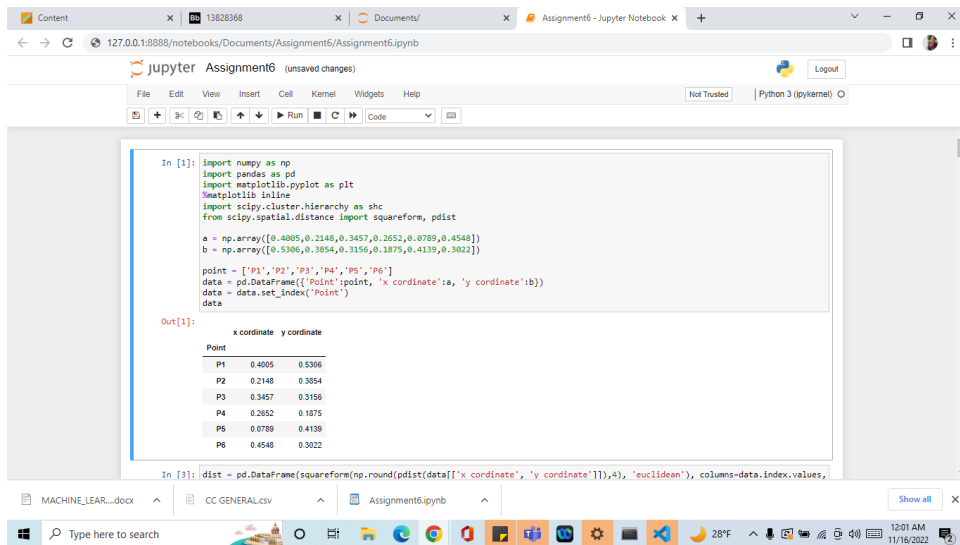


## 1) Dataframe with points P1-P6 where 'a' is x-coordinate and 'b' is y-coordinate



The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import scipy.cluster.hierarchy as shc
from scipy.spatial.distance import squareform, pdist

a = np.array([0.4005, 0.2148, 0.3457, 0.2652, 0.0789, 0.4548])
b = np.array([0.5306, 0.3854, 0.3156, 0.1875, 0.4139, 0.3022])

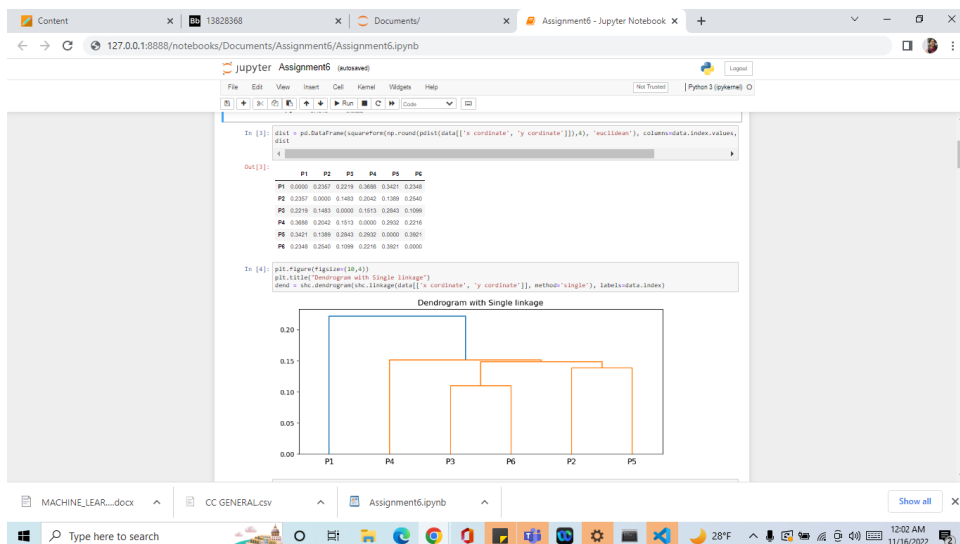
point = ['P1', 'P2', 'P3', 'P4', 'P5', 'P6']
data = pd.DataFrame({'Point': point, 'x coordinate': a, 'y coordinate': b})
data = data.set_index('Point')
```

Out[1]:

	x coordinate	y coordinate
P1	0.4005	0.5306
P2	0.2148	0.3854
P3	0.3457	0.3156
P4	0.2652	0.1875
P5	0.0789	0.4139
P6	0.4548	0.3022

In [3]: dist = pd.DataFrame(squareform(np.round(pdist(data[['x coordinate', 'y coordinate']], 4), 'euclidean'), columns=data.index.values,

## Finding the Euclidean distance



The screenshot shows the continuation of the Jupyter Notebook with the following code and output:

```
In [3]: dist = pd.DataFrame(squareform(np.round(pdist(data[['x coordinate', 'y coordinate']], 4), 'euclidean'), columns=data.index.values,
dist

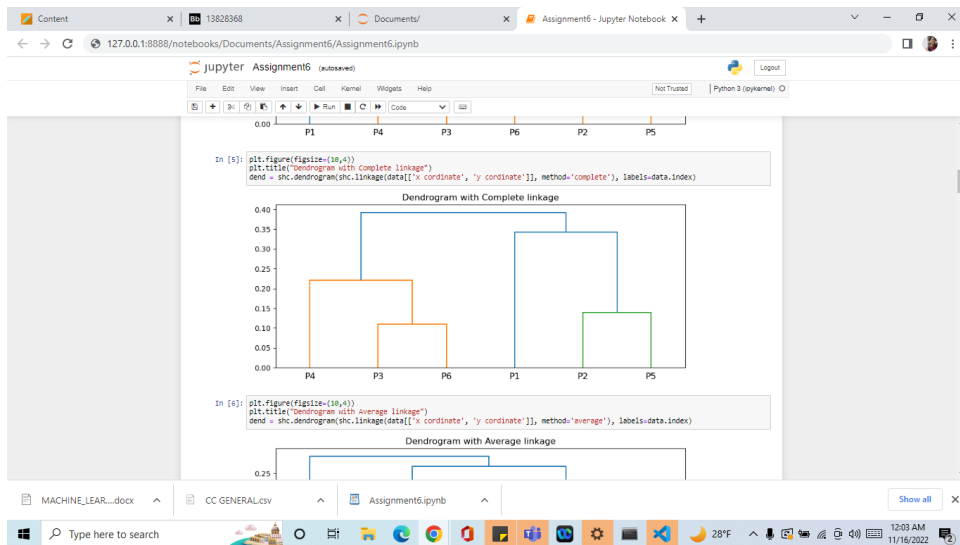
Out[3]:
```

	P1	P2	P3	P4	P5	P6
P1	0.0000	0.2207	0.2219	0.3688	0.3421	0.2546
P2	0.2207	0.0000	0.1403	0.2042	0.1989	0.2540
P3	0.2219	0.1403	0.0000	0.1513	0.2843	0.1086
P4	0.3688	0.2042	0.1513	0.0000	0.2802	0.2216
P5	0.3421	0.1989	0.2843	0.2802	0.0000	0.3621
P6	0.2546	0.2540	0.1086	0.2216	0.3621	0.0000

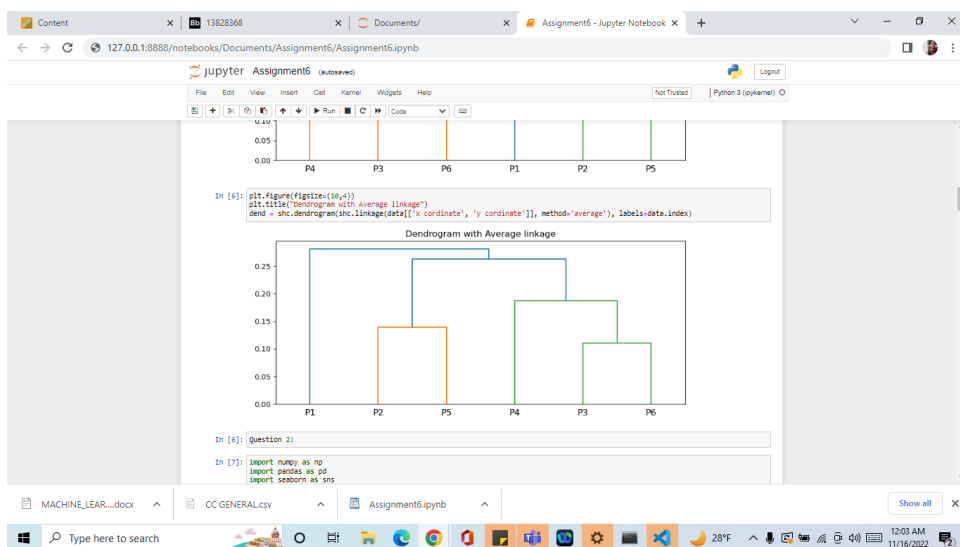
```
In [4]: plt.figure(figsize=(8,4))
plt.title('Dendrogram with Single linkage')
dend = shc.dendrogram(shc.linkage(data[['x coordinate', 'y coordinate']], method='single'), labels=data.index)
```

The output is a dendrogram titled "Dendrogram with Single linkage" showing the hierarchical clustering of points P1 through P6. The x-axis labels are P1, P4, P3, P6, P2, P5. The y-axis represents distance from 0.00 to 0.20. The dendrogram shows that P1 and P4 are the closest, followed by P3 and P6, then P2 and P5, and finally these groups merge into a larger cluster.

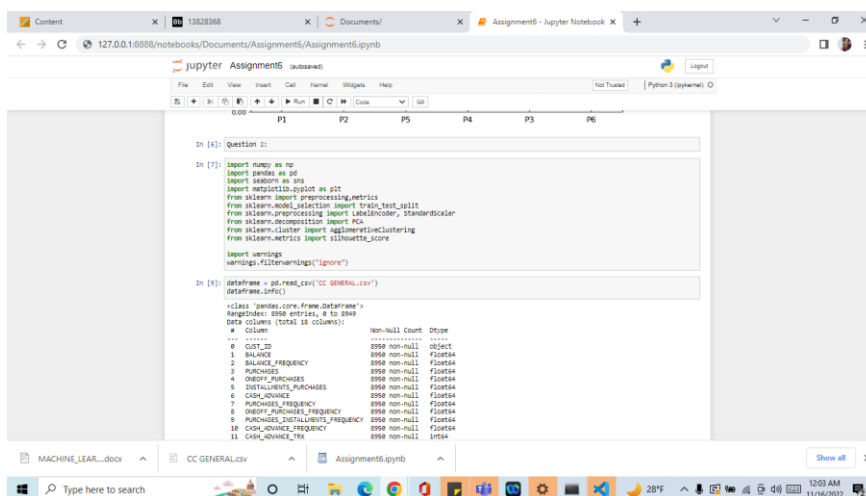
## Dendrogram linkage



## Dendrogram with Average Linkage



## 2. Reading the .csv file and got the info of dataframe



```
Content 13828368 Documents/ Assignment6 - Jupyter Notebook
127.0.0.1:8888/notebooks/Documents/Assignment6/Assignment6.ipynb

jupyter Assignment6 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

In [10]: dataframe.head()
Out[10]:
```

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY
0	C10001	42.000740	0.018182	95.40	0.00	95.4	0.000000	0.100
1	C10002	3202.407416	0.000091	0.00	0.00	0.0	0.442.845483	0.000
2	C10003	2495.148892	1.000000	773.17	773.17	0.0	0.000000	1.000
3	C10004	1688.870542	0.038384	1490.00	1490.00	0.0	206.788017	0.083
4	C10005	817.714335	1.000000	15.00	15.00	0.0	0.000000	0.083

```

In [12]: dataframe.describe()
Out[12]:
```

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY
count	8940.000000	8940.000000	8940.000000	8940.000000	8940.000000	8940.000000	8940.000000
mean	1594.474628	0.877271	1053.204634	562.437371	411.007945	678.871112	0.460261
std	2081.831879	0.238604	2138.034782	1866.887917	904.338115	2297.163877	0.401371
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	128.281915	0.888889	38.858000	0.000000	0.000000	0.000000	0.083333
50%	873.385231	1.000000	381.280000	38.000000	89.000000	0.000000	0.800000
75%	2054.148236	1.000000	1115.130000	877.405000	468.837800	1113.821158	0.918889
max	16043.138000	1.000000	40038.870000	40761.280000	22600.000000	47137.211700	1.000000

```

In [13]: df = dataframe.drop(['CUST_ID'], axis=1)
df.head()
Out[13]:
```

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ONEC
0	42.000740	0.018182	95.40	0.00	95.4	0.000000	0.100887	0.100887
1	3202.407416	0.000091	0.00	0.00	0.0	0.442.845483	0.000000	0.000000
2	2495.148892	1.000000	773.17	773.17	0.0	0.000000	1.000000	1.000000
3	1688.870542	0.038384	1490.00	1490.00	0.0	206.788017	0.083333	0.083333
4	817.714335	1.000000	15.00	15.00	0.0	0.000000	0.083333	0.083333

Checking if there any null values in the dataframe

```
Content 13828368 Documents/ Assignment6 - Jupyter Notebook
127.0.0.1:8888/notebooks/Documents/Assignment6/Assignment6.ipynb

jupyter Assignment6 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

In [13]: df = dataframe.drop(['CUST_ID'], axis=1)
df.head()
Out[13]:
```

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ONEC
0	42.000740	0.018182	95.40	0.00	95.4	0.000000	0.100887	0.100887
1	3202.407416	0.000091	0.00	0.00	0.0	0.442.845483	0.000000	0.000000
2	2495.148892	1.000000	773.17	773.17	0.0	0.000000	1.000000	1.000000
3	1688.870542	0.038384	1490.00	1490.00	0.0	206.788017	0.083333	0.083333
4	817.714335	1.000000	15.00	15.00	0.0	0.000000	0.083333	0.083333

```

In [14]: df.isnull().any()
Out[14]:
```

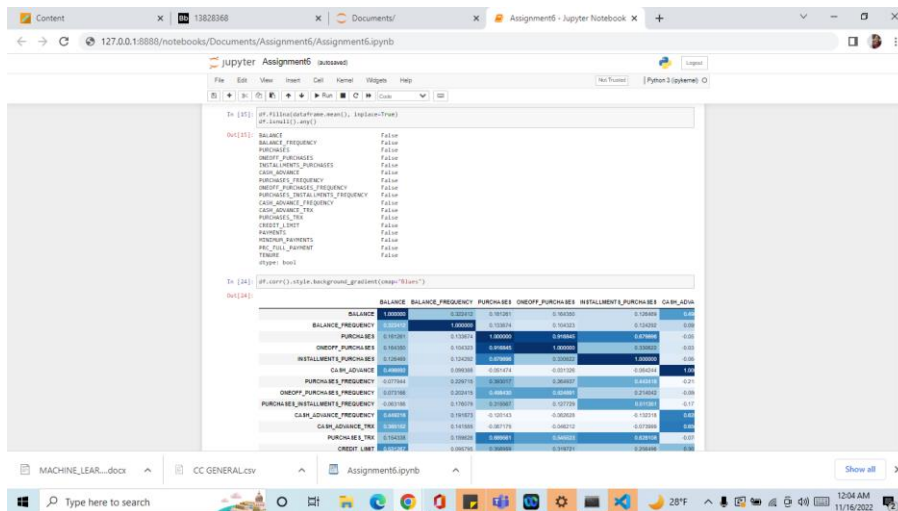
	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ONEC
BALANCE	False	False	False	False	False	False	False	False
BALANCE_FREQUENCY	False	False	False	False	False	False	False	False
PURCHASES	False	False	False	False	False	False	False	False
ONEOFF_PURCHASES	False	False	False	False	False	False	False	False
INSTALLMENTS_PURCHASES	False	False	False	False	False	False	False	False
CASH_ADVANCE	False	False	False	False	False	False	False	False
PURCHASES_FREQUENCY	False	False	False	False	False	False	False	False
ONEOFF_PURCHASES_FREQUENCY	False	False	False	False	False	False	False	False
PURCHASES_INSTALLMENTS_FREQUENCY	False	False	False	False	False	False	False	False
CASH_ADVANCE_FREQUENCY	False	False	False	False	False	False	False	False
CASH_ADVANCE_TRX	False	False	False	False	False	False	False	False
PURCHASES_TRX	False	False	False	False	False	False	False	False
CREDIT_LIMIT	True	True	True	True	True	True	True	True
PAIDMENTS	False	False	False	False	False	False	False	False
INDIVIDUAL_PAYMENTS	True	True	True	True	True	True	True	True
PRC_FULL_PAYMENT	False	False	False	False	False	False	False	False
TENURE	False	False	False	False	False	False	False	False
dtype:	bool	bool	bool	bool	bool	bool	bool	bool

```

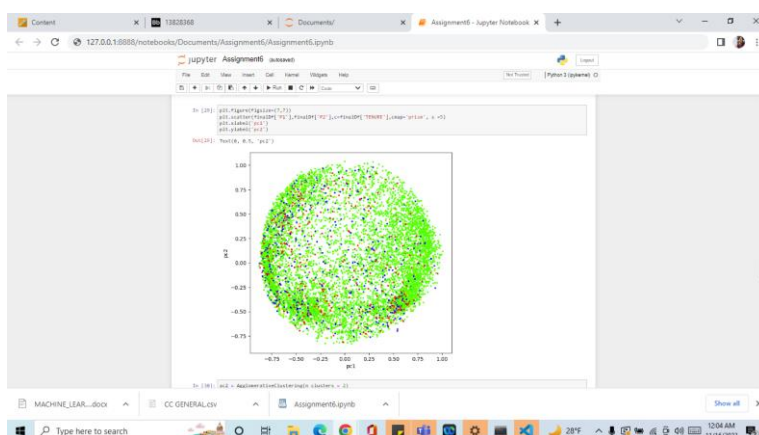
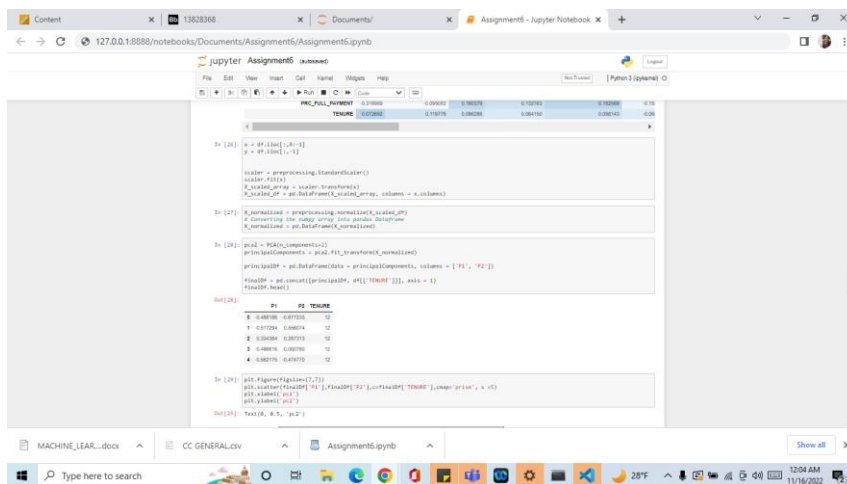
In [15]: df.fillna(dataframe.mean(), inplace=True)
df.isnull().any()
Out[15]:
```

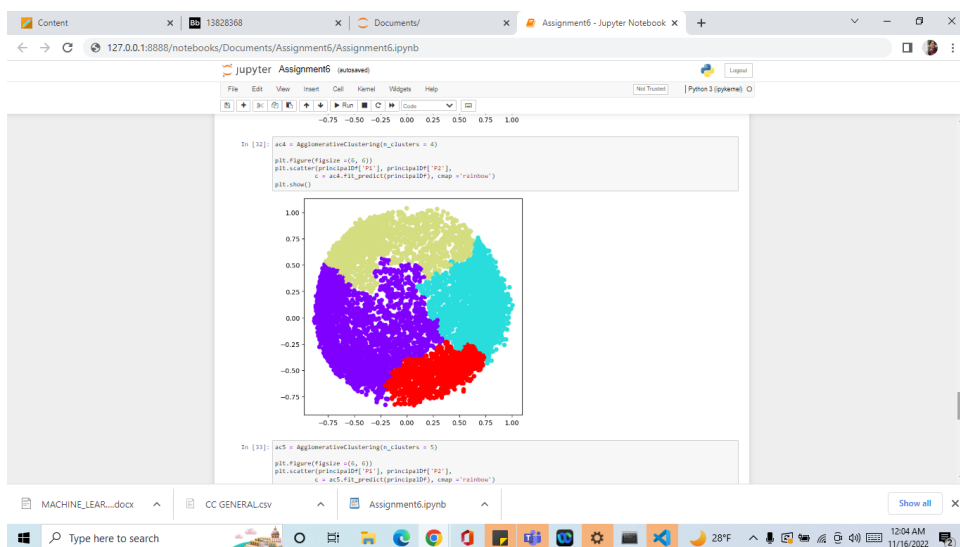
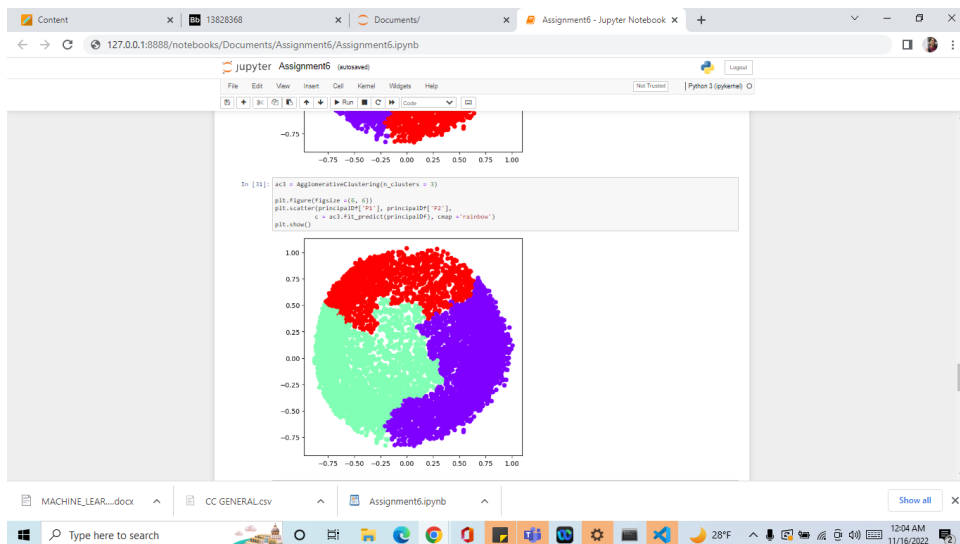
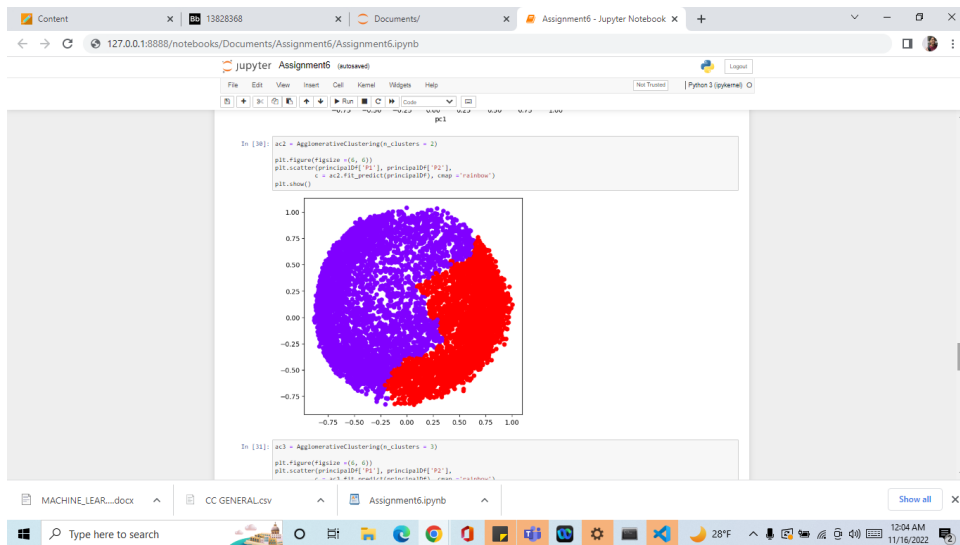
	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ONEC
BALANCE	False	False	False	False	False	False	False	False
BALANCE_FREQUENCY	False	False	False	False	False	False	False	False
PURCHASES	False	False	False	False	False	False	False	False
ONEOFF_PURCHASES	False	False	False	False	False	False	False	False
INSTALLMENTS_PURCHASES	False	False	False	False	False	False	False	False
CASH_ADVANCE	False	False	False	False	False	False	False	False
PURCHASES_FREQUENCY	False	False	False	False	False	False	False	False
ONEOFF_PURCHASES_FREQUENCY	False	False	False	False	False	False	False	False
PURCHASES_INSTALLMENTS_FREQUENCY	False	False	False	False	False	False	False	False
CASH_ADVANCE_FREQUENCY	False	False	False	False	False	False	False	False
CASH_ADVANCE_TRX	False	False	False	False	False	False	False	False
PURCHASES_TRX	False	False	False	False	False	False	False	False
CREDIT_LIMIT	True	True	True	True	True	True	True	True
PAIDMENTS	False	False	False	False	False	False	False	False
INDIVIDUAL_PAYMENTS	True	True	True	True	True	True	True	True
PRC_FULL_PAYMENT	False	False	False	False	False	False	False	False
TENURE	False	False	False	False	False	False	False	False
dtype:	bool	bool	bool	bool	bool	bool	bool	bool

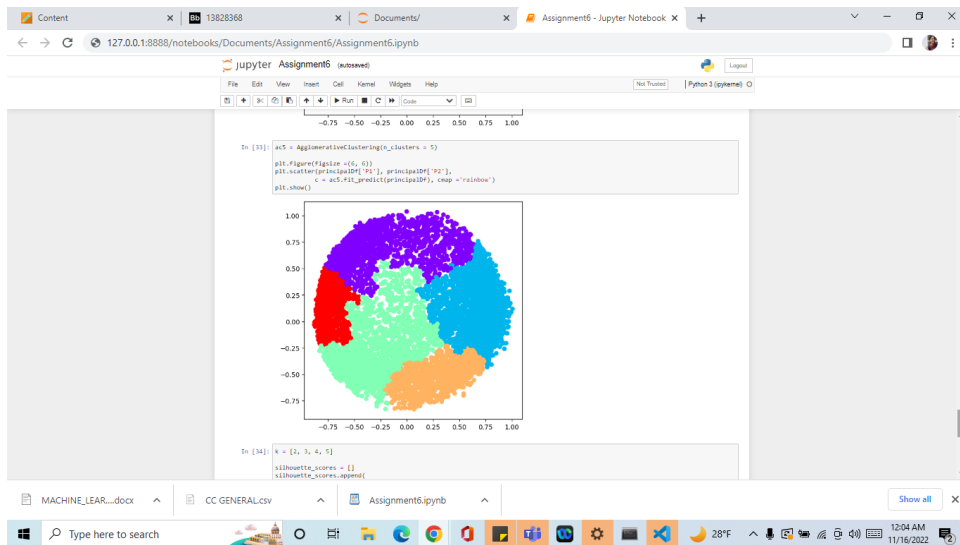
Replacing the null values with mean



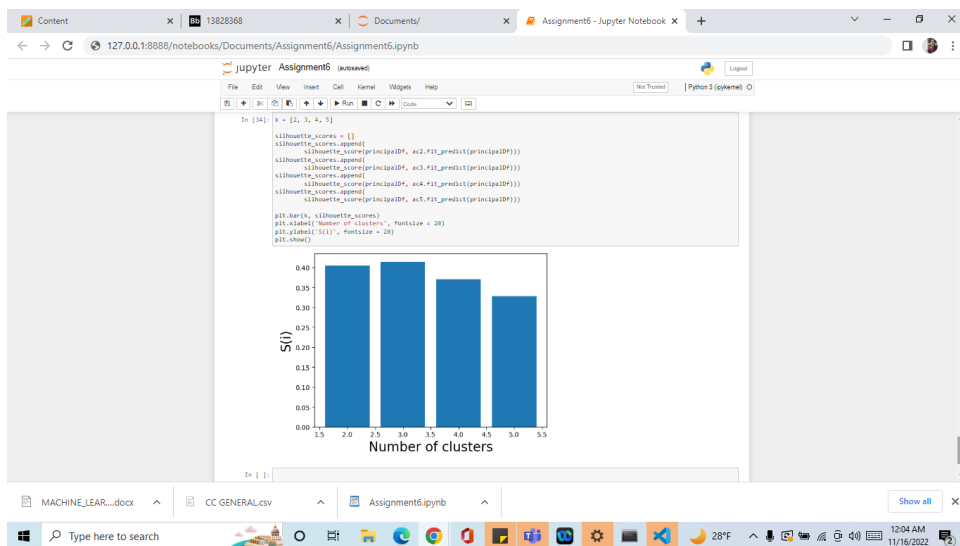
## Converting numpy into pandas dataframe







Evaluating different variations using Silhouette Scores and Visualize results with a bar chart.



Question 1:

Single Link Proximity:

- 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

<b>p3</b>	0.2218	0.1483	0	0.1513	0.2843	0.11
<b>p4</b>	0.3688	0.2042	0.1513	0	0.2932	0.2216
<b>p5</b>	0.3421	0.1388	0.2843	0.2932	0	0.3921
<b>p6</b>	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

	<b>p1</b>	<b>p2</b>	<b>p36</b>	<b>p4</b>	<b>p5</b>
<b>p1</b>	0	0.2357	0.2218	0.3688	0.3421
<b>p2</b>	0.2357	0	0.1483	0.2042	0.1388
<b>p36</b>	0.2218	0.1483	0	0.1513	0.2843
<b>p4</b>	0.3688	0.2042	0.1513	0	0.2932
<b>p5</b>	0.3421	0.1388	0.2843	0.2932	0

Smallest distance is 0.1388

so p2 and p5 forms 2nd cluster

	<b>p1</b>	<b>p25</b>	<b>p36</b>	<b>p4</b>
<b>p1</b>	0	0.2357	0.2218	0.3688
<b>p25</b>	0.2357	0	0.1483	0.2042
<b>p36</b>	0.2218	0.1483	0	0.1513
<b>p4</b>	0.3688	0.2042	0.1513	0

Smallest distance is 0.1483

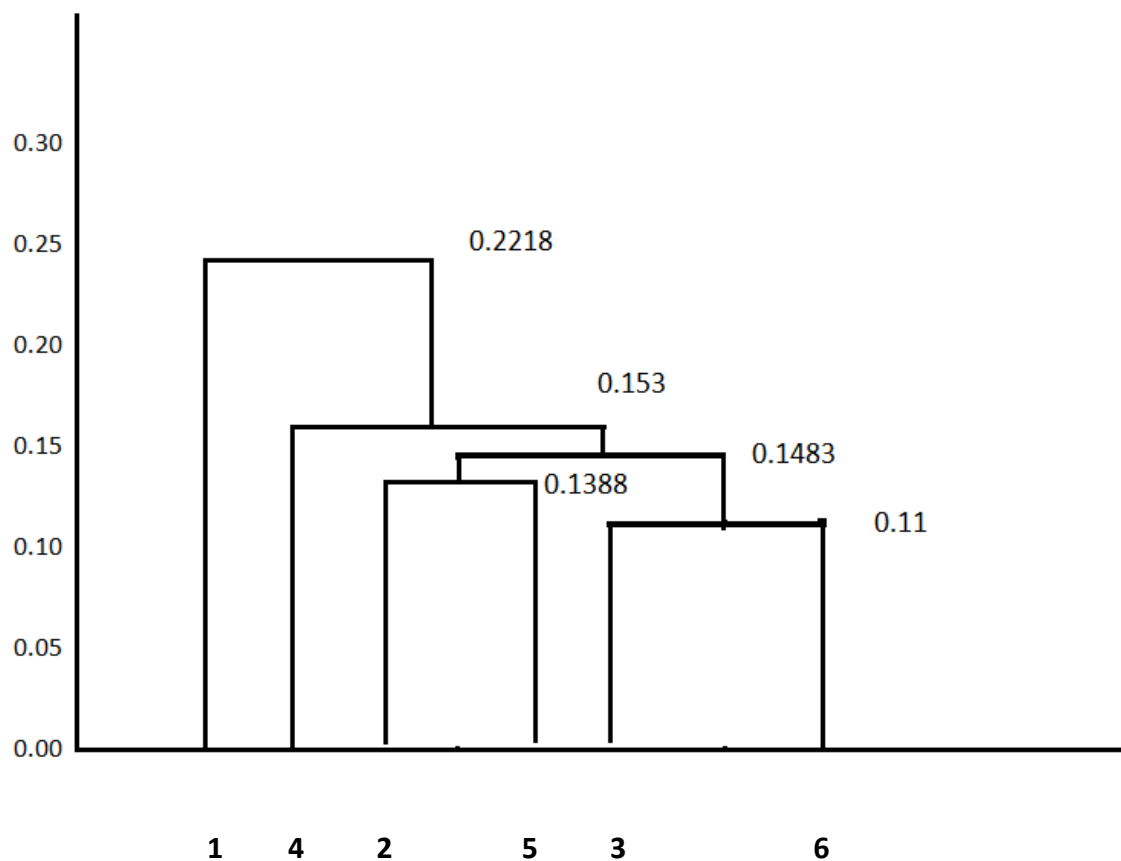
so p25 and p36 forms 3rdcluster

	<b>p1</b>	<b>p(25)(36)</b>	<b>p4</b>
<b>p1</b>	0	0.2218	0.3688
<b>p(25)(36)</b>	0.2218	0	0.1513
<b>p4</b>	0.3688	0.1513	0

Smallest distance is: 0.153

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

	<b>p1</b>	<b>p4(25)(36)</b>
<b>p1</b>	0	0.2218
<b>p4(25)(36)</b>	0.2218	0



### Complete Link Proximity:

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



<b>p36</b>	0.2347	0.254	0	0.2216	0.3921
<b>p4</b>	0.3688	0.2042	0.2216	0	0.2932
<b>p5</b>	0.3421	0.1388	0.3921	0.2932	0

smallest distance from above data is 0.1388

so p2 and p5 forms 2nd cluster

	<b>p1</b>	<b>p25</b>	<b>p36</b>	<b>p4</b>
<b>p1</b>	0	0.3421	0.2347	0.3688
<b>p25</b>	0.3421	0	0.3921	0.2932
<b>p36</b>	0.2347	0.3921	0	0.2216
<b>p4</b>	0.3688	0.2932	0.2216	0

smallest distance from above data is 0.2216

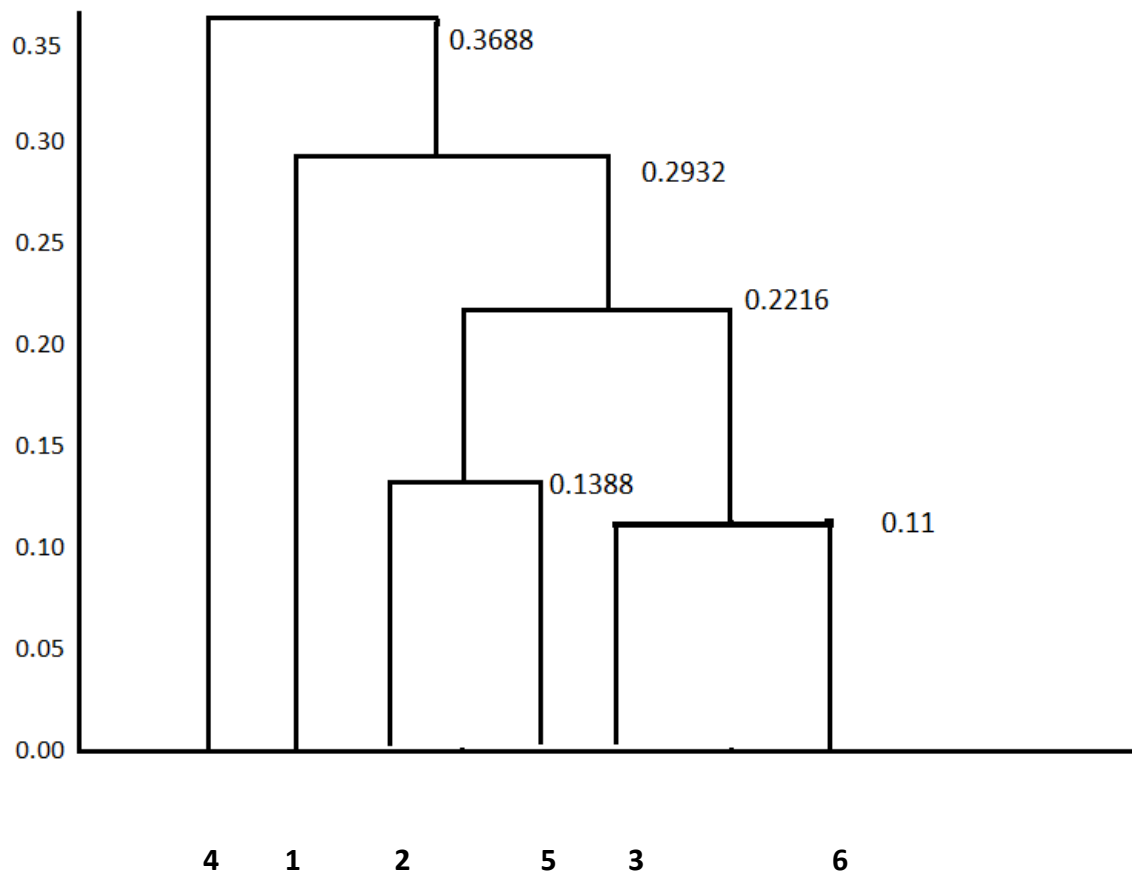
so p25 and p36 forms 3rd cluster

	<b>p1</b>	<b>p(25)(36)</b>	<b>p4</b>
<b>p1</b>	0	0.3421	0.3688
<b>p(25)(36)</b>	0.3421	0	0.2932
<b>p4</b>	0.3688	0.2932	0

smallest distance from above data is 0.2932

so p(25)(36) and p1 forms 4th cluster

	<b>p1(25)(36)</b>	<b>p4</b>
<b>p1(25)(36)</b>	0	0.1483
<b>p4</b>	0.3688	0



Average Link Proximity:

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	0.18645	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	0.269675	0.2487
p36	0.2347	0.269675	0	0.18645
p4	0.3688	0.2487	0.18645	0

smallest distance from above data is 0.18645

so p25 and p36 forms 3rd cluster

	p1	p(25)(36)	p4
p1	0	0.2618	0.3688
p(25)(36)	0.2618	0	0.217575
p4	0.3688	0.217575	0

smallest distance from above data is 0.217575

so p(25)(36) and p1 forms 4th cluster

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

