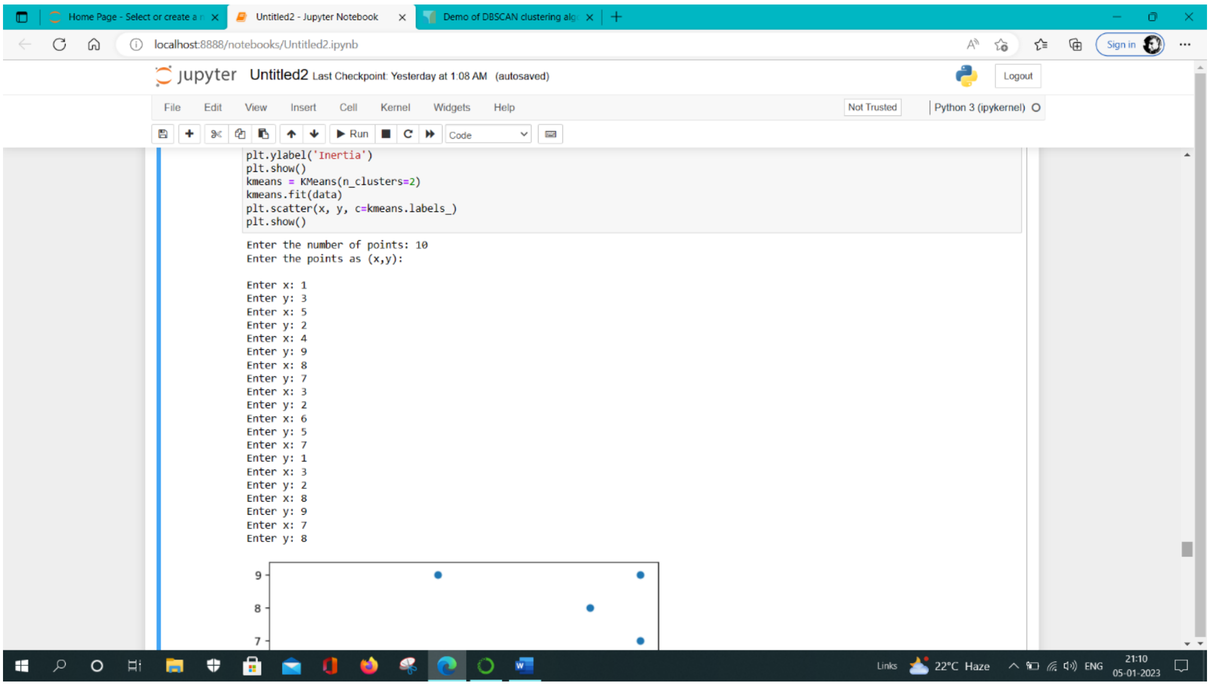
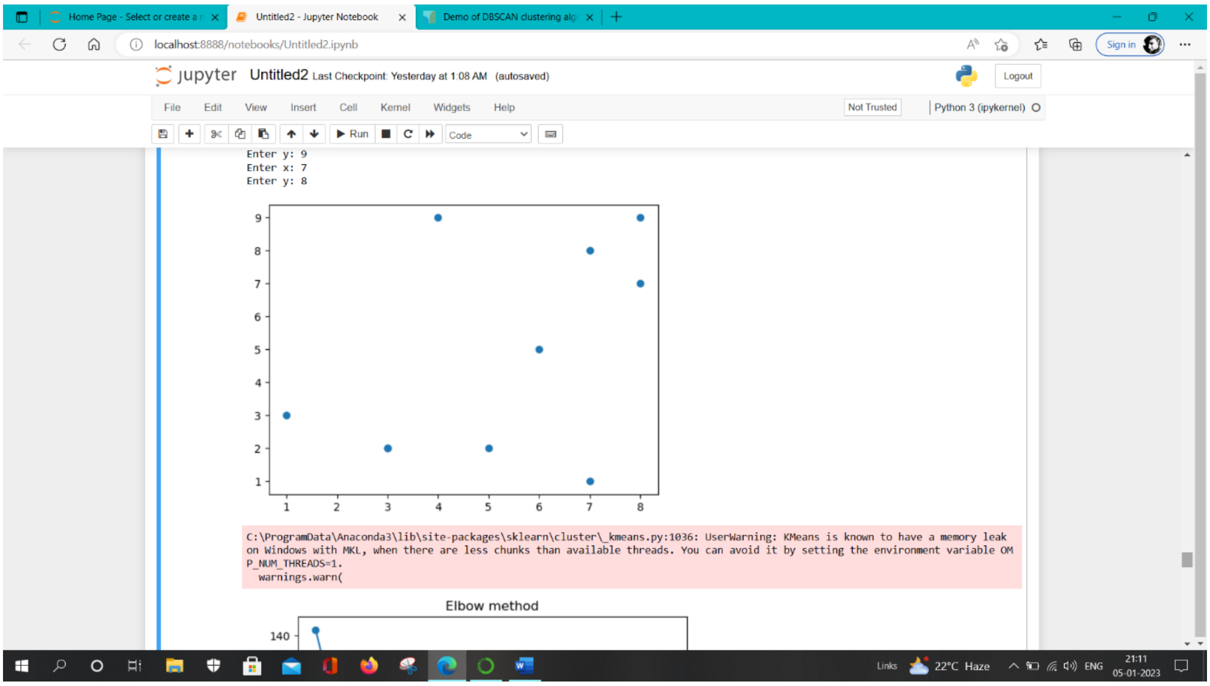
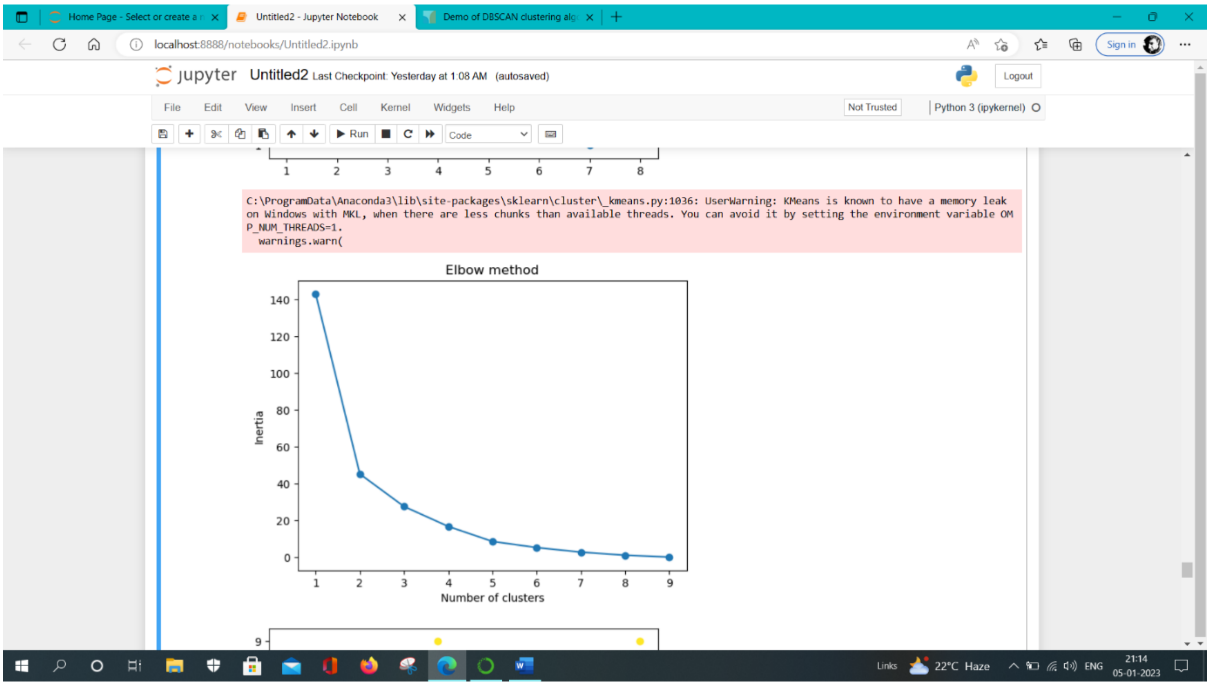
10. Take a few points as input [Hint: use make_blobs from scikit-learn] a. Apply K-means clustering algorithm. i. Vary different values of k. ii. Implement Elbow's method for determining the value of k. b. Apply DBSCAN, Agglomerative clustering algorithm.

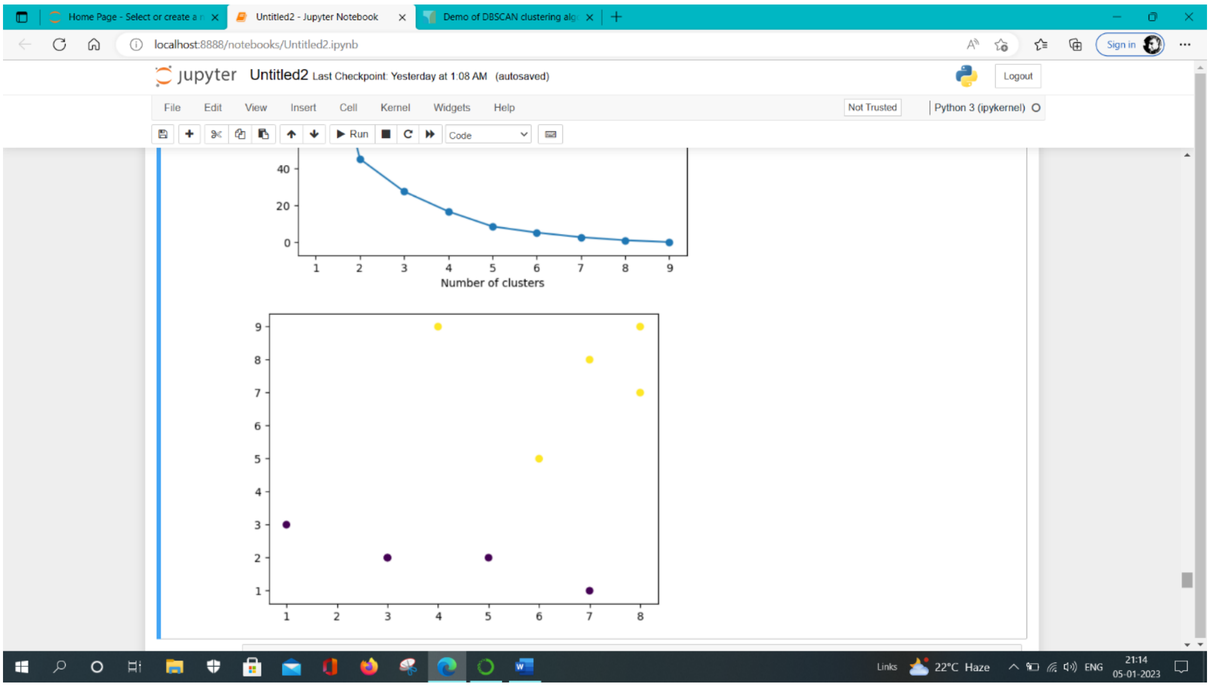
```
import matplotlib.pyplot as plt

x = y = []

j = int(input("Enter the number of points: "))

print("Enter the points as (x,y): \n")

for i in range(0,j):

    a = int(input("Enter x: "))

    x.append(a)

    b = int(input("Enter y: "))

    y.append(b)

plt.scatter(x, y)

plt.show()

from sklearn.cluster import KMeans

data = list(zip(x, y))

inertias = []

for i in range(1,(i+1)):

    kmeans = KMeans(n_clusters=i)

    kmeans.fit(data)

    inertias.append(kmeans.inertia_)

plt.plot(range(1,i+1), inertias, marker='o')

plt.title('Elbow method')

plt.xlabel('Number of clusters')

plt.ylabel('Inertia')

plt.show()

kmeans = KMeans(n_clusters=2)

kmeans.fit(data)

plt.scatter(x, y, c=kmeans.labels_)

plt.show()
```


Home Page - Select or create a
Untitled2 - Jupyter Notebook
Demo of DBSCAN clustering alg
localhost:8888/notebooks/Untitled2.ipynb
jupyter Untitled2 Last Checkpoint: Yesterday at 1:08 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
plt.ylabel('Inertia')
plt.show()
kmeans = KMeans(n_clusters=2)
kmeans.fit(data)
plt.scatter(x, y, c=kmeans.labels_)
plt.show()
Enter the number of points: 10
Enter the points as (x,y):
Enter x: 1
Enter y: 3
Enter x: 5
Enter y: 2
Enter x: 4
Enter y: 9
Enter x: 8
Enter y: 7
Enter x: 3
Enter y: 2
Enter x: 6
Enter y: 5
Enter x: 7
Enter y: 1
Enter x: 3
Enter y: 2
Enter x: 8
Enter y: 9
Enter x: 7
Enter y: 8
9
8
7
Links
22°C Haze
ENG
21:10
05-01-2023


Home Page - Select or create a
Untitled2 - Jupyter Notebook
Demo of DBSCAN clustering alg
localhost:8888/notebooks/Untitled2.ipynb
jupyter Untitled2 Last Checkpoint: Yesterday at 1:08 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Enter y: 9
Enter x: 7
Enter y: 8
9
8
7
6
5
4
3
2
1
1 2 3 4 5 6 7 8
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\cluster_kmeans.py:1036: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
warnings.warn(
Elbow method
140
Links
22°C Haze
ENG
21:11
05-01-2023



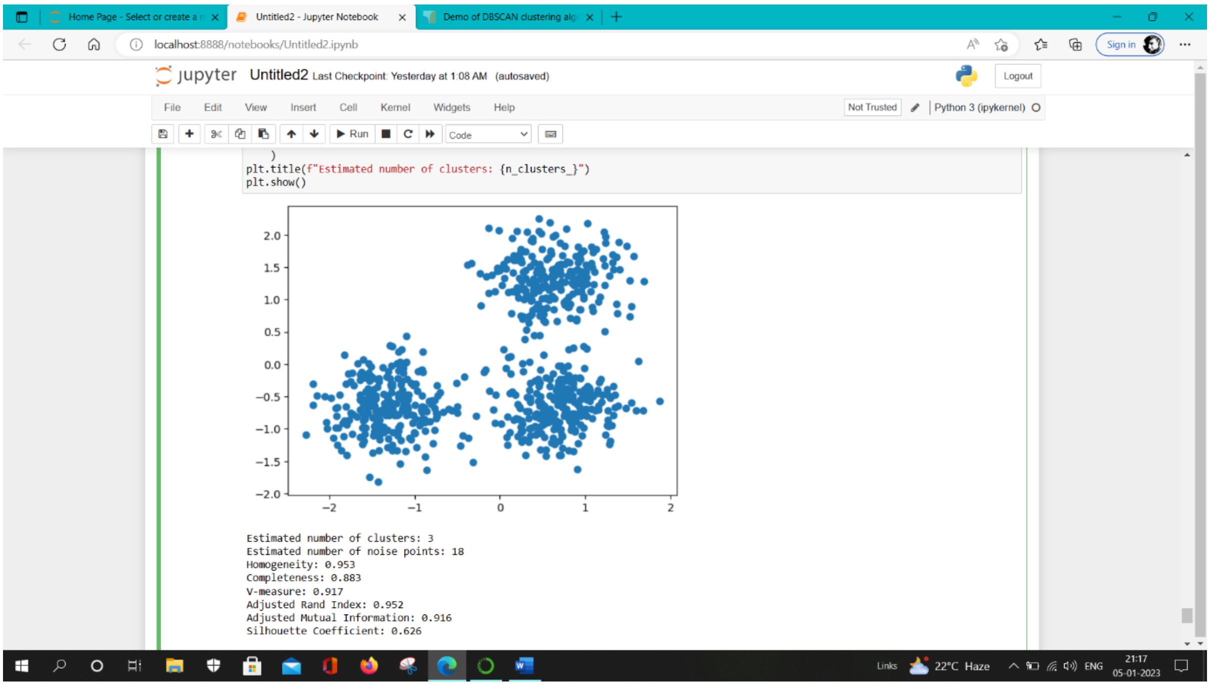
Home Page - Select or create a
Untitled2 - Jupyter Notebook
Demo of DBSCAN clustering alg
localhost:8888/notebooks/Untitled2.ipynb
jupyter Untitled2 Last Checkpoint: Yesterday at 1:08 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\cluster_kmeans.py:1036: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
warnings.warn(
Elbow method
Inertia
Number of clusters
Links
22°C Haze
ENG
21:14
05-01-2023

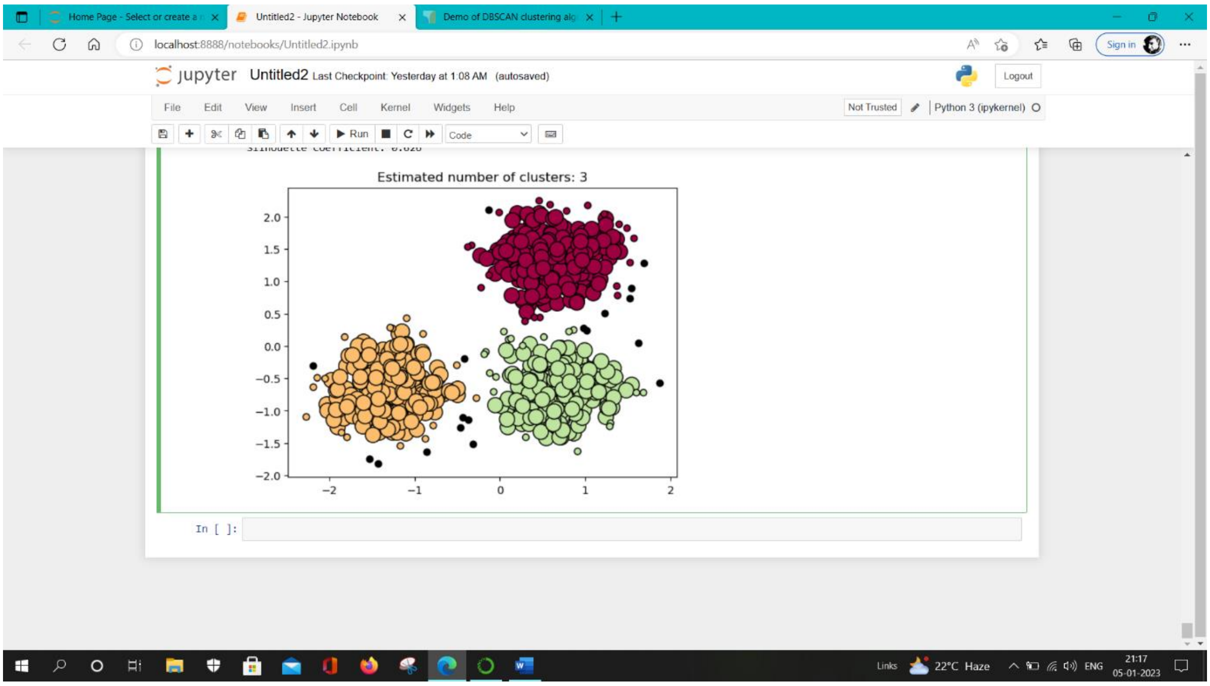


Home Page - Select or create a
Untitled2 - Jupyter Notebook
Demo of DBSCAN clustering alg
localhost:8888/notebooks/Untitled2.ipynb
jupyter Untitled2 Last Checkpoint: Yesterday at 1:08 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Number of clusters
Links
22°C Haze
ENG
21:14
05-01-2023

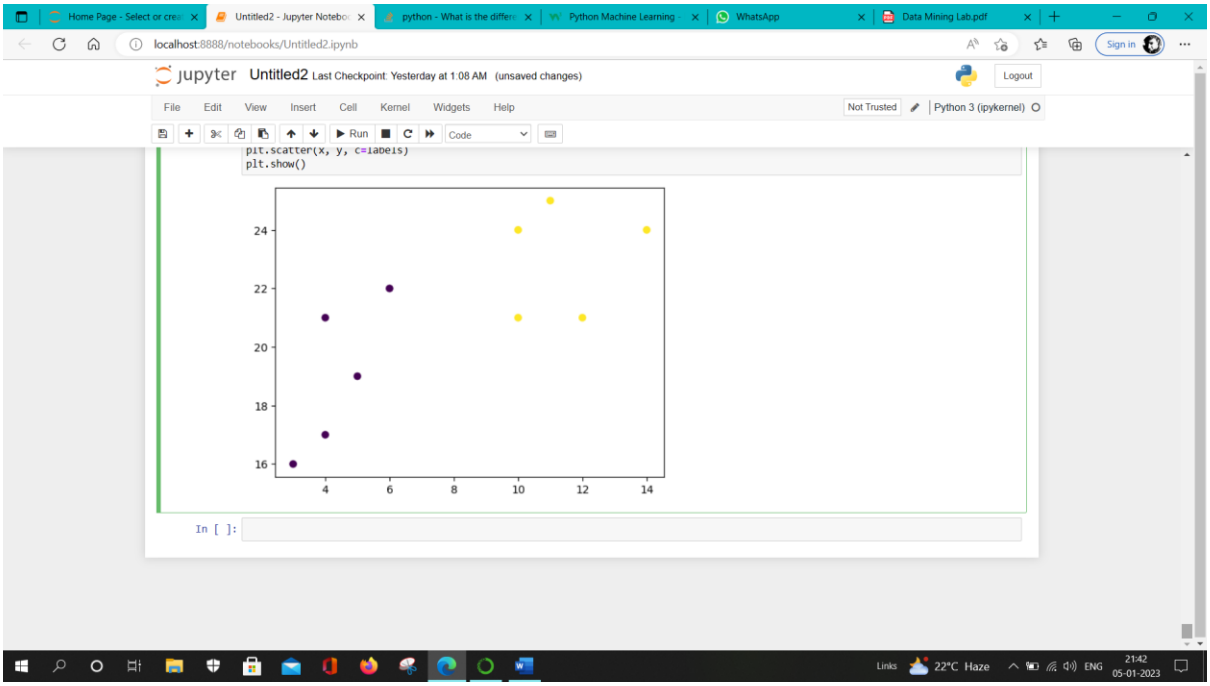
```
#dbscan
from sklearn.datasets import make_blobs
from sklearn.preprocessing import StandardScaler
centers = [[1, 1], [-1, -1], [1, -1]]
X, labels_true = make_blobs(
    n_samples=750, centers=centers, cluster_std=0.4, random_state=0)
X = StandardScaler().fit_transform(X)
import matplotlib.pyplot as plt
plt.scatter(X[:, 0], X[:, 1])
plt.show()
import numpy as np
from sklearn.cluster import DBSCAN
from sklearn import metrics
db = DBSCAN(eps=0.3, min_samples=10).fit(X)
labels = db.labels_
# Number of clusters in labels, ignoring noise if present.
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
n_noise_ = list(labels).count(-1)
print("Estimated number of clusters: %d" % n_clusters_)
print("Estimated number of noise points: %d" % n_noise_)
print(f"Homogeneity: {metrics.homogeneity_score(labels_true, labels):.3f}")
print(f"Completeness: {metrics.completeness_score(labels_true, labels):.3f}")
print(f"V-measure: {metrics.v_measure_score(labels_true, labels):.3f}")
print(f"Adjusted Rand Index: {metrics.adjusted_rand_score(labels_true, labels):.3f}")
print(
    "Adjusted Mutual Information:"
    f" {metrics.adjusted_mutual_info_score(labels_true, labels):.3f}"
)
print(f"Silhouette Coefficient: {metrics.silhouette_score(X, labels):.3f}")
unique_labels = set(labels)
core_samples_mask = np.zeros_like(labels, dtype=bool)
```

```
core_samples_mask[db.core_sample_indices_] = True
colors = [plt.cm.Spectral(each) for each in np.linspace(0, 1, len(unique_labels))]
for k, col in zip(unique_labels, colors):
    if k == -1:
        # Black used for noise.
        col = [0, 0, 0, 1]
    class_member_mask = labels == k
    xy = X[class_member_mask & core_samples_mask]
    plt.plot(
        xy[:, 0],
        xy[:, 1],
        "o",
        markerfacecolor=tuple(col),
        markeredgecolor="k",
        markersize=14,
    )
    xy = X[class_member_mask & ~core_samples_mask]
    plt.plot(
        xy[:, 0],
        xy[:, 1],
        "o",
        markerfacecolor=tuple(col),
        markeredgecolor="k",
        markersize=6,
    )
plt.title(f"Estimated number of clusters: {n_clusters_}")
plt.show()
```


Home Page - Select or create a
Untitled2 - Jupyter Notebook
Demo of DBSCAN clustering alg
localhost:8888/notebooks/Untitled2.ipynb
Sign in
jupyter Untitled2 Last Checkpoint: Yesterday at 1:08 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
)
plt.title(f"Estimated number of clusters: {n_clusters_}")
plt.show()
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2
Estimated number of clusters: 3
Estimated number of noise points: 18
Homogeneity: 0.953
Completeness: 0.883
V-measure: 0.917
Adjusted Rand Index: 0.952
Adjusted Mutual Information: 0.916
Silhouette Coefficient: 0.626
Links
22°C Haze
ENG
21:17
05-01-2023


Home Page - Select or create a
Untitled2 - Jupyter Notebook
Demo of DBSCAN clustering alg
localhost:8888/notebooks/Untitled2.ipynb
Sign in
jupyter Untitled2 Last Checkpoint: Yesterday at 1:08 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Estimated number of clusters: 3
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2
In [ ]:
Links
22°C Haze
ENG
21:17
05-01-2023

```
#agglomerative heirarchial clustering

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import AgglomerativeClustering

x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]

y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]

data = list(zip(x, y))

hierarchical_cluster = AgglomerativeClustering(n_clusters=2, affinity='euclidean', linkage='ward')

labels = hierarchical_cluster.fit_predict(data)

plt.scatter(x, y, c=labels)

plt.show()
```



Home Page - Select or crea
Untitled2 - Jupyter Notebo
python - What is the differe
Python Machine Learning -
WhatsApp
Data Mining Lab.pdf
localhost:8888/notebooks/Untitled2.ipynb
Sign in
jupyter
Untitled2 Last Checkpoint: Yesterday at 1:08 AM (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
plt.scatter(x, y, c=labels)
plt.show()
24
22
20
18
16
4
6
8
10
12
14
In [ ]:
Links
22°C Haze
ENG
21:42
05-01-2023