Machine Learning: Assignment 6

Student ID: 700740357

Student name: Sarathchandra Nekkalapu

Github repository Link:
https://github.com/sarathchandra-99/ML_ASSIGNMENT6_700740357

1) (Provide only mathematical solutions for this question) Six points with the following attributes are given, calculate and find out clustering representations and dendrogram using Single, complete, and average link proximity function in hierarchical clustering technique.

Single Link Proximity:

- In **Single Linkage,** the distance between two clusters is the minimum distance between members of the two clusters

	p1	p2	p3	p4	p5	p6
p1	0	0.2357	0.2218	0.3688	0.3421	0.2347
p2	0.2357	0	0.1483	0.2042	0.1388	0.254
p3	0.2218	0.1483	0	0.1513	0.2843	0.11
p4	0.3688	0.2042	0.1513	0	0.2932	0.2216
p5	0.3421	0.1388	0.2843	0.2932	0	0.3921
p6	0.2347	0.254	0.11	0.2216	0.3921	0

smallest distance from above data is 0.11

so p3 and p6 forms first cluster

	p1	p2	p36	p4	p5
p1	0	0.2357	0.2218	0.3688	0.3421
p2	0.2357	0	0.1483	0.2042	0.1388
p3 6	0.2218	0.1483	0	0.1513	0.2843
p4	0.3688	0.2042	0.1513	0	0.2932
p5	0.3421	0.1388	0.2843	0.2932	0

smallest distance from above data is 0.1388

so p2 and p5 forms 2nd cluster

	p1	p25	p36	p4
p1	0	0.2357	0.2218	0.3688
p25	0.2357	0	0.1483	0.2042
p36	0.2218	0.1483	0	0.1513
p4	0.3688	0.2042	0.1513	0

smallest distance from above data is 0.1483

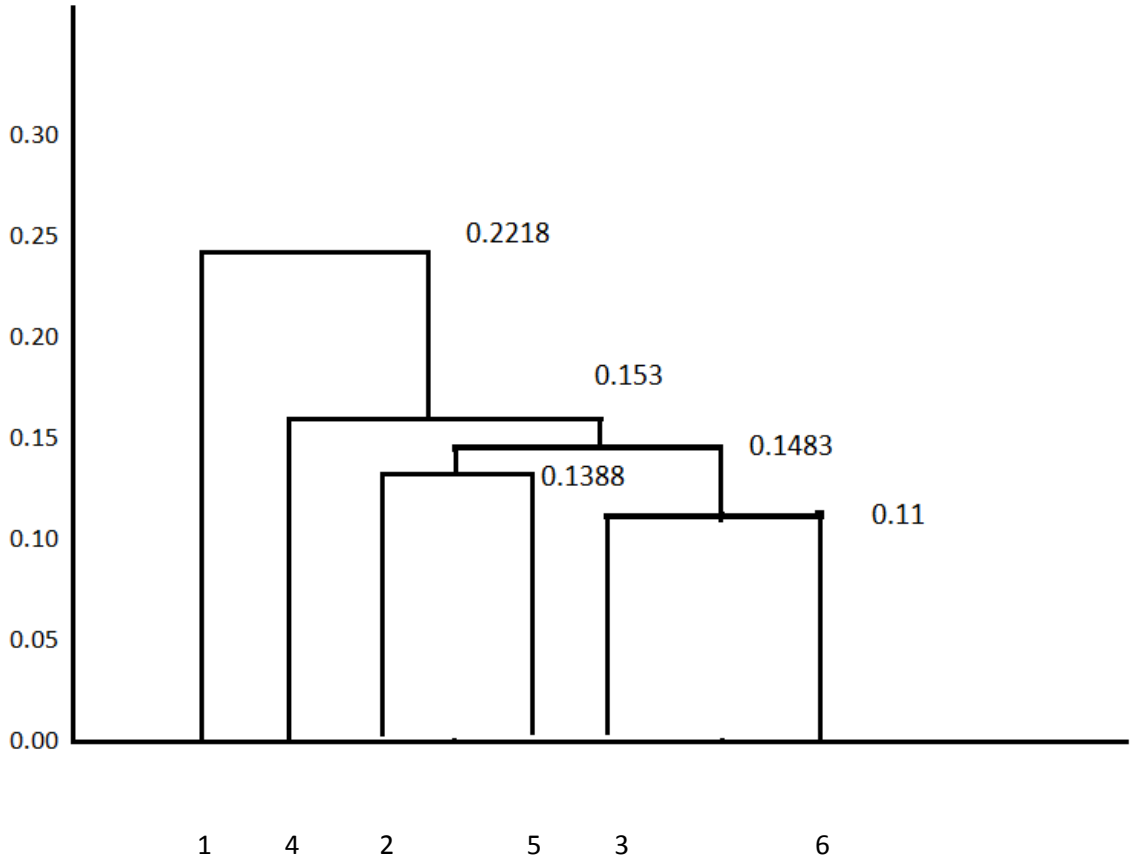
so p25 and p36 forms 3rd cluster

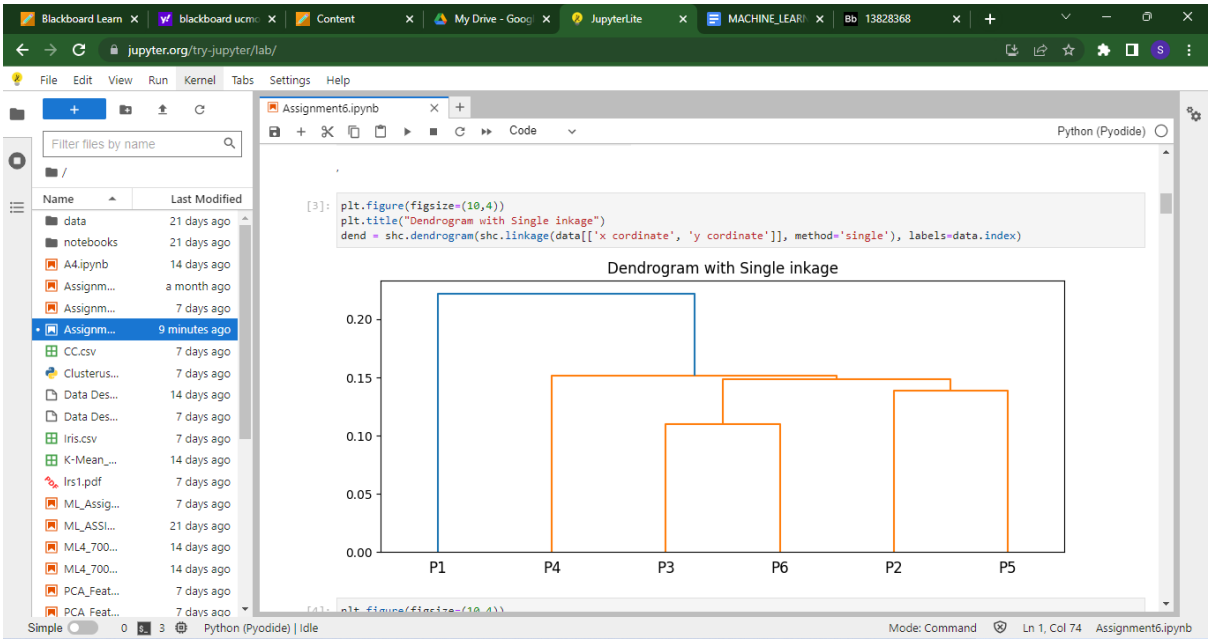
	p1	p(25)(36)	p4
p1	0	0.2218	0.3688
p(25)(36)	0.2218	0	0.1513
p4	0.3688	0.1513	0

smallest distance from above data is 0.153

so p(25)(36)and p4 forms 4thcluster

	p1	p4(25)(36)
p1	0	0.2218
p4(25)(36)	0.2218	0





Complete Link Proximity:

- In **Complete Linkage,** the distance between two clusters is the maximum distance between members of the two clusters

	p1	p2	p3	p4	p5	p6
p1	0	0.2357	0.2218	0.3688	0.3421	0.2347
p2	0.2357	0	0.1483	0.2042	0.1388	0.254
p3	0.2218	0.1483	0	0.1513	0.2843	0.11
p4	0.3688	0.2042	0.1513	0	0.2932	0.2216
p5	0.3421	0.1388	0.2843	0.2932	0	0.3921
p6	0.2347	0.254	0.11	0.2216	0.3921	0

smallest distance from above data is 0.11

so p3 and p6 forms first cluster

	p1	p2	p36	p4	p5
p1	0	0.2357	0.2347	0.3688	0.3421
p2	0.2357	0	0.254	0.2042	0.1388
p36	0.2347	0.254	0	0.2216	0.3921
p4	0.3688	0.2042	0.2216	0	0.2932
p5	0.3421	0.1388	0.3921	0.2932	0

smallest distance from above data is 0.1388

so p2 and p5 forms 2nd cluster

	p1	p25	p36	p4
p1	0	0.3421	0.2347	0.3688
p25	0.3421	0	0.3921	0.2932
p36	0.2347	0.3921	0	0.2216
p4	0.3688	0.2932	0.2216	0

smallest distance from above data is 0.2216

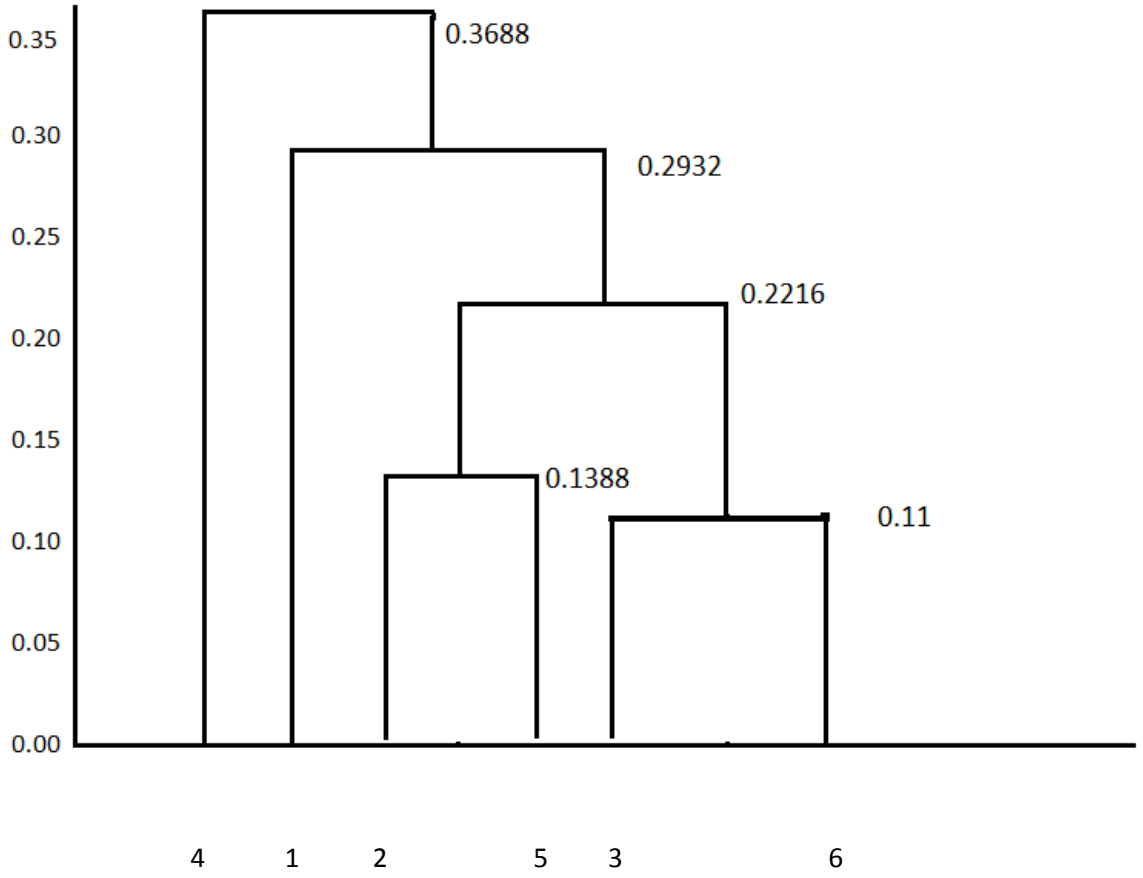
so p25 and p36 forms 3rdcluster

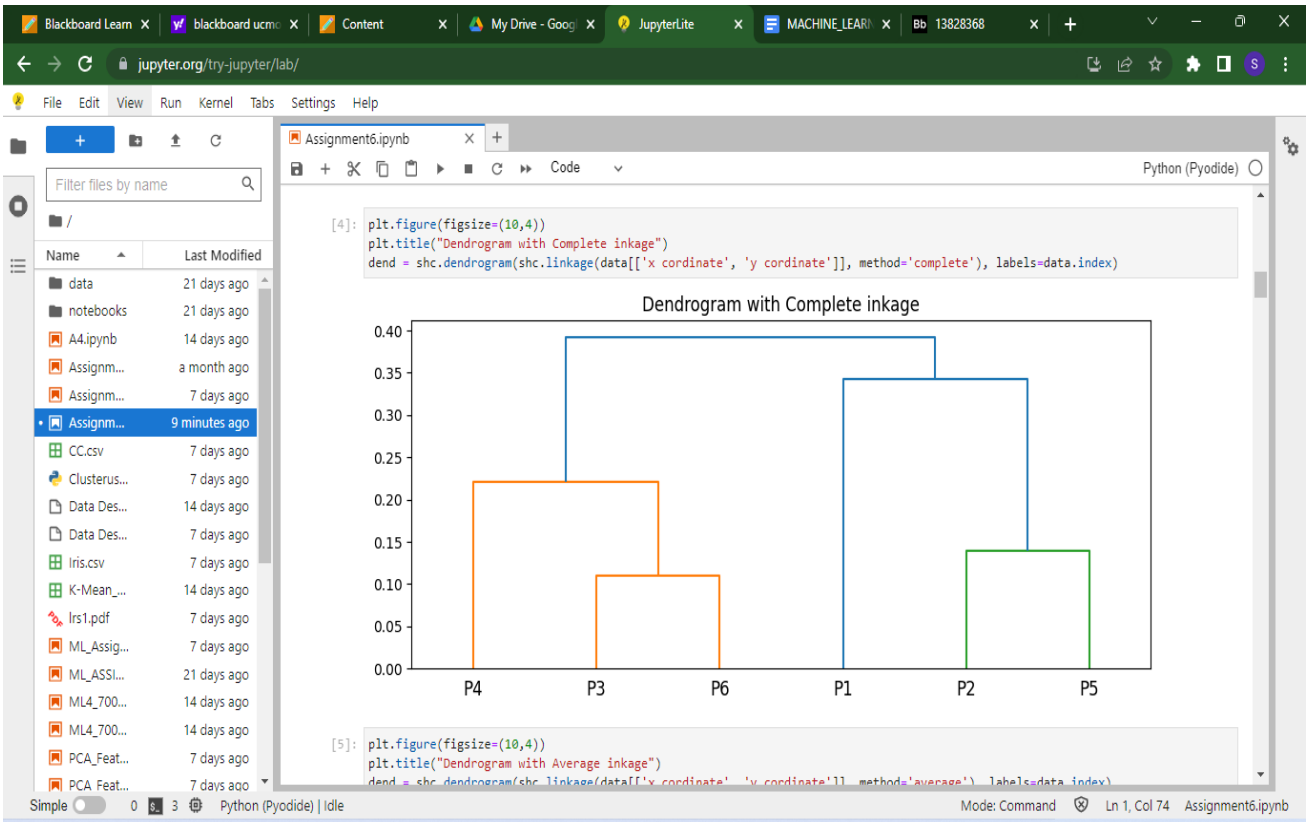
	p1	p(25)(36)	p4
p1	0	0.3421	0.3688
p(25)(36)	0.3421	0	0.2932
p4	0.3688	0.2932	0

smallest distance from above data is 0.2932

so p(25)(36)and p1 forms 4thcluster

	p1(25)(36)	p4
p1(25)(36)	0	0.1483
p4	0.3688	0


0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.3688
0.2932
0.2216
0.1388
0.11
4
1
2
5
3
6



Average Link Proximity:

In **Average Linkage,** the distance between two clusters is the average of all distances between members of the two clusters

	p1	p2	p3	p4	p5	p6
p1	0	0.2357	0.2218	0.3688	0.3421	0.2347
p2	0.2357	0	0.1483	0.2042	0.1388	0.254
p3	0.2218	0.1483	0	0.1513	0.2843	0.11
p4	0.3688	0.2042	0.1513	0	0.2932	0.2216
p5	0.3421	0.1388	0.2843	0.2932	0	0.3921
p6	0.2347	0.254	0.11	0.2216	0.3921	0

smallest distance from above data is 0.11

so p3 and p6 forms first cluster

	p1	p2	p36	p4	p5
p1	0	0.2357	0.22825	0.3688	0.3421
p2	0.2357	0	0.20115	0.2042	0.1388
p36	0.22825	0.20115	0	0.1864 5	0.3382
p4	0.3688	0.2042	0.18645	0	0.2932
p5	0.3421	0.1388	0.3382	0.2932	0

smallest distance from above data is 0.1388

so p2 and p5 forms 2nd cluster

	p1	p25	p36	p4
p1	0	0.2889	0.2347	0.3688
p25	0.2889	0	0.26967 5	0.2487
p36	0.2347	0.26967 5	0	0.1864 5
p4	0.3688	0.2487	0.18645	0

smallest distance from above data is 0.18645

so p25 and p36 forms 3rdcluster

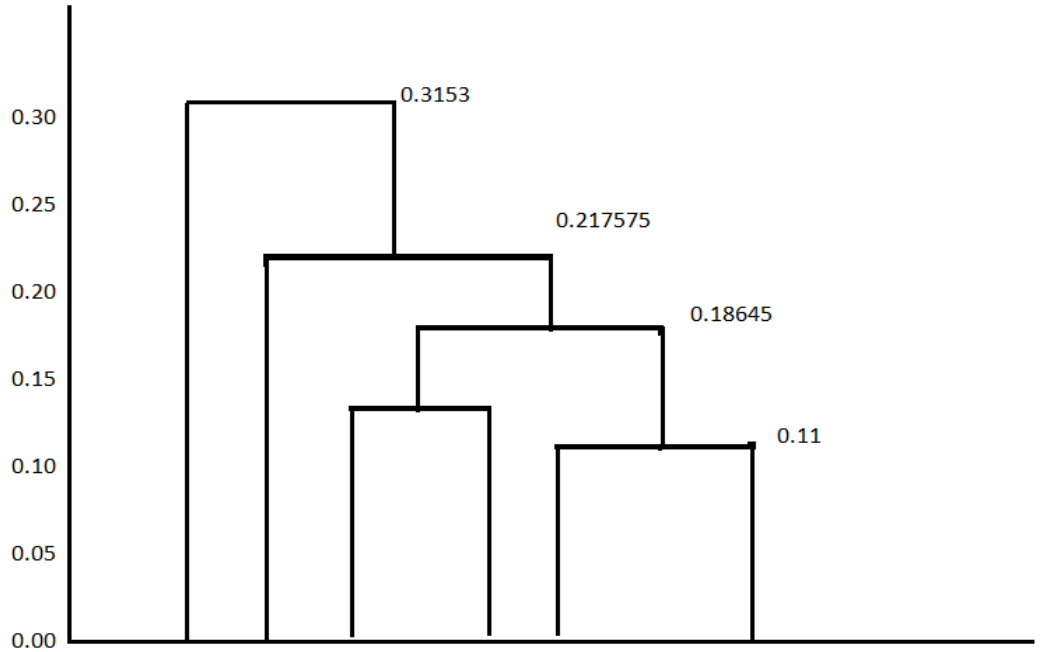
	p1	p(25)(36)	p4
p1	0	0.2618	0.3688
p(25)(36)	0.2618	0	0.21757 5
p4	0.3688	0.21757 5	0

0.21757 5

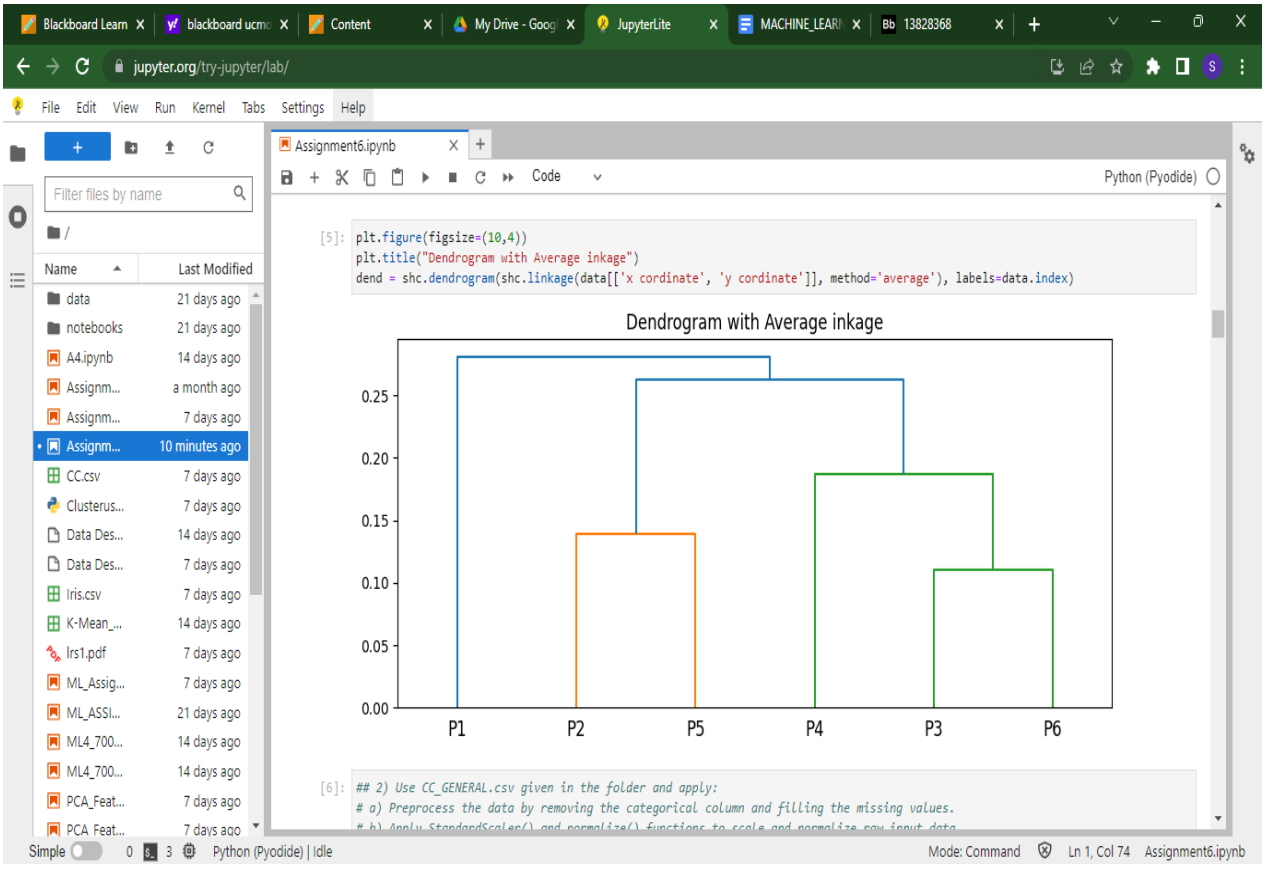
smallest distance from above data is

so p(25)(36)and p1 forms 4thcluster

	p1(25)(36)	p4
p1(25)(36)	0	0.3153
p4	0.3153	0



4 1 2 5 3 6



2) Use CC_GENERAL.csv given in the folder and apply:

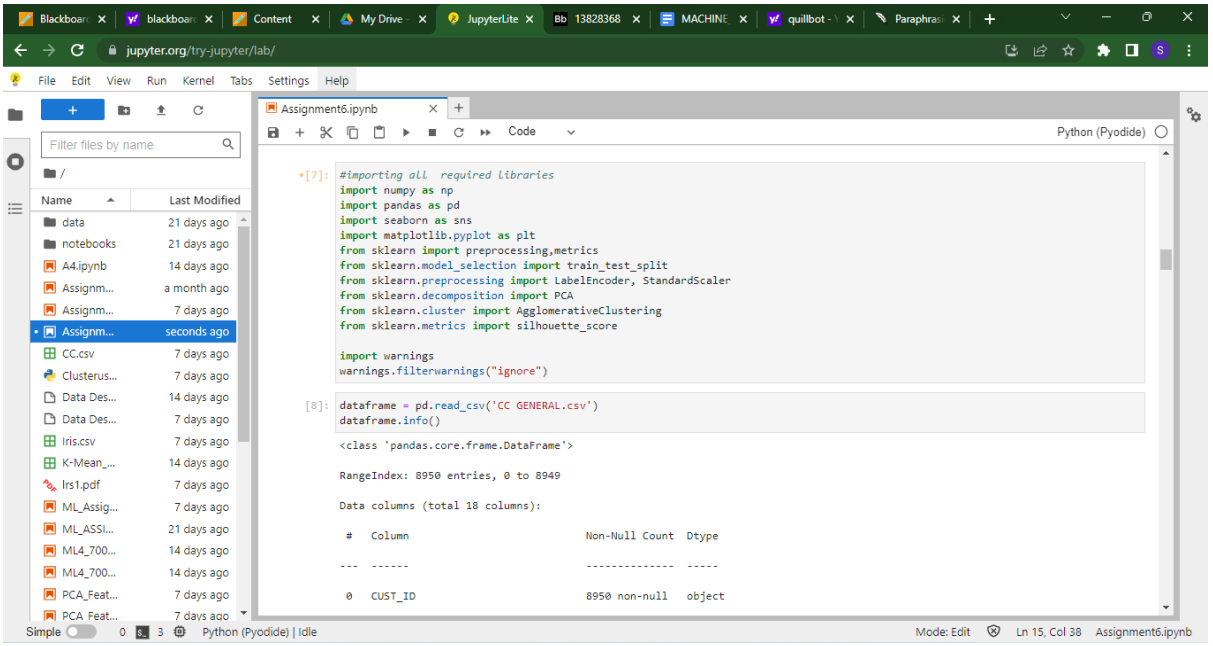
a) Preprocess the data by removing the categorical column and filling the missing values.

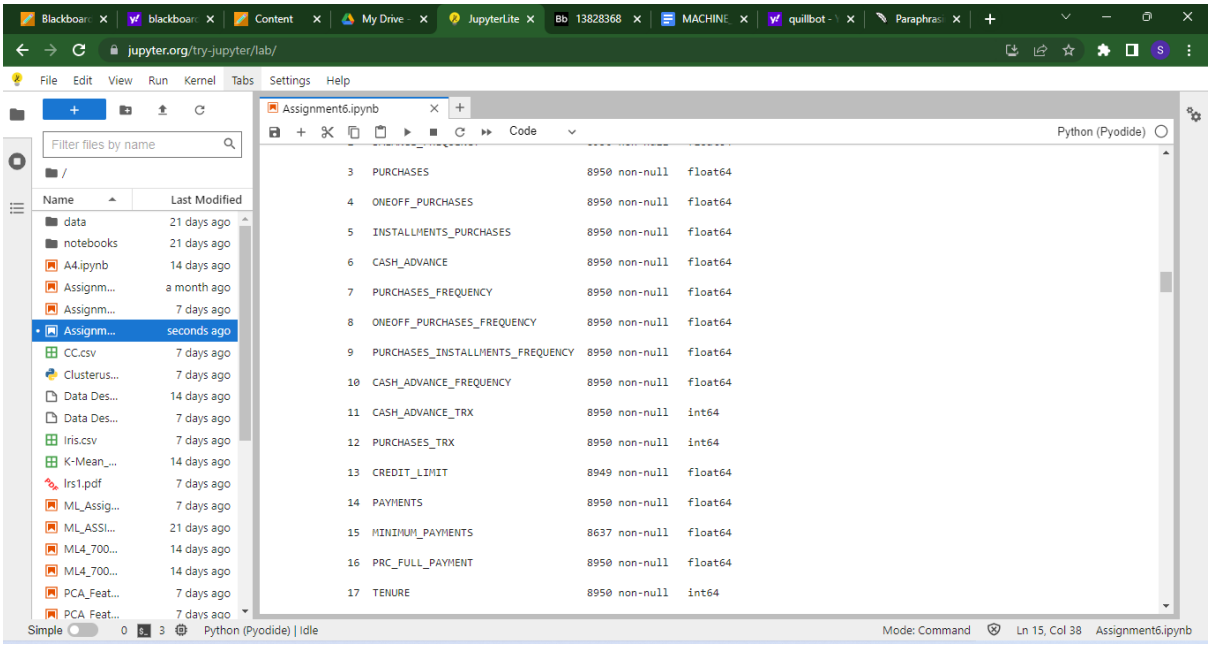
b) Apply StandardScaler() and normalize() functions to scale and normalize raw input data.

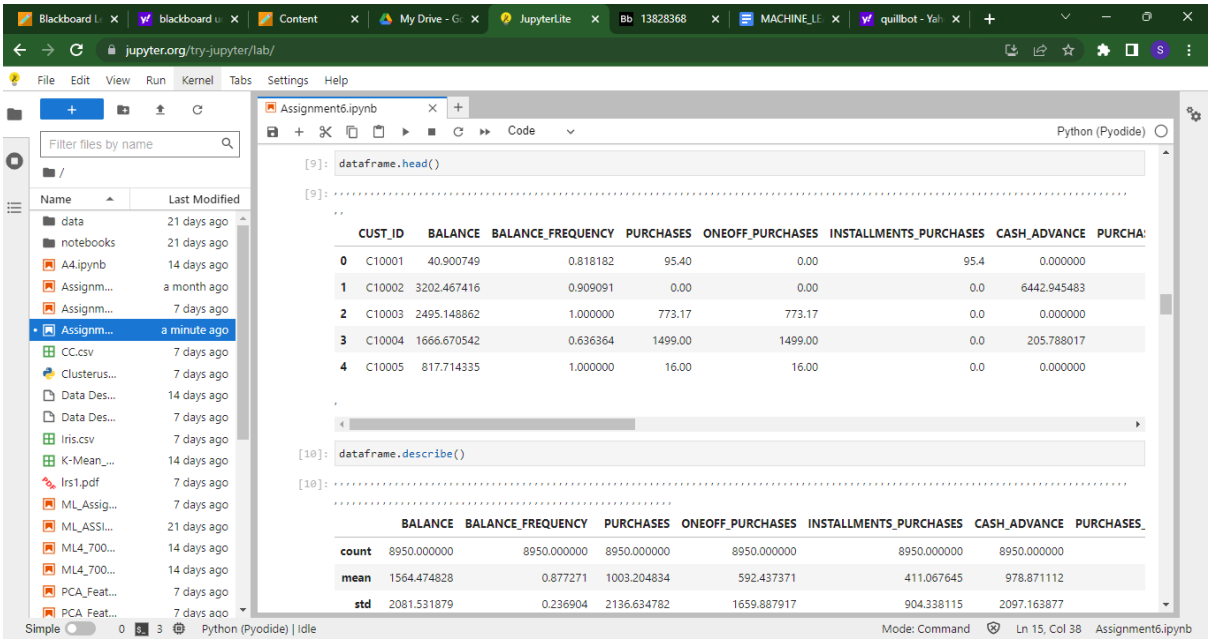
c) Use PCA with K=2 to reduce the input dimensions to two features.

d) Apply Agglomerative Clustering with k=2,3,4 and 5 on reduced features and visualize result for each k value using scatter plot.

e) Evaluate different variations using Silhouette Scores and Visualize results with a bar chart

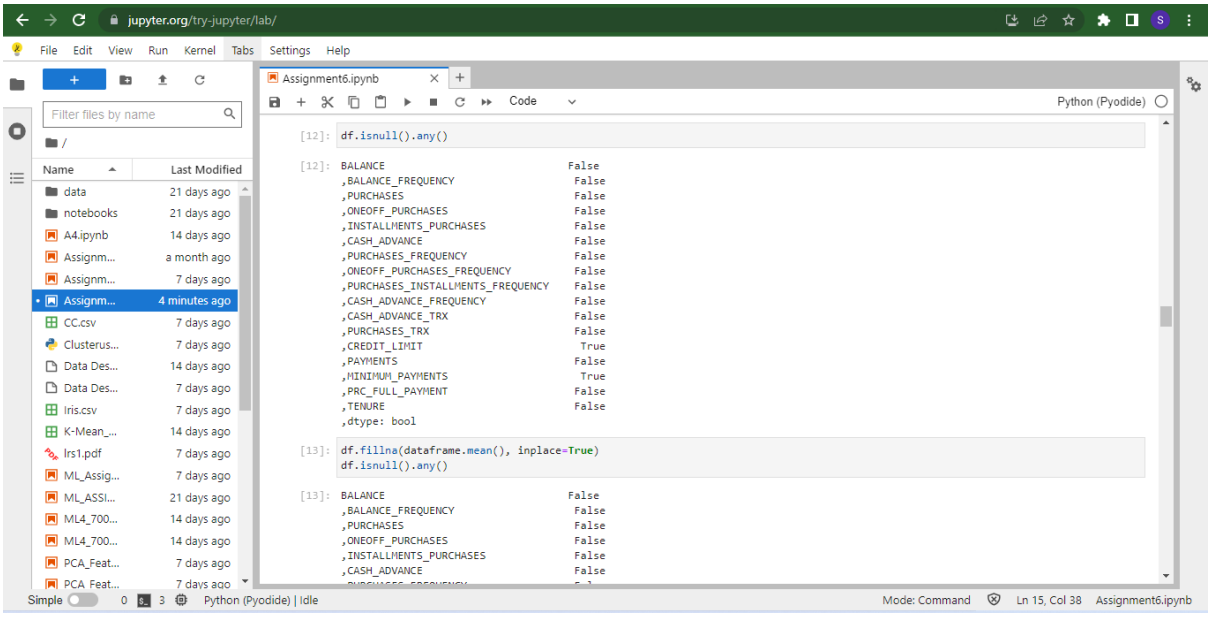

jupyter.org/try-jupyter/lab/
File Edit View Run Kernel Tabs Settings Help
Assignment6.ipynb
Code
Python (Pyodide)
Filter files by name
Name
Last Modified
[7]:
#importing all required libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing,metrics
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import AgglomerativeClustering
from sklearn.metrics import silhouette_score
import warnings
warnings.filterwarnings("ignore")
[8]:
dataframe = pd.read_csv('CC GENERAL.csv')
dataframe.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8950 entries, 0 to 8949
Data columns (total 18 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 CUST_ID 8950 non-null object
Simple
Python (Pyodide) | Idle
Mode: Edit
Ln 15, Col 38
Assignment6.ipynb

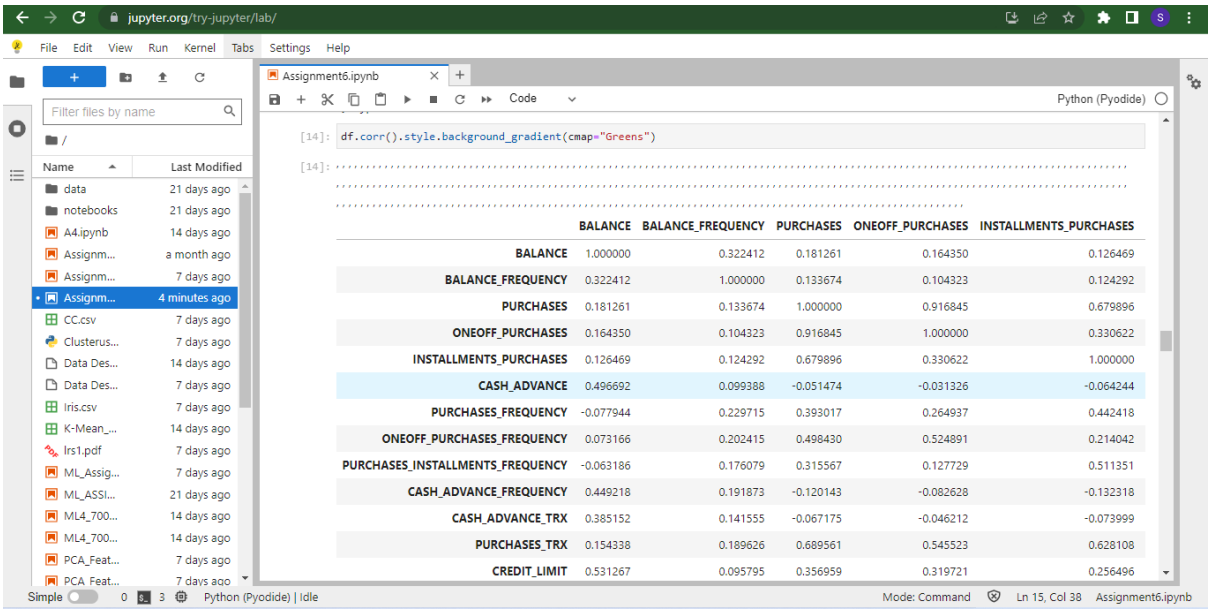

jupyter.org/try-jupyter/lab/
File Edit View Run Kernel Tabs Settings Help
Assignment6.ipynb
Code
Python (Pyodide)
3 PURCHASES 8950 non-null float64
4 ONEOFF_PURCHASES 8950 non-null float64
5 INSTALLMENTS_PURCHASES 8950 non-null float64
6 CASH_ADVANCE 8950 non-null float64
7 PURCHASES_FREQUENCY 8950 non-null float64
8 ONEOFF_PURCHASES_FREQUENCY 8950 non-null float64
9 PURCHASES_INSTALLMENTS_FREQUENCY 8950 non-null float64
10 CASH_ADVANCE_FREQUENCY 8950 non-null float64
11 CASH_ADVANCE_TRX 8950 non-null int64
12 PURCHASES_TRX 8950 non-null int64
13 CREDIT_LIMIT 8949 non-null float64
14 PAYMENTS 8950 non-null float64
15 MINIMUM_PAYMENTS 8637 non-null float64
16 PRC_FULL_PAYMENT 8950 non-null float64
17 TENURE 8950 non-null int64
Simple
Python (Pyodide) | Idle
Mode: Command
Ln 15, Col 38
Assignment6.ipynb

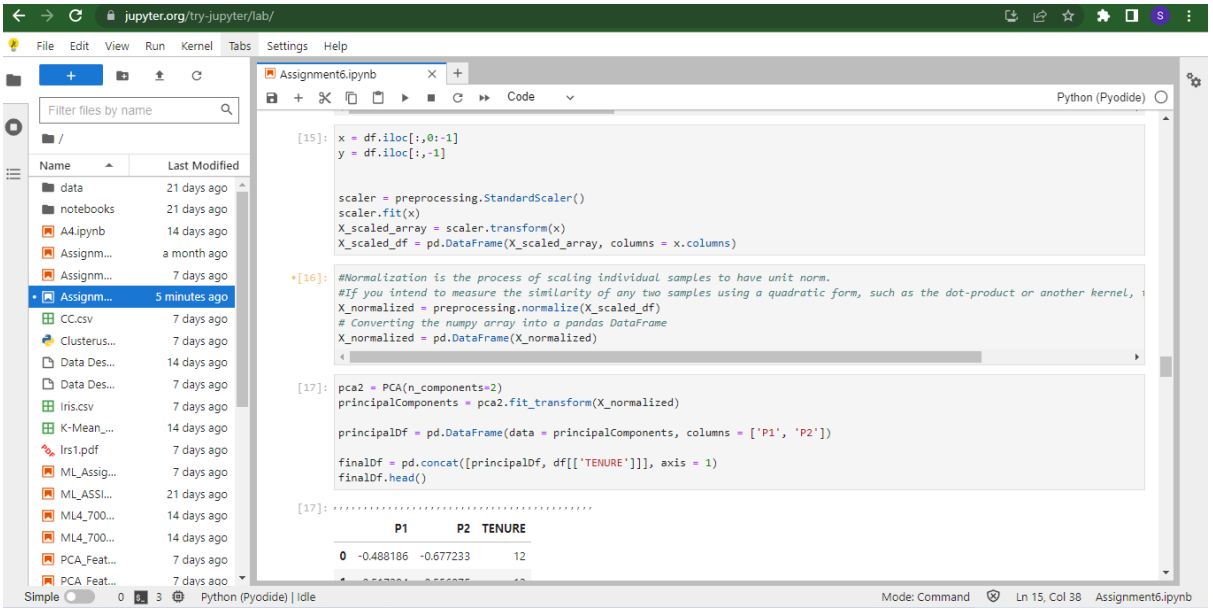


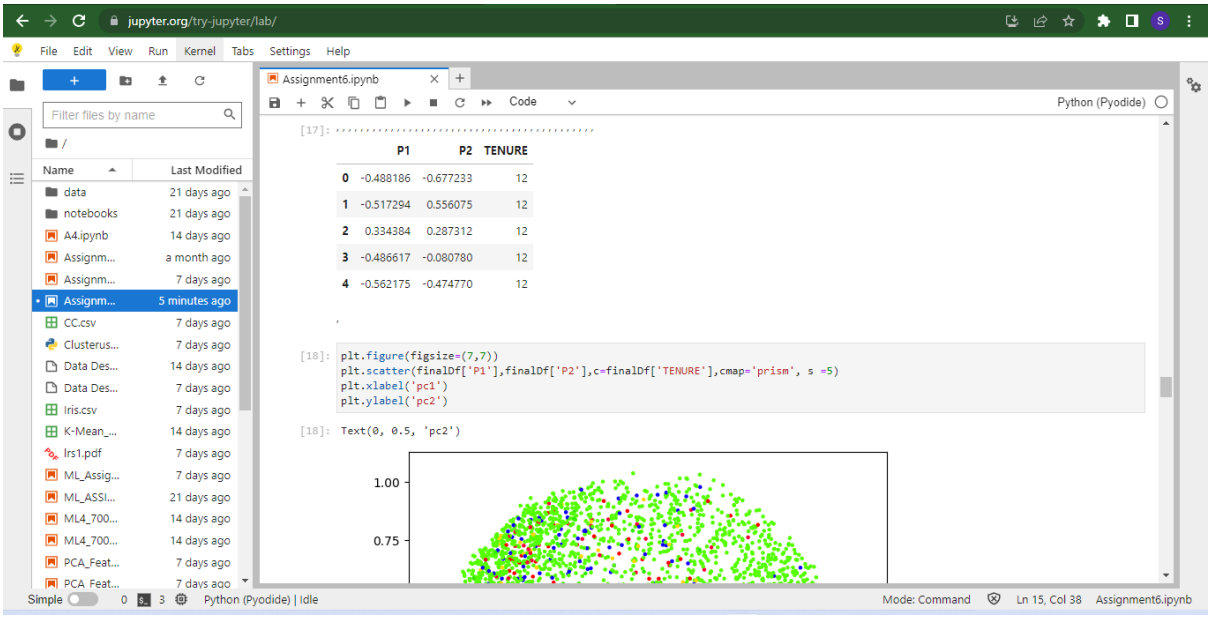
	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHA
0	C10001	40.900749	0.818182	95.40	0.00	95.4	0.000000	
1	C10002	3202.467416	0.909091	0.00	0.00	0.0	6442.945483	
2	C10003	2495.148862	1.000000	773.17	773.17	0.0	0.000000	
3	C10004	1666.670542	0.636364	1499.00	1499.00	0.0	205.788017	
4	C10005	817.714335	1.000000	16.00	16.00	0.0	0.000000	

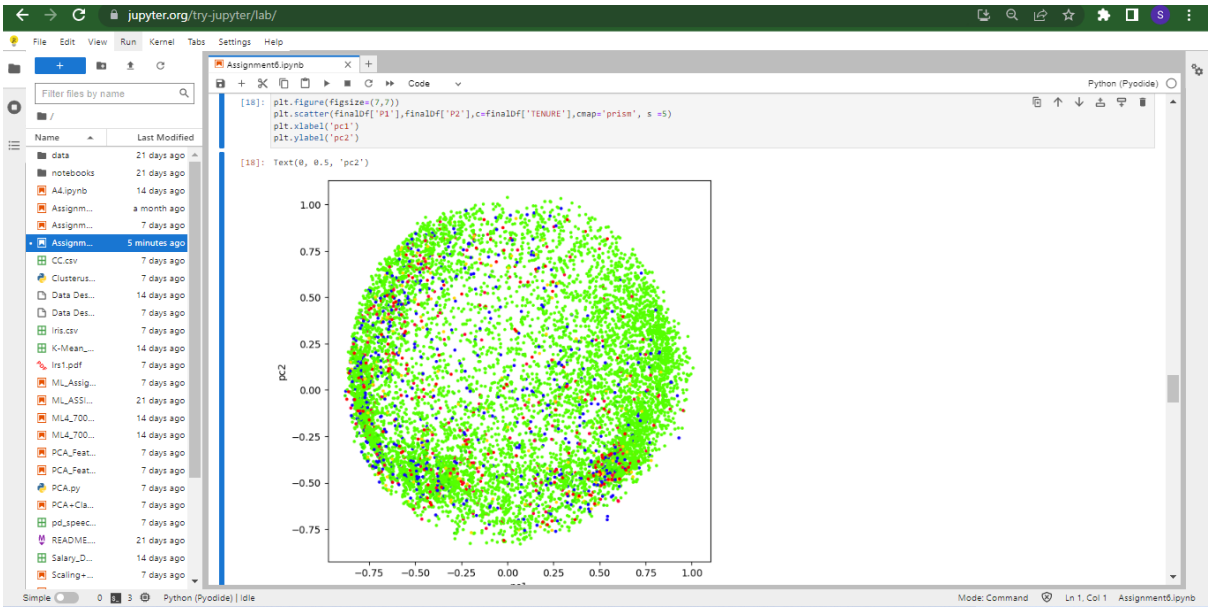
	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_
count	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000	
mean	1564.474828	0.877271	1003.204834	592.437371	411.067645	978.871112	
std	2081.531879	0.236904	2136.634782	1659.887917	904.338115	2097.163877	

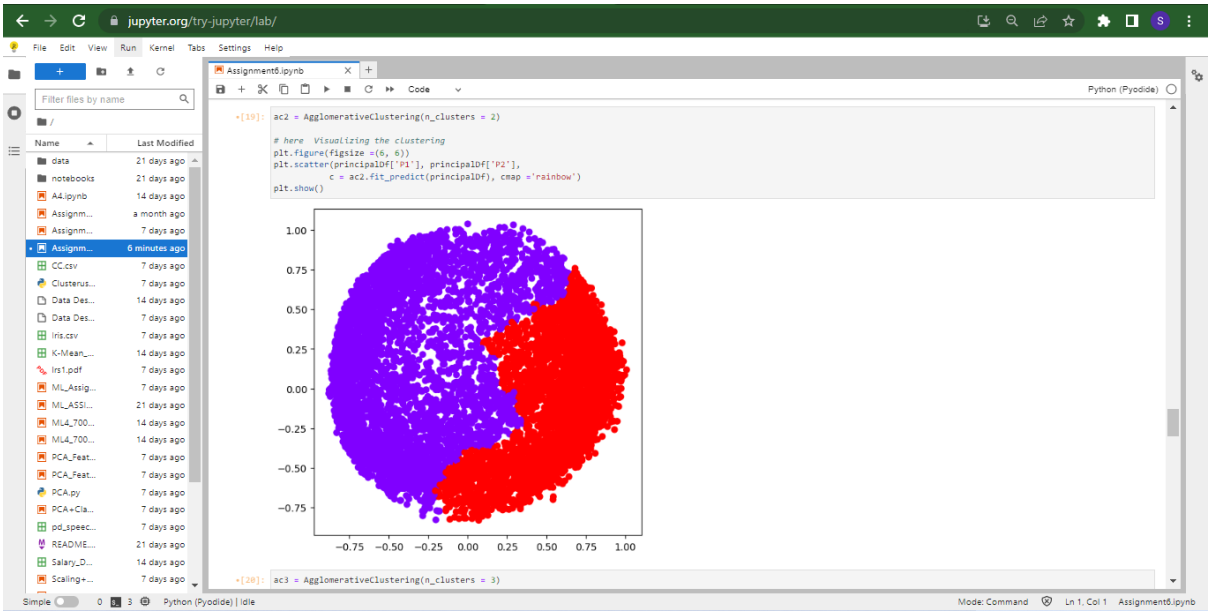

jupyter.org/try-jupyter/lab/
File Edit View Run Kernel Tabs Settings Help
Assignment6.ipynb
Code
Python (Pyodide)
Filter files by name
Name
Last Modified
data 21 days ago
notebooks 21 days ago
A4.ipynb 14 days ago
Assignm... a month ago
Assignm... 7 days ago
Assignm... 4 minutes ago
CC.csv 7 days ago
Clusterus... 7 days ago
Data Des... 14 days ago
Data Des... 7 days ago
Iris.csv 7 days ago
K-Mean_... 14 days ago
lrs1.pdf 7 days ago
ML_Assig... 7 days ago
ML_ASSI... 21 days ago
ML4_700... 14 days ago
ML4_700... 14 days ago
PCA_Feat... 7 days ago
[12]: df.isnull().any()
[12]: BALANCE False
,BALANCE_FREQUENCY False
,PURCHASES False
,ONEOFF_PURCHASES False
,INSTALLMENTS_PURCHASES False
,CASH_ADVANCE False
,PURCHASES_FREQUENCY False
,ONEOFF_PURCHASES_FREQUENCY False
,PURCHASES_INSTALLMENTS_FREQUENCY False
,CASH_ADVANCE_FREQUENCY False
,CASH_ADVANCE_TRX False
,PURCHASES_TRX False
,CREDIT_LIMIT True
,PAYMENTS False
,MINIMUM_PAYMENTS True
,PRC_FULL_PAYMENT False
,TENURE False
,dtype: bool
[13]: df.fillna(dataframe.mean(), inplace=True)
df.isnull().any()
[13]: BALANCE False
,BALANCE_FREQUENCY False
,PURCHASES False
,ONEOFF_PURCHASES False
,INSTALLMENTS_PURCHASES False
,CASH_ADVANCE False
Simple 0 3 Python (Pyodide) | Idle
Mode: Command Ln 15, Col 38 Assignment6.ipynb

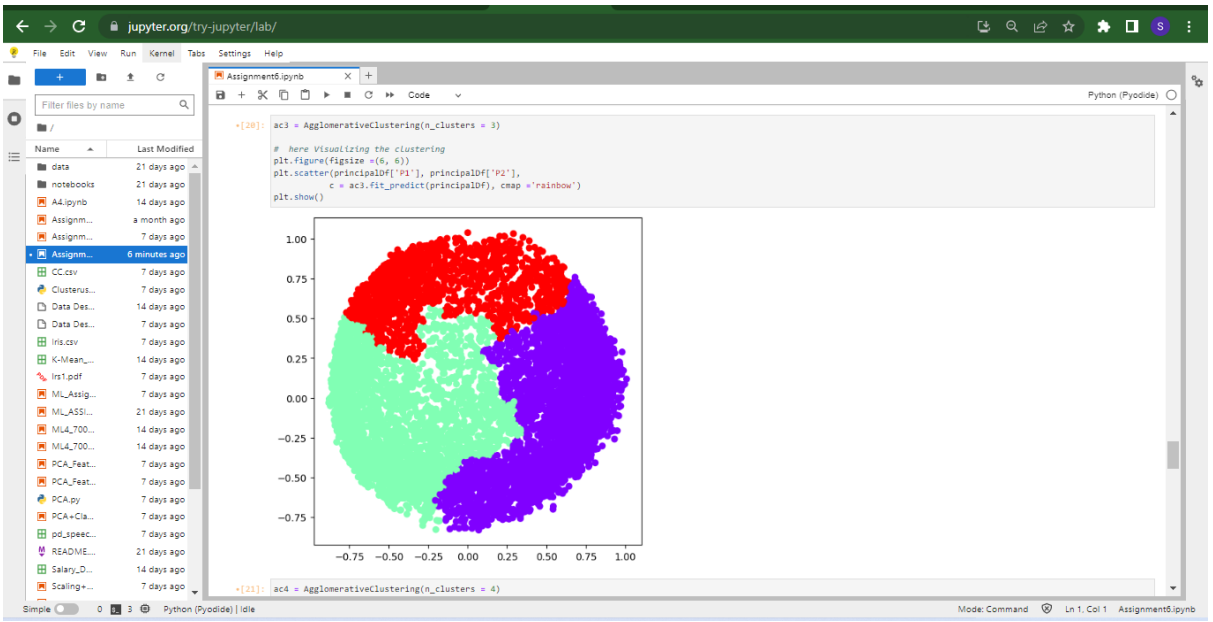

jupyter.org/try-jupyter/lab/
File Edit View Run Kernel Tabs Settings Help
Assignment6.ipynb
Code
Python (Pyodide)
Filter files by name
Name
Last Modified
data 21 days ago
notebooks 21 days ago
A4.ipynb 14 days ago
Assignm... a month ago
Assignm... 7 days ago
Assignm... 4 minutes ago
CC.csv 7 days ago
Clusterus... 7 days ago
Data Des... 14 days ago
Data Des... 7 days ago
Iris.csv 7 days ago
K-Mean_... 14 days ago
lrs1.pdf 7 days ago
ML_Assig... 7 days ago
ML_ASSI... 21 days ago
ML4_700... 14 days ago
ML4_700... 14 days ago
PCA_Feat... 7 days ago
[14]: df.corr().style.background_gradient(cmap="Greens")
[14]:
BALANCE BALANCE_FREQUENCY PURCHASES ONEOFF_PURCHASES INSTALLMENTS_PURCHASES
BALANCE 1.000000 0.322412 0.181261 0.164350 0.126469
BALANCE_FREQUENCY 0.322412 1.000000 0.133674 0.104323 0.124292
PURCHASES 0.181261 0.133674 1.000000 0.916845 0.679896
ONEOFF_PURCHASES 0.164350 0.104323 0.916845 1.000000 0.330622
INSTALLMENTS_PURCHASES 0.126469 0.124292 0.679896 0.330622 1.000000
CASH_ADVANCE 0.496692 0.099388 -0.051474 -0.031326 -0.064244
PURCHASES_FREQUENCY -0.077944 0.229715 0.393017 0.264937 0.442418
ONEOFF_PURCHASES_FREQUENCY 0.073166 0.202415 0.498430 0.524891 0.214042
PURCHASES_INSTALLMENTS_FREQUENCY -0.063186 0.176079 0.315567 0.127729 0.511351
CASH_ADVANCE_FREQUENCY 0.449218 0.191873 -0.120143 -0.082628 -0.132318
CASH_ADVANCE_TRX 0.385152 0.141555 -0.067175 -0.046212 -0.073999
PURCHASES_TRX 0.154338 0.189626 0.689561 0.545523 0.628108
CREDIT_LIMIT 0.531267 0.095795 0.356959 0.319721 0.256496
Simple 0 3 Python (Pyodide) | Idle
Mode: Command Ln 15, Col 38 Assignment6.ipynb


jupyter.org/try-jupyter/lab/
File Edit View Run Kernel Tabs Settings Help
Assignment6.ipynb
Code
Python (Pyodide)
Filter files by name
Name
Last Modified
data 21 days ago
notebooks 21 days ago
A4.ipynb 14 days ago
Assignm... a month ago
Assignm... 7 days ago
Assignm... 5 minutes ago
CC.csv 7 days ago
Clusterus... 7 days ago
Data Des... 14 days ago
Data Des... 7 days ago
Iris.csv 7 days ago
K-Mean_... 14 days ago
lrs1.pdf 7 days ago
ML_Assig... 7 days ago
ML_ASSI... 21 days ago
ML4_700... 14 days ago
ML4_700... 14 days ago
PCA_Feat... 7 days ago
[15]: x = df.iloc[:,0:-1]
y = df.iloc[:,-1]
scaler = preprocessing.StandardScaler()
scaler.fit(x)
X_scaled_array = scaler.transform(x)
X_scaled_df = pd.DataFrame(X_scaled_array, columns = x.columns)
[16]: #Normalization is the process of scaling individual samples to have unit norm.
#If you intend to measure the similarity of any two samples using a quadratic form, such as the dot-product or another kernel,
X_normalized = preprocessing.normalize(X_scaled_df)
# Converting the numpy array into a pandas DataFrame
X_normalized = pd.DataFrame(X_normalized)
[17]: pca2 = PCA(n_components=2)
principalComponents = pca2.fit_transform(X_normalized)
principalDf = pd.DataFrame(data = principalComponents, columns = ['P1', 'P2'])
finalDf = pd.concat([principalDf, df[['TENURE']]], axis = 1)
finalDf.head()
[17]:
P1 P2 TENURE
0 -0.488186 -0.677233 12
Simple 0 3 Python (Pyodide) | Idle
Mode: Command Ln 15, Col 38 Assignment6.ipynb


jupyter.org/try-jupyter/lab/
File Edit View Run Kernel Tabs Settings Help
Filter files by name
Name
Last Modified
Assignment6.ipynb
Code
Python (Pyodide)
[17]:
P1 P2 TENURE
0 -0.488186 -0.677233 12
1 -0.517294 0.556075 12
2 0.334384 0.287312 12
3 -0.486617 -0.080780 12
4 -0.562175 -0.474770 12
[18]:
plt.figure(figsize=(7,7))
plt.scatter(finalDf['P1'],finalDf['P2'],c=finalDf['TENURE'],cmap='prism', s =5)
plt.xlabel('pc1')
plt.ylabel('pc2')
[18]: Text(0, 0.5, 'pc2')
1.00
0.75
Simple
Python (Pyodide) | Idle
Mode: Command
Ln 15, Col 38
Assignment6.ipynb


jupyter.org/try-jupyter/lab/
File Edit View Run Kernel Tabs Settings Help
Assignment6.ipynb
Code
Python (Pyodide)
[18]:
plt.figure(figsize=(7,7))
plt.scatter(finalDf['P1'],finalDf['P2'],c=finalDf['TENURE'],cmap='prism', s =5)
plt.xlabel('pc1')
plt.ylabel('pc2')
[18]: Text(0, 0.5, 'pc2')
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
pc2
−0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
Simple
Python (Pyodide) | Idle
Mode: Command
Ln 1, Col 1
Assignment6.ipynb


jupyter.org/try-jupyter/lab/
File Edit View Run Kernel Tabs Settings Help
Assignment6.ipynb
Code
Python (Pyodide)
Filter files by name
Name
Last Modified
[19]: ac2 = AgglomerativeClustering(n_clusters = 2)
# here Visualizing the clustering
plt.figure(figsize =(6, 6))
plt.scatter(principalDf['P1'], principalDf['P2'],
c = ac2.fit_predict(principalDf), cmap ='rainbow')
plt.show()
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
[20]: ac3 = AgglomerativeClustering(n_clusters = 3)
Simple
Python (Pyodide) | Idle
Mode: Command
Ln 1, Col 1
Assignment6.ipynb


jupyter.org/try-jupyter/lab/
File Edit View Run Kernel Tabs Settings Help
Assignment6.ipynb
Code
Python (Pyodide)
Filter files by name
Name
Last Modified
[20]: ac3 = AgglomerativeClustering(n_clusters = 3)
# here Visualizing the clustering
plt.figure(figsize =(6, 6))
plt.scatter(principalDf['P1'], principalDf['P2'],
c = ac3.fit_predict(principalDf), cmap ='rainbow')
plt.show()
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
[21]: ac4 = AgglomerativeClustering(n_clusters = 4)
Simple
Python (Pyodide) | Idle
Mode: Command
Ln 1, Col 1
Assignment6.ipynb

jupyter.org/try-jupyter/lab/

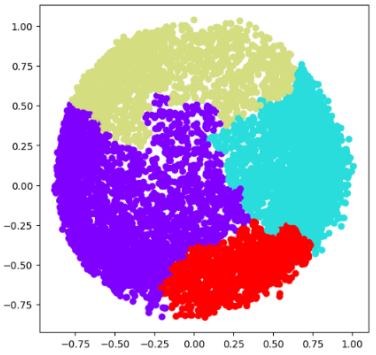
File Edit View Run Kernel Tabs Settings Help

Assignment6.ipynb

Code

Python (Pyodide)

```
[21]: ac4 = AgglomerativeClustering(n_clusters = 4)

      # here Visualizing the clustering
      plt.figure(figsize =(6, 6))
      plt.scatter(principalDf['P1'], principalDf['P2'],
                  c = ac4.fit_predict(principalDf), cmap ='rainbow')

      plt.show()
```




```
[22]: ac5 = AgglomerativeClustering(n_clusters = 5)
```

Simple 0 3 Python (Pyodide) | Idle Mode: Command Ln 1, Col 1 Assignment6.ipynb

jupyter.org/try-jupyter/lab/

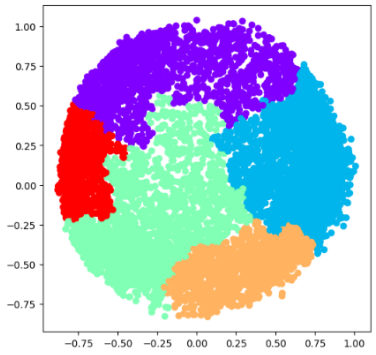
File Edit View Run Kernel Tabs Settings Help

Assignment6.ipynb

Code

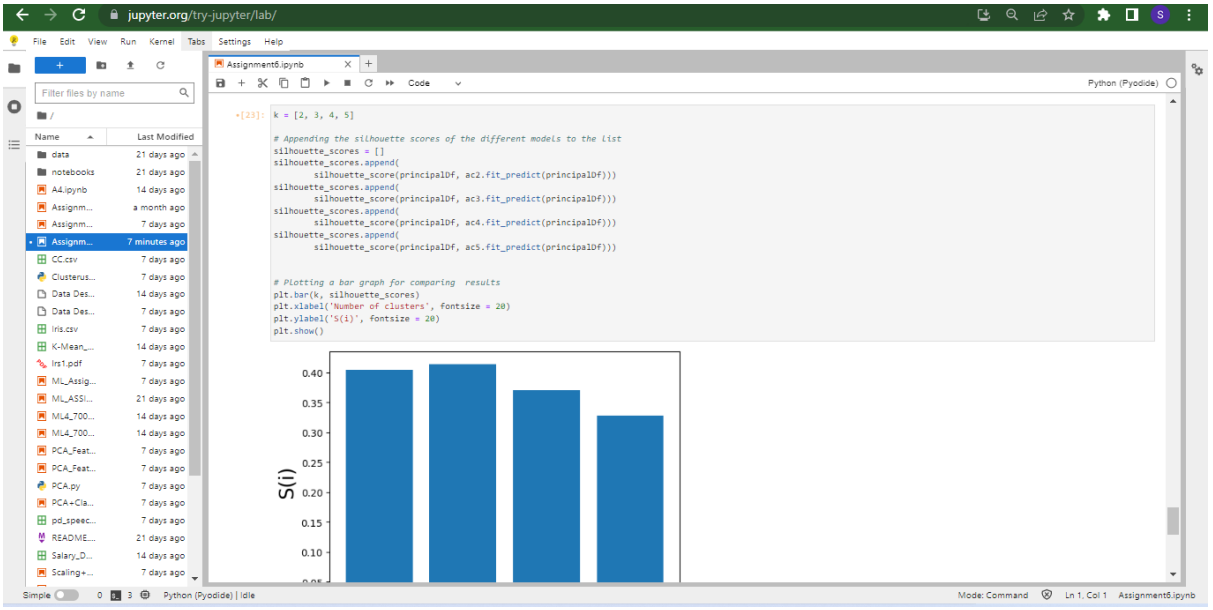
Python (Pyodide)

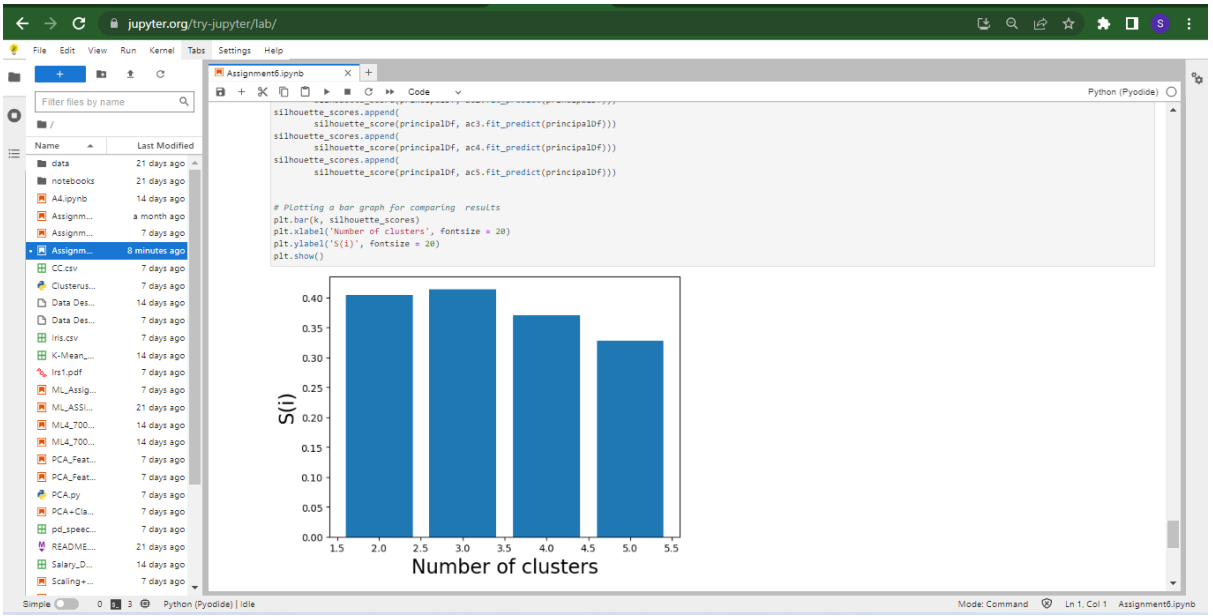
```
[22]: ac5 = AgglomerativeClustering(n_clusters = 5)

      # Visualizing the clustering
      plt.figure(figsize =(6, 6))
      plt.scatter(principalDf['P1'], principalDf['P2'],
                  c = ac5.fit_predict(principalDf), cmap ='rainbow')
      plt.show()
```




```
[23]: k = [2, 3, 4, 5]
```

Simple 0 3 Python (Pyodide) | Idle Mode: Command Ln 1, Col 1 Assignment6.ipynb


jupyter.org/try-jupyter/lab/
File Edit View Run Kernel Tabs Settings Help
Assignment6.ipynb
Code
Python (Pyodide)
Filter files by name
Name
Last Modified
[23]:
k = [2, 3, 4, 5]
# Appending the silhouette scores of the different models to the list
silhouette_scores = []
silhouette_scores.append(
silhouette_score(principalDf, ac2.fit_predict(principalDf)))
silhouette_scores.append(
silhouette_score(principalDf, ac3.fit_predict(principalDf)))
silhouette_scores.append(
silhouette_score(principalDf, ac4.fit_predict(principalDf)))
silhouette_scores.append(
silhouette_score(principalDf, ac5.fit_predict(principalDf)))
# Plotting a bar graph for comparing results
plt.bar(k, silhouette_scores)
plt.xlabel('Number of clusters', fontsize = 20)
plt.ylabel('S(i)', fontsize = 20)
plt.show()
S(i)
0.40
0.35
0.30
0.25
0.20
0.15
0.10
Simple
Python (Pyodide) | Idle
Mode: Command
Ln 1, Col 1
Assignment6.ipynb


jupyter.org/try-jupyter/lab/
File Edit View Run Kernel Tabs Settings Help
Assignment6.ipynb
Code
Python (Pyodide)
Filter files by name
Name
Last Modified
silhouette_scores.append(
silhouette_score(principalDf, ac3.fit_predict(principalDf)))
silhouette_scores.append(
silhouette_score(principalDf, ac4.fit_predict(principalDf)))
silhouette_scores.append(
silhouette_score(principalDf, ac5.fit_predict(principalDf)))
# Plotting a bar graph for comparing results
plt.bar(k, silhouette_scores)
plt.xlabel('Number of clusters', fontsize = 20)
plt.ylabel('S(i)', fontsize = 20)
plt.show()
S(i)
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
Number of clusters
Simple
Python (Pyodide) | Idle
Mode: Command
Ln 1, Col 1
Assignment6.ipynb