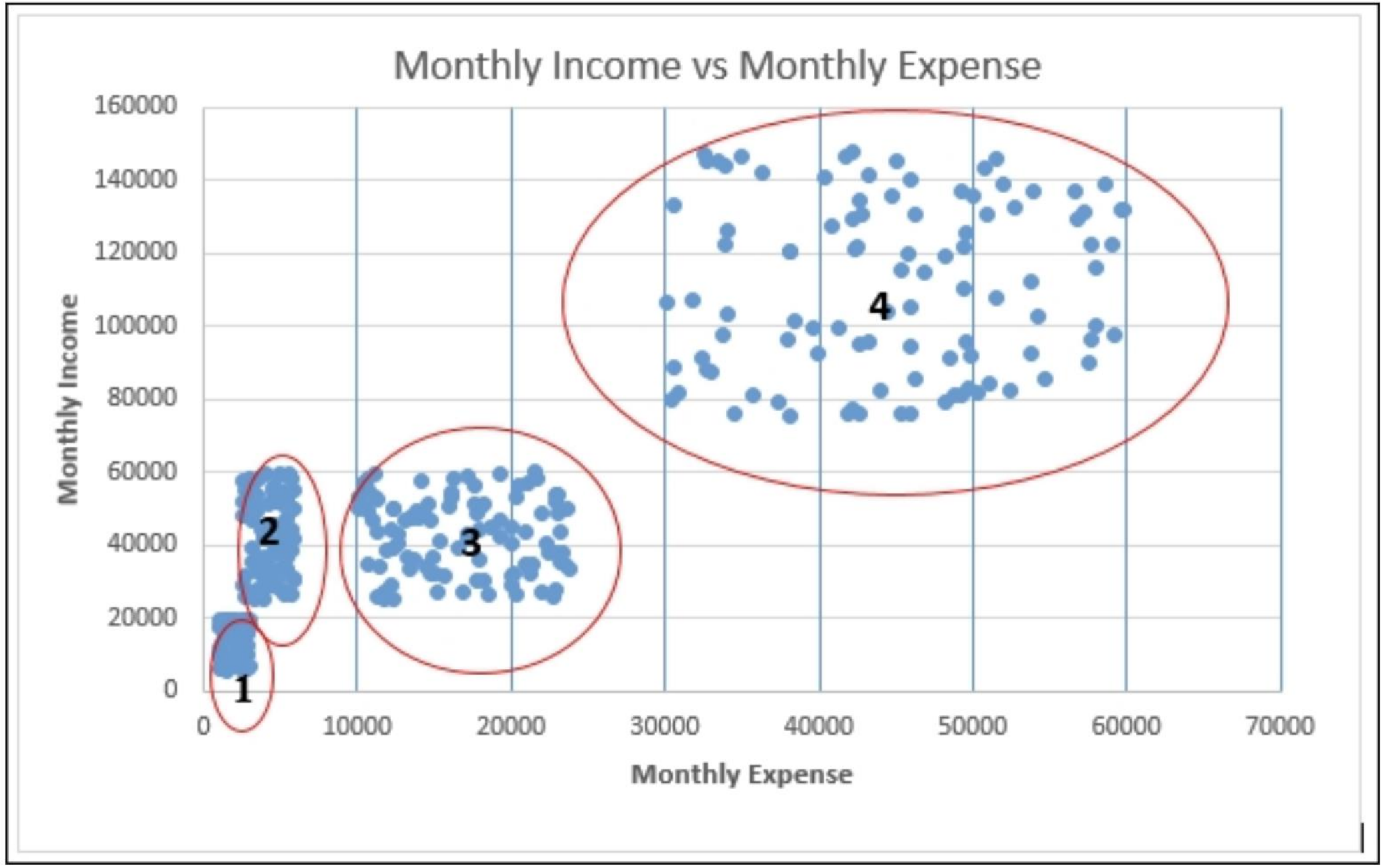
ITP 449

Clustering
K-means

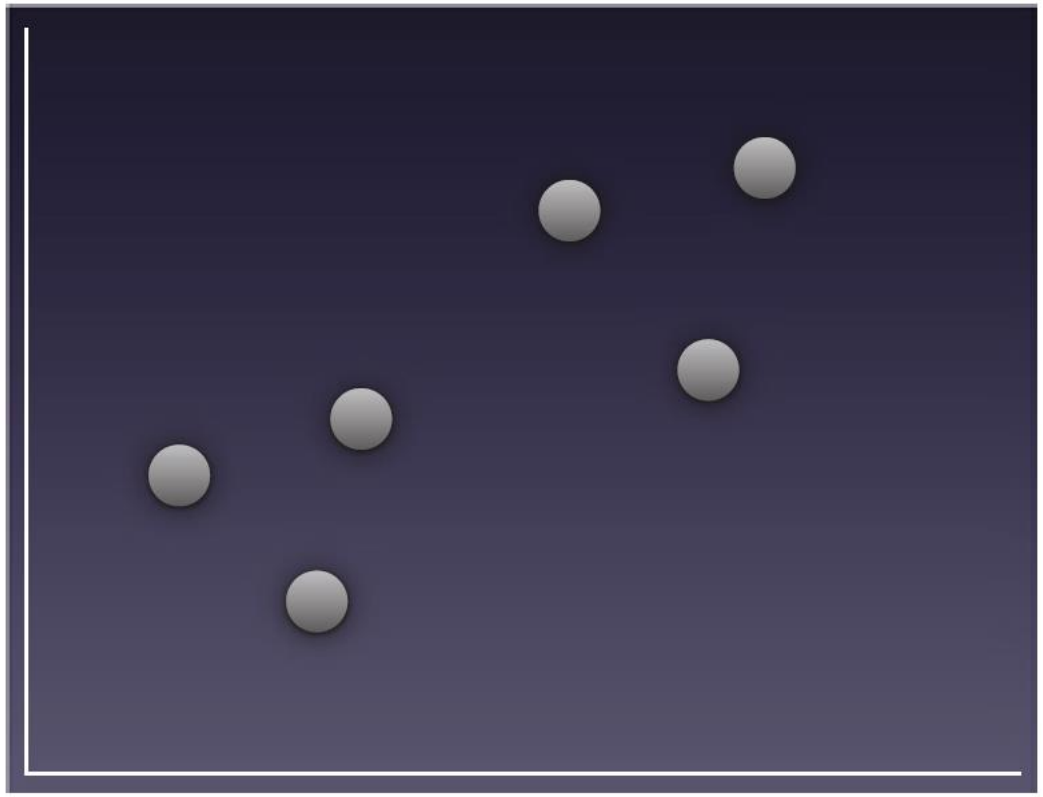
Lecture 10

Clustering:

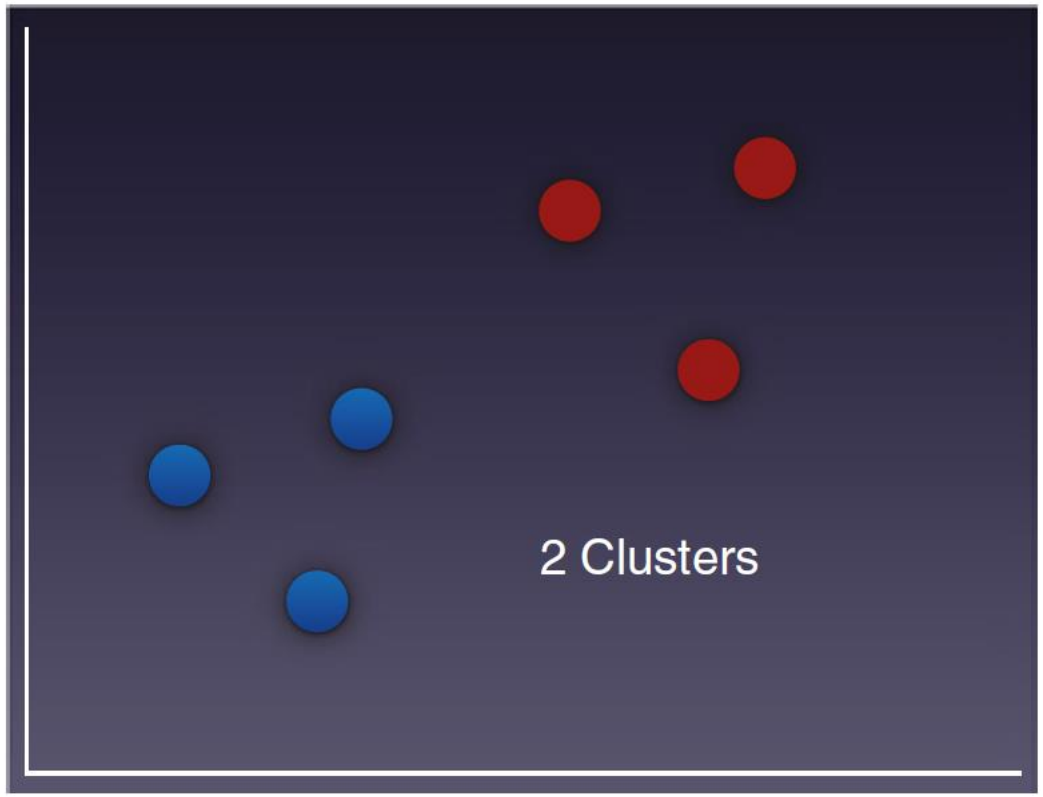
A set of unsupervised algorithms that categorize samples into clusters in which the samples are more similar to each other than the samples outside the cluster.



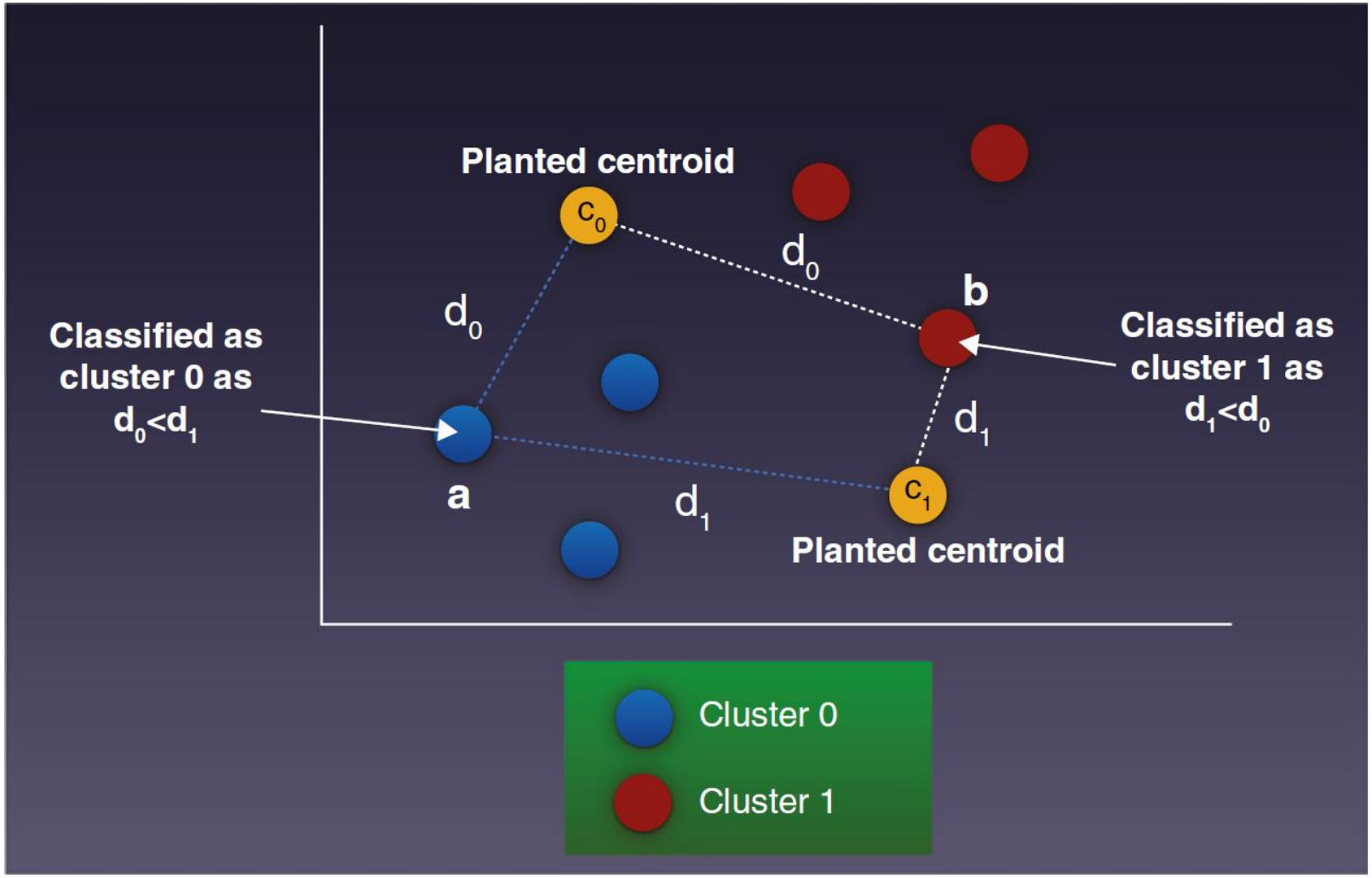
1 - low/low
2 - med/low
3 - med/med
4 - high/high



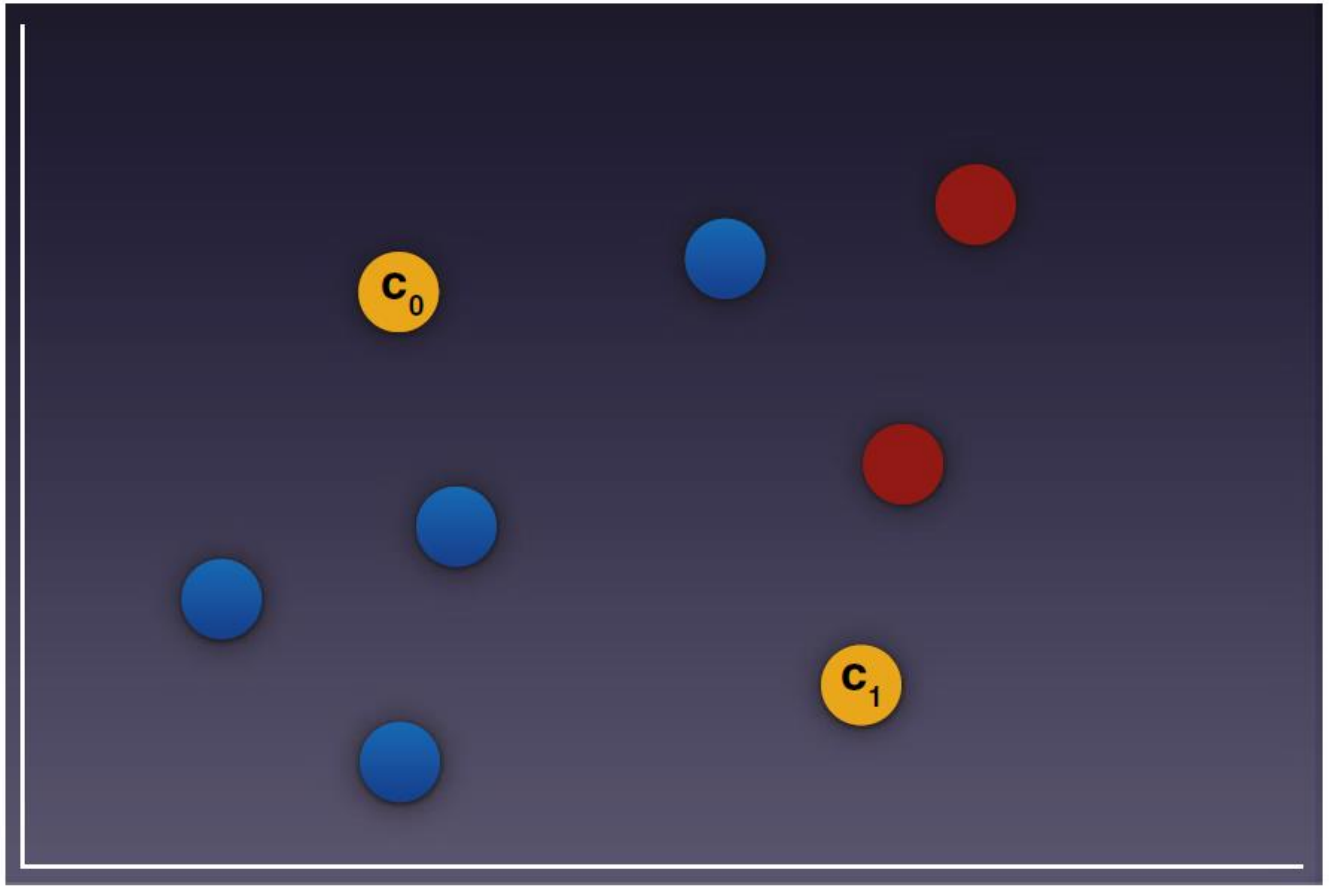
A set of unlabeled data points



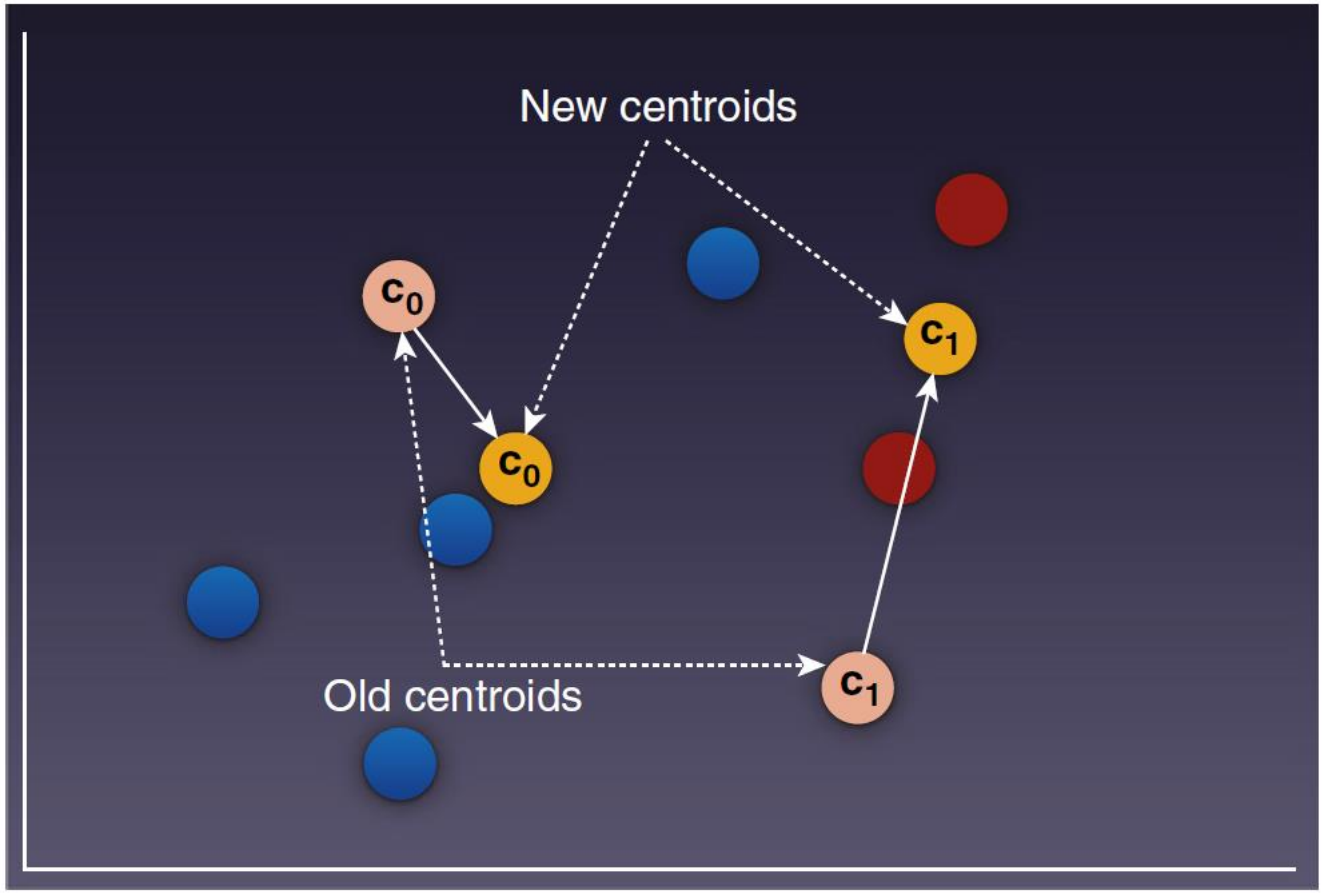
Clustering the points into 2 distinct clusters



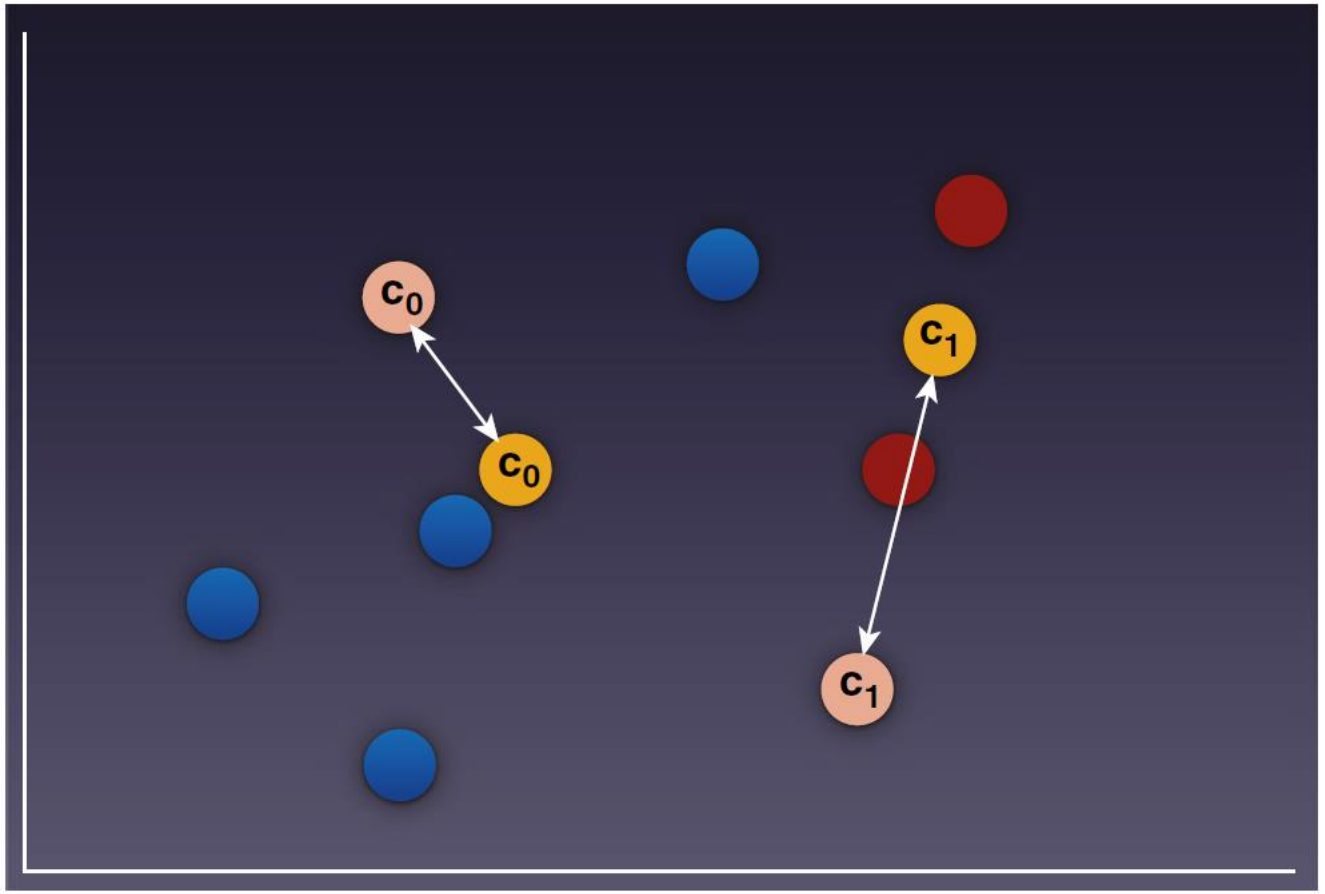
Measuring the distance of each point with respect to each centroid and finding the shortest distance



Groupings of the points after the first round of clustering



Repositioning the centroids by taking the average of all the points in each cluster



Measuring the distance between each centroid; if the distance is 0, the centroid is found

Distance between observations

Student	English	Maths	Science
1	0.12	0.49	0.21
2	0.21	0.81	0.79
3	0.73	0.30	0.99
4	0.55	0.03	0.17
5	0.15	0.83	0.25
6	0.24	0.37	0.63
7	0.20	0.82	0.85
8	0.17	0.92	0.45
9	0.26	0.16	0.31
10	0.15	0.47	0.23

Clustering requires us to measure the similarity between samples:

Consider each sample as a point in n-dimensional space.

Calculate the Euclidean Distance between two points as a measure of similarity:

$$D_{i.j} = \sqrt{\left(X_{i.1} - X_{j.1}\right)^2 + \left(X_{i.2} - X_{j.2}\right)^2 + \ldots\ldots + \left(X_{i.n} - X_{j.n}\right)^2}$$

	1	2	3	4	5	6	7	8	9	10
1	0.00	0.67	1.01	0.63	0.33	0.45	0.72	0.49	0.37	0.04
2	0.67	0.00	0.76	1.06	0.55	0.47	0.06	0.36	0.81	0.66
3	1.01	0.76	0.00	0.88	1.08	0.62	0.76	1.00	0.84	0.97
4	0.63	1.06	0.88	0.00	0.89	0.65	1.10	1.01	0.35	0.60
5	0.33	0.55	1.08	0.89	0.00	0.60	0.60	0.23	0.68	0.35
6	0.45	0.47	0.62	0.65	0.60	0.00	0.50	0.58	0.38	0.41
7	0.72	0.06	0.76	1.10	0.60	0.50	0.00	0.41	0.85	0.71
8	0.49	0.36	1.00	1.01	0.23	0.58	0.41	0.00	0.78	0.50
9	0.37	0.81	0.84	0.35	0.68	0.38	0.85	0.78	0.00	0.34
10	0.04	0.66	0.97	0.60	0.35	0.41	0.71	0.50	0.34	0.00

Normalizing distances

Customer	Monthly Expense	Monthly Income	Education Level
1	4319	28799	5
2	4513	20282	2
3	2959	28743	3
4	4315	28570	3
5	2706	20234	1
6	3794	21981	5
7	2923	24780	4
8	3645	28487	5
9	2561	21092	2
10	4794	22153	5

Customer	Monthly Expense	Monthly Income	Education Level
1	0.79	1.00	1.00
2	0.87	0.01	0.25
3	0.18	0.99	0.50
4	0.79	0.97	0.50
5	0.06	0.00	0.00
6	0.55	0.20	1.00
7	0.16	0.53	0.75
8	0.49	0.96	1.00
9	0.00	0.10	0.25
10	1.00	0.22	1.00

$$Z_i = \left(X_i - X_{min}\right) / \left(X_{max} - X_{min}\right)$$

Linkage methods

The distance between two clusters is computed with the following between a point in cluster 1 and cluster 2:

- **Single** – the minimum distance (closest point between clusters)
- **Complete** – the maximum distance (farthest point between clusters)
- **Average** – the average distance
- **Centroid** – the centroid (mean) of all points

Hierarchical clustering

1. Start with each observation as a cluster so that you have N clusters to start with.
2. Find the smallest distance in the distance matrix. Join (merge) the two observations having the smallest distance to form a cluster.
3. Recompute the distance between all the old clusters and the new clusters.
4. Repeat steps 2 and 3 until all observations fall into a single cluster

Step 1

	1	2	3	4	5	6	7	8	9	10
1	0	0.67	1.01	0.63	0.33	0.45	0.72	0.49	0.37	0.04
2	0.67	0	0.76	1.06	0.55	0.47	0.06	0.36	0.81	0.66
3	1.01	0.76	0	0.88	1.08	0.62	0.76	1	0.84	0.97
4	0.63	1.06	0.88	0	0.89	0.65	1.1	1.01	0.35	0.6
5	0.33	0.55	1.08	0.89	0	0.6	0.6	0.23	0.68	0.35
6	0.45	0.47	0.62	0.65	0.6	0	0.5	0.58	0.38	0.41
7	0.72	0.06	0.76	1.1	0.6	0.5	0	0.41	0.85	0.71
8	0.49	0.36	1	1.01	0.23	0.58	0.41	0	0.78	0.5
9	0.37	0.81	0.84	0.35	0.68	0.38	0.85	0.78	0	0.34
10	0.04	0.66	0.97	0.6	0.35	0.41	0.71	0.5	0.34	0

Step 1

	1|10	2	3	4	5	6	7	8	9
1|10	0	0.66	0.97	0.6	0.33	0.41	0.71	0.49	0.34
2	0.66	0	0.76	1.06	0.55	0.47	0.06	0.36	0.81
3	0.97	0.76	0	0.88	1.08	0.62	0.76	1	0.84
4	0.6	1.06	0.88	0	0.89	0.65	1.1	1.01	0.35
5	0.33	0.55	1.08	0.89	0	0.6	0.6	0.23	0.68
6	0.41	0.47	0.62	0.65	0.6	0	0.5	0.58	0.38
7	0.71	0.06	0.76	1.1	0.6	0.5	0	0.41	0.85
8	0.49	0.36	1	1.01	0.23	0.58	0.41	0	0.78
9	0.34	0.81	0.84	0.35	0.68	0.38	0.85	0.78	0

Step 2

	1|10	2|7	3	4	5	6	8	9
1|10	0	0.66	0.97	0.6	0.33	0.41	0.49	0.34
2|7	0.66	0	0.76	1.06	0.55	0.47	0.36	0.81
3	0.97	0.76	0	0.88	1.08	0.62	1	0.84
4	0.6	1.06	0.88	0	0.89	0.65	1.01	0.35
5	0.33	0.55	1.08	0.89	0	0.6	0.23	0.68
6	0.41	0.47	0.62	0.65	0.6	0	0.58	0.38
8	0.49	0.36	1	1.01	0.23	0.58	0	0.78
9	0.34	0.81	0.84	0.35	0.68	0.38	0.78	0

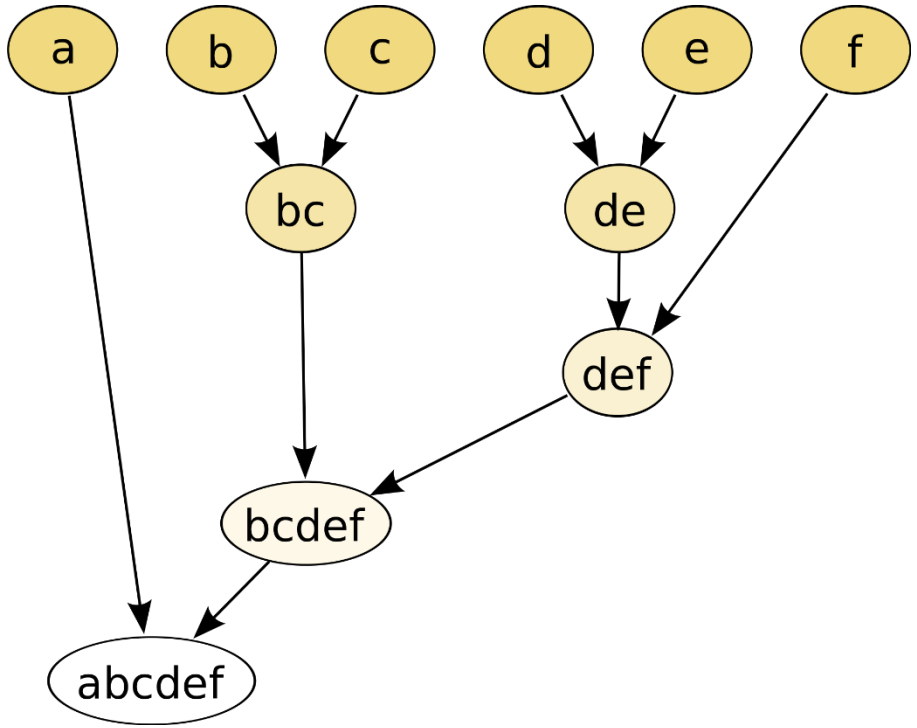
Step 3

	1|10	2|7	5|8	3	4	6	9
1|10	0	0.66	0.33	0.97	0.6	0.41	0.34
2|7	0.66	0	0.36	0.76	1.06	0.47	0.81
5|8	0.33	0.36	0	1	0.89	0.58	0.68
3	0.97	0.76	1	0	0.88	0.62	0.84
4	0.6	1.06	0.89	0.88	0	0.65	0.35
6	0.41	0.47	0.58	0.62	0.65	0	0.38
9	0.34	0.81	0.68	0.84	0.35	0.38	0

Step 4

	1|10|5|8	2|7	3	4	6	9
1|10|5|8	0	0.36	0.97	0.6	0.41	0.34
2|7	0.36	0	0.76	1.06	0.47	0.81
3	0.97	0.76	0	0.88	0.62	0.84
4	0.6	1.06	0.88	0	0.65	0.35
6	0.41	0.47	0.62	0.65	0	0.38
9	0.34	0.81	0.84	0.35	0.38	0

- The result of hierarchical clustering is a tree called ***dendrogram***. By slicing the tree horizontally at any level gives you the number of clusters and cluster members.
- Where to slice the tree is an iterative process to minimize intra-cluster distance and maximizing inter-cluster distance.



Demo: Wine clustering

https://www.kaggle.com/piyushgoyal443/red-wine-dataset

Do the following:

Create a DataFrame variable containing the red wine data CSV file.

```
   fixed.acidity  volatile.acidity  citric.acid  residual.sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free.sulfur.dioxide  total.sulfur.dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65
3                 17.0                  60.0   0.9980  3.16       0.58
4                 11.0                  34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5
(1599, 12)

Process finished with exit code 0
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - slide18.py - PyCharm

ITP449 Master 〉 Lecture 10 〉 slide18.py

slide18

slide18.py

```
1  import pandas as pd
2
3  wineData = pd.read_csv("wineQualityReds.csv")
4
5  pd.set_option("display.max_columns", None)
6
7  # Drop the wine number
8  wineData.drop(wineData.columns[0], axis=1, inplace=True)
9
10 print(wineData.head())
11 print(wineData.shape)
```

Run: wine

```
2                15.0                54.0   0.9970  3.26       0.65
3                17.0                60.0   0.9980  3.16       0.58
4                11.0                34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5
(1599, 12)

Process finished with exit code 0
```

6: TODO 4: Run Terminal Python Console Event Log

8:57 CRLF UTF-8 4 spaces Python 3.8 (venv)

Do the following:

Analyze the quality feature in further detail. Plot the distribution of quality.

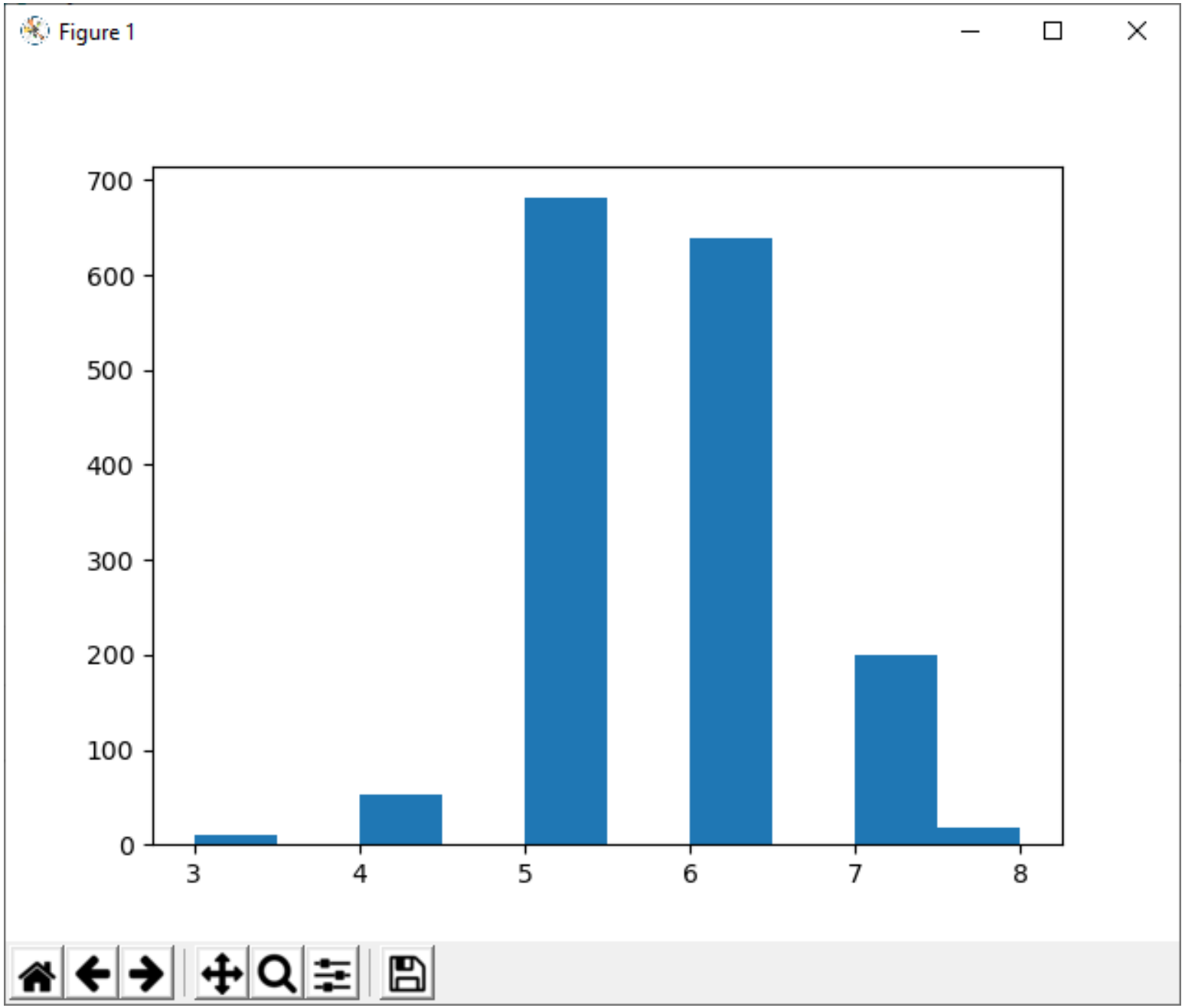


Figure 1
700
600
500
400
300
200
100
0
3
4
5
6
7
8

in_class_coding.py

```python
import pandas as pd
import os
import matplotlib.pyplot as plt

os.chdir('C:/Users/Reza/Desktop/ITP449_Fall2020/Class/Files')
pd.set_option('display.max_columns', None)

wineData = pd.read_csv('wineQualityReds.csv', header=0)

wineData.drop(wineData.columns[[0]], axis=1, inplace=True)

plt.hist(wineData['quality'])
plt.show()
```

13 220

Run: in_class_coding

```
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Cl
```

Do the following:

Determine the mean value for all other features for each level of the quality feature.

quality	fixed.acidity	volatile.acidity	citric.acid	residual.sugar
3	8.360000	0.884500	0.171000	2.635000
4	7.779245	0.693962	0.174151	2.694340
5	8.167254	0.577041	0.243686	2.528855
6	8.347179	0.497484	0.273824	2.477194
7	8.872362	0.403920	0.375176	2.720603
8	8.566667	0.423333	0.391111	2.577778

quality	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density
3	0.122500	11.000000	24.900000	0.997464
4	0.090679	12.264151	36.245283	0.996542
5	0.092736	16.983847	56.513950	0.997104
6	0.084956	15.711599	40.869906	0.996615
7	0.076588	14.045226	35.020101	0.996104
8	0.068444	13.277778	33.444444	0.995212

quality	pH	sulphates	alcohol
3	3.398000	0.570000	9.955000
4	3.381509	0.596415	10.265094
5	3.304949	0.620969	9.899706
6	3.318072	0.675329	10.629519
7	3.290754	0.741256	11.465913
8	3.267222	0.767778	12.094444

- The lesser the volatile acidity and chlorides, the higher the wine quality
- The more the sulphates and citric acid content, the higher the wine quality
- The density and pH don't vary much across the wine quality

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - slide24.py - PyCharm

ITP449 Master 〉 Lecture 10 〉 slide24.py

slide24

slide24.py

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  wineData = pd.read_csv("wineQualityReds.csv")
5
6  pd.set_option("display.max_columns", None)
7
8  # Drop the wine number
9  wineData.drop(wineData.columns[0], axis=1, inplace=True)
10
11 plt.hist(wineData["quality"])
12 plt.show()
13
14 print(wineData.groupby("quality").mean())
```

Run: slide24

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 10/slide2
         fixed.acidity  volatile.acidity  citric.acid  residual.sugar  \
quality
3             8.360000          0.884500     0.171000        2.635000
4             7.779245          0.693962     0.174151        2.694340
5             8.167254          0.577041     0.243686        2.528855
6             8.347179          0.497484     0.273824        2.477194
7             8.872362          0.403920     0.375176        2.720603
```

6: TODO 4: Run Terminal Python Console Event Log

11:30 CRLF UTF-8 4 spaces Python 3.8 (venv)

Do the following:

Normalize the dataset.

	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	\
0	0.247788	0.397260	0.00	0.068493	0.106845	
1	0.283186	0.520548	0.00	0.116438	0.143573	
2	0.283186	0.438356	0.04	0.095890	0.133556	
3	0.584071	0.109589	0.56	0.068493	0.105175	
4	0.247788	0.397260	0.00	0.068493	0.106845	

	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	\
0	0.140845	0.098940	0.567548	0.606299	0.137725	
1	0.338028	0.215548	0.494126	0.362205	0.209581	
2	0.197183	0.169611	0.508811	0.409449	0.191617	
3	0.225352	0.190813	0.582232	0.330709	0.149701	
4	0.140845	0.098940	0.567548	0.606299	0.137725	

	alcohol	quality
0	0.153846	0.4
1	0.215385	0.4
2	0.215385	0.4
3	0.215385	0.6
4	0.153846	0.4

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - slide 27.py - PyCharm

ITP449 Master 〉 Lecture 10 〉 slide 27.py

slide 27

slide 27.py

```
import matplotlib.pyplot as plt

wineData = pd.read_csv("wineQualityReds.csv")

pd.set_option("display.max_columns", None)

# Drop the wine number
wineData.drop(wineData.columns[0], axis=1, inplace=True)

plt.hist(wineData["quality"])
plt.show()

print(wineData.groupby("quality").mean())
wineData_norm = (wineData-wineData.min())/(wineData.max()-wineData.min())
print(wineData_norm.head())
```

Run: slide 27

```
    alcohol  quality
0  0.153846      0.4
1  0.215385      0.4
2  0.215385      0.4
3  0.215385      0.6
4  0.153846      0.4
```

6: TODO 4: Run Terminal Python Console Event Log

5:1 UTF-8 4 spaces Python 3.8 (venv)

Do the following:

Use hierarchical clustering to identify wine groupings based on the available features.

Hint: you may use a method that requires you to specify the number of clusters.

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - slide 27.py - PyCharm

ITP449 Master 〉 Lecture 10 〉 slide 27.py

slide 27

slide 27.py

```
5
6   pd.set_option("display.max_columns", None)
7
8   # Drop the wine number
9   wineData.drop(wineData.columns[0], axis=1, inplace=True)
10
11  plt.hist(wineData["quality"])
12  plt.show()
13
14  print(wineData.groupby("quality").mean())
15  wineData_norm = (wineData-wineData.min())/(wineData.max()-wineData.min())
16  print(wineData_norm.head())
17
18  from sklearn.cluster import AgglomerativeClustering
19  clusters = AgglomerativeClustering(n_clusters=6).fit(wineData_norm)
```

Run: slide 27

```
1   0.215385      0.4
2   0.215385      0.4
3   0.215385      0.6
4   0.153846      0.4

Process finished with exit code 0
```

1: Project 7: Structure 2: Favorites

6: TODO 4: Run Terminal Python Console Event Log

8:23 CRLF UTF-8 4 spaces Python 3.8 (venv)

Do the following:

Plot the distribution of clusters you computed.

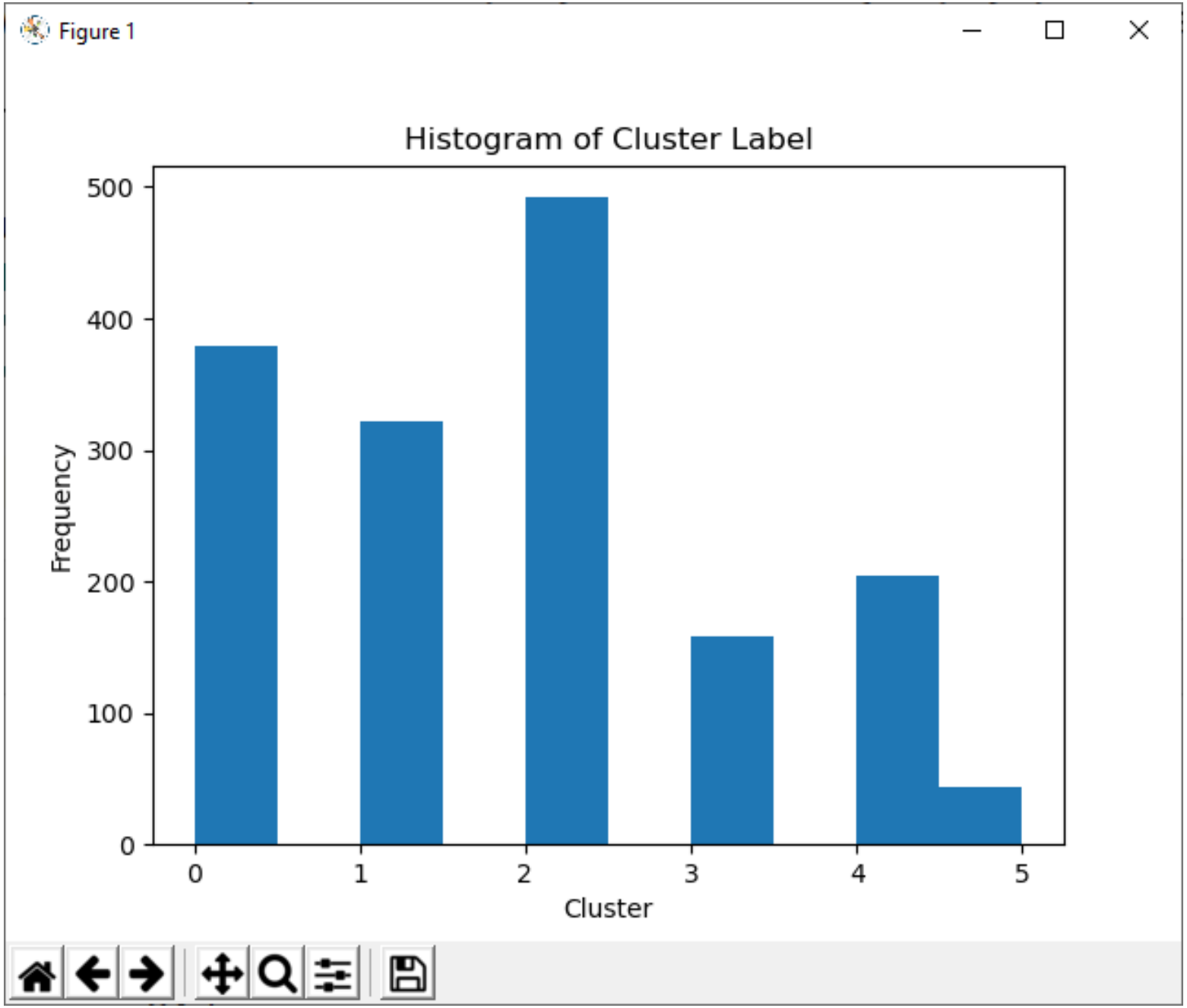


Figure 1
Histogram of Cluster Label
Frequency
500
400
300
200
100
0
0
1
2
3
4
5
Cluster

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - slide 27.py - PyCharm

ITP449 Master 〉 Lecture 10 〉 slide 27.py

slide 27

slide 27.py

```
14  print(wineData.groupby("quality").mean())
15  wineData_norm = (wineData-wineData.min())/(wineData.max()-wineData.min())
16  print(wineData_norm.head())
17
18  from sklearn.cluster import AgglomerativeClustering
19  clusters = AgglomerativeClustering(n_clusters=6).fit(wineData_norm)
20
21  clusterLabels = pd.Series(clusters.labels_)
22
23  plt.hist(clusterLabels)
24  plt.title("Histogram of Cluster Label")
25  plt.xlabel("Cluster")
26  plt.ylabel("Frequency")
27
28  plt.show()
```

Run: slide 27

```
1   0.215385      0.4
2   0.215385      0.4
3   0.215385      0.6
4   0.153846      0.4

Process finished with exit code 0
```

1: Project 7: Structure 2: Favorites

6: TODO 4: Run Terminal Python Console Event Log

23:24 CRLF UTF-8 4 spaces Python 3.8 (venv)

Do the following:

Determine the average values for all the features for each cluster.

cluster	fixed.acidity	volatile.acidity	citric.acid	residual.sugar
0	0.380858	0.171341	0.422691	0.109661
1	0.311109	0.282598	0.299255	0.159683
2	0.265613	0.356714	0.113557	0.091213
3	0.644449	0.206477	0.548734	0.128576
4	0.169313	0.344531	0.080000	0.089394
5	0.335881	0.279577	0.405455	0.069894

cluster	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density
0	0.109928	0.178844	0.098362	0.450974
1	0.126075	0.349926	0.283037	0.545567
2	0.124109	0.147873	0.105073	0.486659
3	0.124564	0.142271	0.098135	0.684780
4	0.095731	0.253521	0.122151	0.329572
5	0.430073	0.172535	0.184388	0.508911

cluster	pH	sulphates	alcohol	quality
0	0.417198	0.226202	0.439821	0.653298
1	0.441899	0.174080	0.203376	0.450932
2	0.485132	0.156005	0.228685	0.455285
3	0.301455	0.226218	0.296754	0.559494
4	0.589432	0.182165	0.482944	0.578431
5	0.274875	0.517148	0.171678	0.450000

The wine quality and taste mainly depends on the quantity of acid, alcohol, and sugar.

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 35.py - PyCharm

ITP449 Master 〉 Lecture 10 〉 Slide 35.py

Slide 35

Slide 35.py

```
17
18  from sklearn.cluster import AgglomerativeClustering
19  clusters = AgglomerativeClustering(n_clusters=6).fit(wineData_norm)
20
21  clusterLabels = pd.Series(clusters.labels_)
22
23  plt.hist(clusterLabels)
24  plt.title("Histogram of Cluster Label")
25  plt.xlabel("Cluster")
26  plt.ylabel("Frequency")
27
28  plt.show()
29
30  wineData_norm["cluster"] = clusterLabels
31  print(wineData_norm.groupby("cluster").mean())
```

Run: Slide 35

```
0          0.417198   0.226202  0.439821  0.653298
1          0.441899   0.174080  0.203376  0.450932
2          0.485132   0.156005  0.228685  0.455285
3          0.301455   0.226218  0.296754  0.559494
4          0.589432   0.182165  0.482944  0.578431
5          0.274875   0.517148  0.171678  0.450000

Process finished with exit code 0
```

6: TODO 4: Run Terminal Python Console Event Log

26:24 UTF-8 4 spaces Python 3.8 (venv)

Do the following:

Display all the wines with their cluster label.

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 35.py - PyCharm

ITP449 Master 〉 Lecture 10 〉 Slide 35.py

Slide 35

Slide 35.py

```
19  clusters = AgglomerativeClustering(n_clusters=6).fit(wineData_norm)
20
21  clusterLabels = pd.Series(clusters.labels_)
22
23  plt.hist(clusterLabels)
24  plt.title("Histogram of Cluster Label")
25  plt.xlabel("Cluster")
26  plt.ylabel("Frequency")
27
28  plt.show()
29
30  wineData_norm["cluster"] = clusterLabels
31  print(wineData_norm.groupby("cluster").mean())
32
33  print(wineData_norm)
```

Run: Slide 35

```
5          0.274875   0.517148  0.171678  0.450000
      fixed.acidity  volatile.acidity  citric.acid  residual.sugar  chlorides  \
0          0.247788          0.397260         0.00        0.068493   0.106845
1          0.283186          0.520548         0.00        0.116438   0.143573
2          0.283186          0.438356         0.04        0.095890   0.133556
3          0.584071          0.109589         0.56        0.068493   0.105175
4          0.247788          0.397260         0.00        0.068493   0.106845
...             ...               ...          ...             ...        ...
```

6: TODO 4: Run Terminal Python Console Event Log

29:1 CRLF UTF-8 4 spaces Python 3.8 (venv)