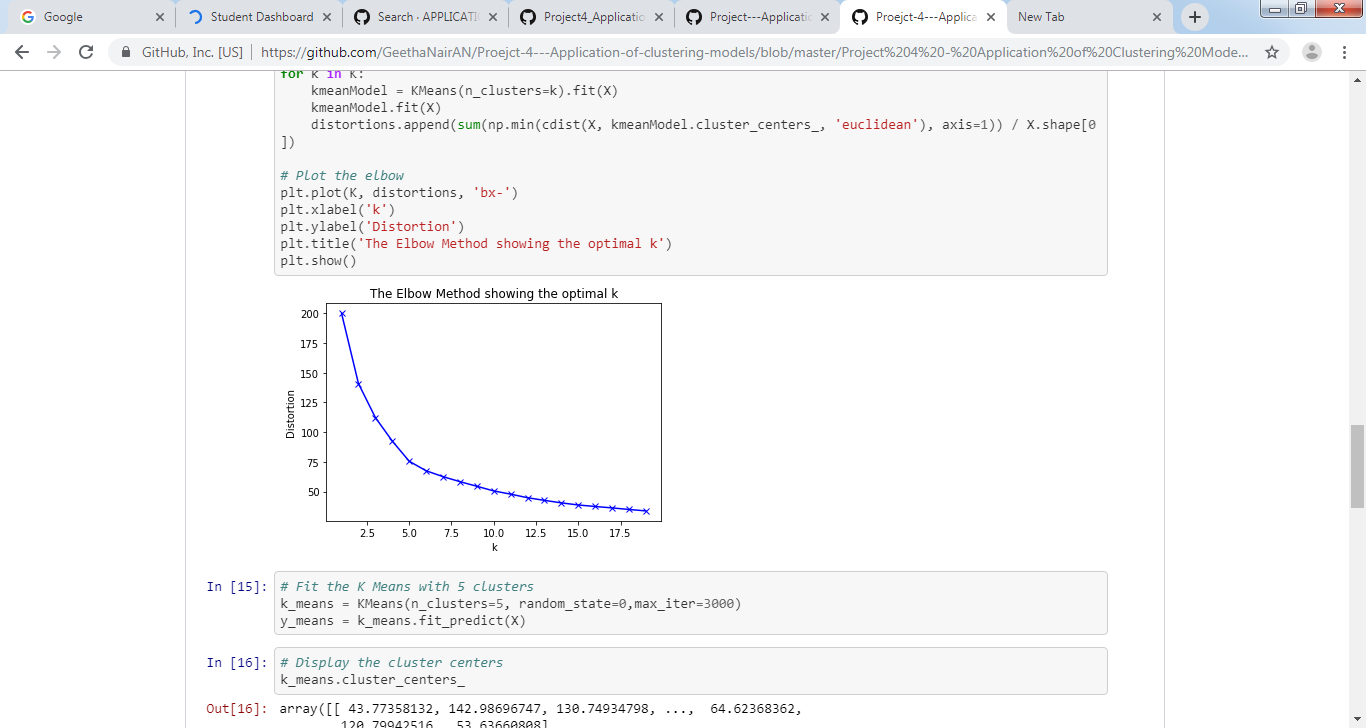
plt.xlabel('k')

plt.ylabel('Distortion')

plt.title('The Elbow Method showing the optimal k')

plt.show()



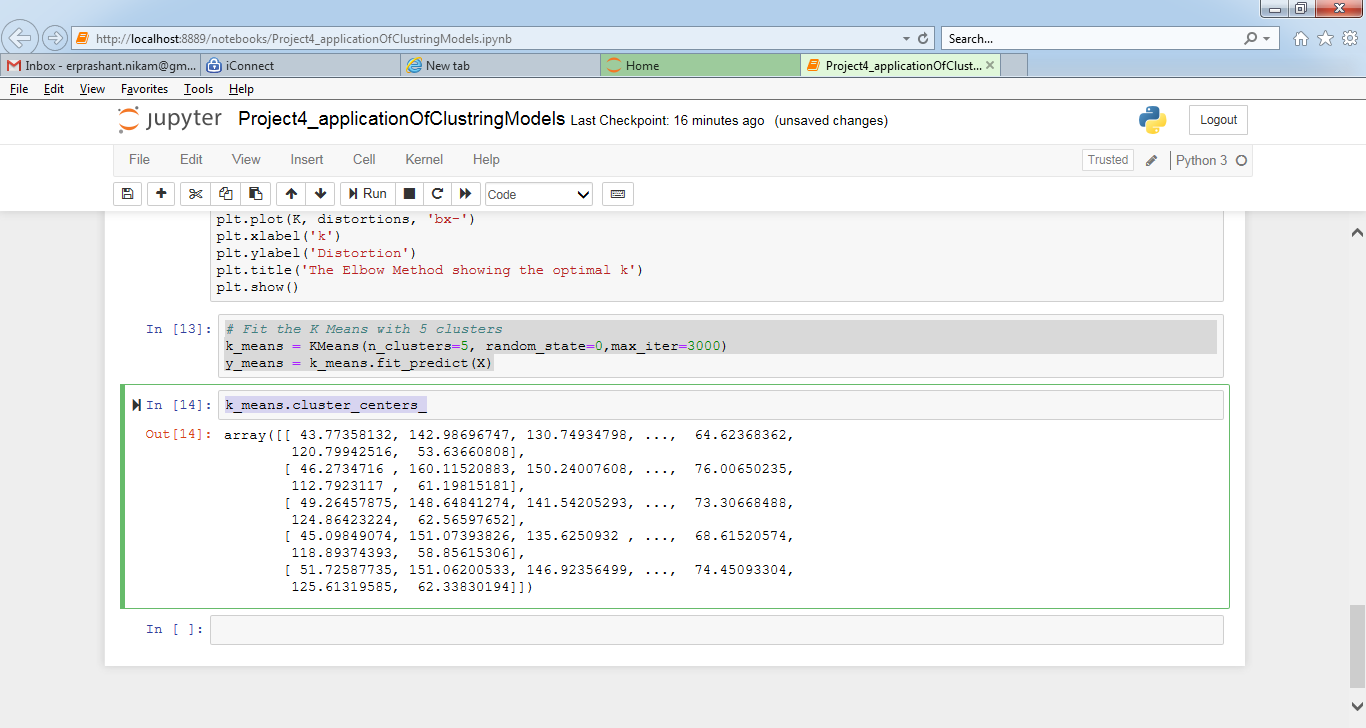


# Fit the K Means with 5 clusters

k\_means = KMeans(n\_clusters=5, random\_state=0,max\_iter=3000)

y\_means = k\_means.fit\_predict(X)

k\_means.cluster\_centers\_



k\_means.labels\_.ravel()

k\_means\_cluster\_centers = np.sort(k\_means.cluster\_centers\_, axis=0)

k\_means\_labels = metrics.pairwise\_distances\_argmin(X, k\_means\_cluster\_centers)

# Plot the cluster values.

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

plt.scatter(X.iloc[y\_means == 0,0], X.iloc[y\_means == 0,499], s= 100, color = 'red')

plt.scatter(X.iloc[y\_means == 1,0], X.iloc[y\_means == 1,499], s= 100, color = 'blue')

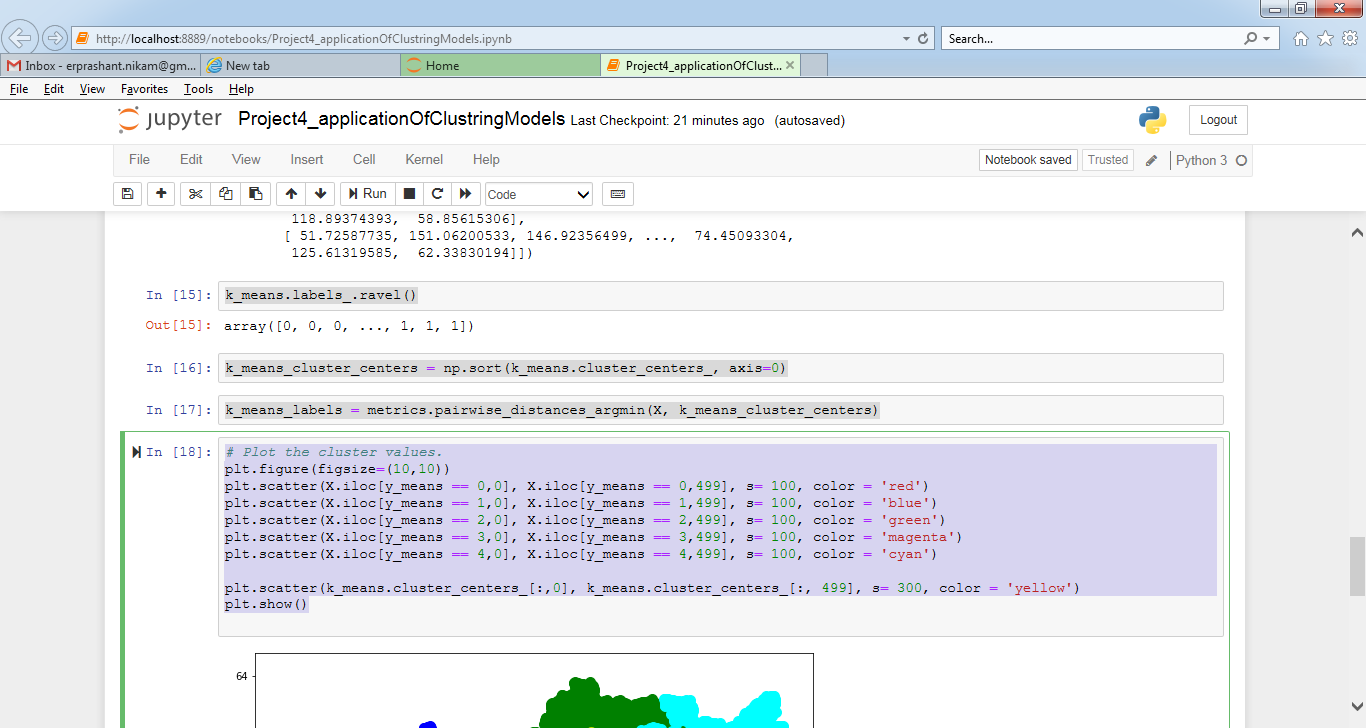
plt.scatter(X.iloc[y\_means == 2,0], X.iloc[y\_means == 2,499], s= 100, color = 'green')

plt.scatter(X.iloc[y\_means == 3,0], X.iloc[y\_means == 3,499], s= 100, color = 'magenta')

plt.scatter(X.iloc[y\_means == 4,0], X.iloc[y\_means == 4,499], s= 100, color = 'cyan')

plt.scatter(k\_means.cluster\_centers\_[:,0], k\_means.cluster\_centers\_[:, 499], s= 300, color = 'yellow')

plt.show()



# Plot the cluster values.

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

plt.scatter(X.iloc[y\_means == 0,0], X.iloc[y\_means == 0,499], s= 100, color = 'red')

plt.scatter(X.iloc[y\_means == 1,0], X.iloc[y\_means == 1,499], s= 100, color = 'blue')

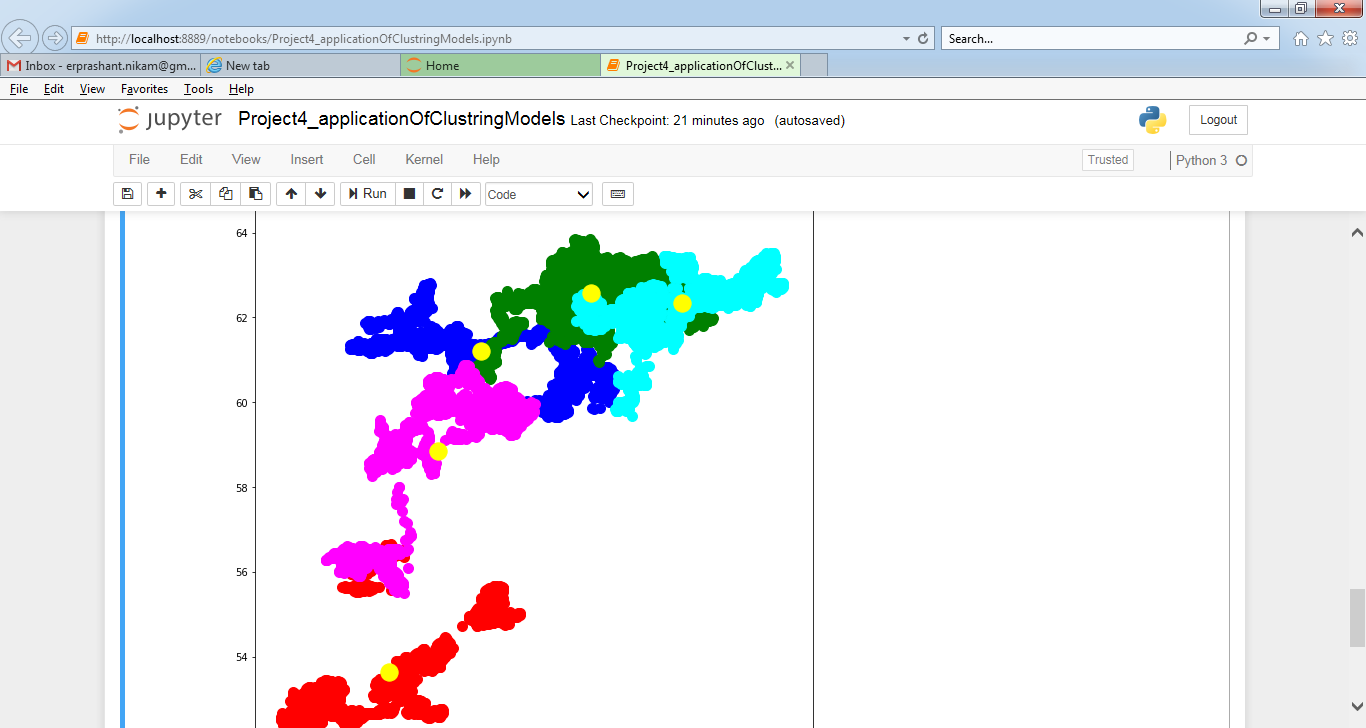
plt.scatter(X.iloc[y\_means == 2,0], X.iloc[y\_means == 2,499], s= 100, color = 'green')

plt.scatter(X.iloc[y\_means == 3,0], X.iloc[y\_means == 3,499], s= 100, color = 'magenta')

plt.scatter(X.iloc[y\_means == 4,0], X.iloc[y\_means == 4,499], s= 100, color = 'cyan')

plt.scatter(k\_means.cluster\_centers\_[:,0], k\_means.cluster\_centers\_[:, 499], s= 300, color = 'yellow')

plt.show()



# How many Unique patterns that exist in the historical stock data set, based on fluctuations in price.

print ('Number of patterns in the first cluster : ', X.iloc[y\_means == 0,0].size)

print ('Number of patterns in the second cluster : ', X.iloc[y\_means == 1,0].size)

print ('Number of patterns in the third cluster : ', X.iloc[y\_means == 2,0].size)

print ('Number of patterns in the fourth cluster : ', X.iloc[y\_means == 3,0].size)

print ('Number of patterns in the fifth cluster : ', X.iloc[y\_means == 4,0].size)

