人工智能概论

实验三：聚类模型

丁烨，计算机科学与技术学院
dingye@dgut.edu.cn

实验环境

https://websitesetup.org/wp-content/uploads/2020/04/Python-Cheat-Sheet.pdf
https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf

实验环境

- scikit-learn
- https://scikit-learn.org/

- 一个开源的科学计算及机器学习工具包
- 属于 SciPy 项目的一部分
- 包含了常见的、基础的机器学习算法
- 不支持深度学习
- 较难支持 GPU 加速

实验环境

- NumPy
- https://www.numpy.org/

- 针对数组运算提供大量的数学函数库
- 支持大规模的多维数组与矩阵运算
- NumPy 是 SciPy、Matplotlib 等扩展程序库的基础组件

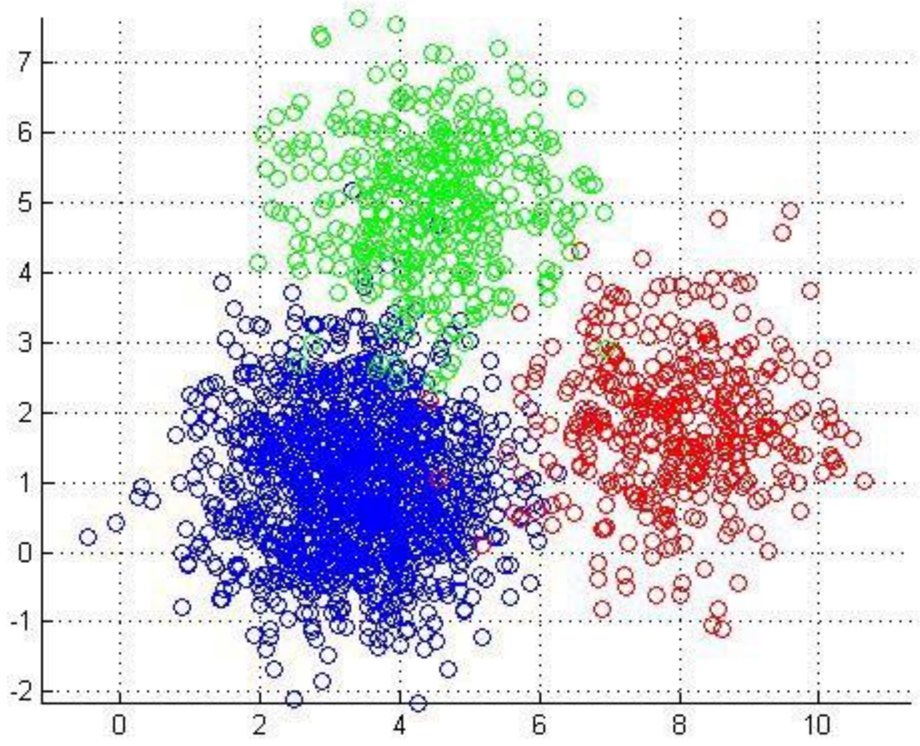
实验环境

- 使用 pip 安装 scikit-learn：
- `pip3 install --user -U scikit-learn`

- 如果安装不成功，可尝试使用 apt 安装：
- `sudo apt install python3-sklearn`

基本概念

- 聚类（Clustering）
- 无监督学习中研究最多、应用最广的任务
- 聚类试图将数据集中的样本划分为若干个通常是不相交的子集
- 每个子集称为一个“簇（Cluster）”



k 均值

- 原型聚类算法
- 给定样本集 $D = \{x_1, x_2, \dots, x_m\}$
- 针对聚类所得簇划分 $C = \{C_1, C_2, \dots, C_k\}$ 最小化平方误差：

$$E = \sum_{i=1}^{k} \sum_{x \in C_j} \|x - \mu_i\|_2^2$$

k 均值

ai.py

```python
from matplotlib import pyplot
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
from sklearn.metrics.cluster import adjusted_mutual_info_score

# Clustering
c = [[1, 1], [-1, -1], [1, -1]]
X, y = make_blobs(n_samples=750, centers=c, cluster_std=0.4, random_state=0)
m = KMeans(n_clusters=len(c))
m.fit(X)
print(adjusted_mutual_info_score(y, m.labels_))
# Plot
fig, axs = pyplot.subplots(2)
axs[0].scatter(X[:, 0], X[:, 1], c=y, alpha=0.5)
axs[1].scatter(X[:, 0], X[:, 1], c=m.labels_, alpha=0.5)
pyplot.show()
```

k 均值

ai.py

```python
from matplotlib import pyplot
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
from sklearn.metrics.cluster import adjusted_mutual_info_score

# Clustering
c = [[1, 1], [-1, -1], [1, -1]]
X, y = make_blobs(n_samples=750, centers=c, cluster_std=0.4, random_state=0)
m = KMeans(n_clusters=len(c))
m.fit(X)
print(adjusted_mutual_info_score(y, m.labels_))
# Plot
fig, axs = pyplot.subplots(2)
axs[0].scatter(X[:, 0], X[:, 1], c=y, alpha=0.5)
axs[1].scatter(X[:, 0], X[:, 1], c=m.labels_, alpha=0.5)
pyplot.show()
```

k 均值

ai.py

```python
from matplotlib import pyplot
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
from sklearn.metrics.cluster import adjusted_mutual_info_score

# Clustering
c = [[1, 1], [-1, -1], [1, -1]]
X, y = make_blobs(n_samples=750 [illegible]=c, cluster_std=0.4, random_state=0)
m = KMeans(n_clusters=len(c))
m.fit(X)
print(adjusted_mutual_info_score(y, m.labels_))
# Plot
fig, axs = pyplot.subplots(2)
axs[0].scatter(X[:, 0], X[:, 1], c=y, alpha=0.5)
axs[1].scatter(X[:, 0], X[:, 1], c=m.labels_, alpha=0.5)
pyplot.show()

```

k 均值

ai.py

```python
from matplotlib import pyplot
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
from sklearn.metrics.cluster import adjusted_mutual_info_score

# Clustering
c = [[1, 1], [-1, -1], [1, -1]]
X, y = make_blobs(n_samples=750, centers=c, cluster_std=0.4, random_state=0)
m = KMeans(n_clusters=len(c))
m.fit(X)
print(adjusted_mutual_info_score(y, m.labels_))
# Plot
fig, axs = pyplot.subplots(2)
axs[0].scatter(X[:, 0], X[:, 1], c=y, alpha=0.5)
axs[1].scatter(X[:, 0], X[:, 1], c=m.labels_, alpha=0.5)
pyplot.show()
```

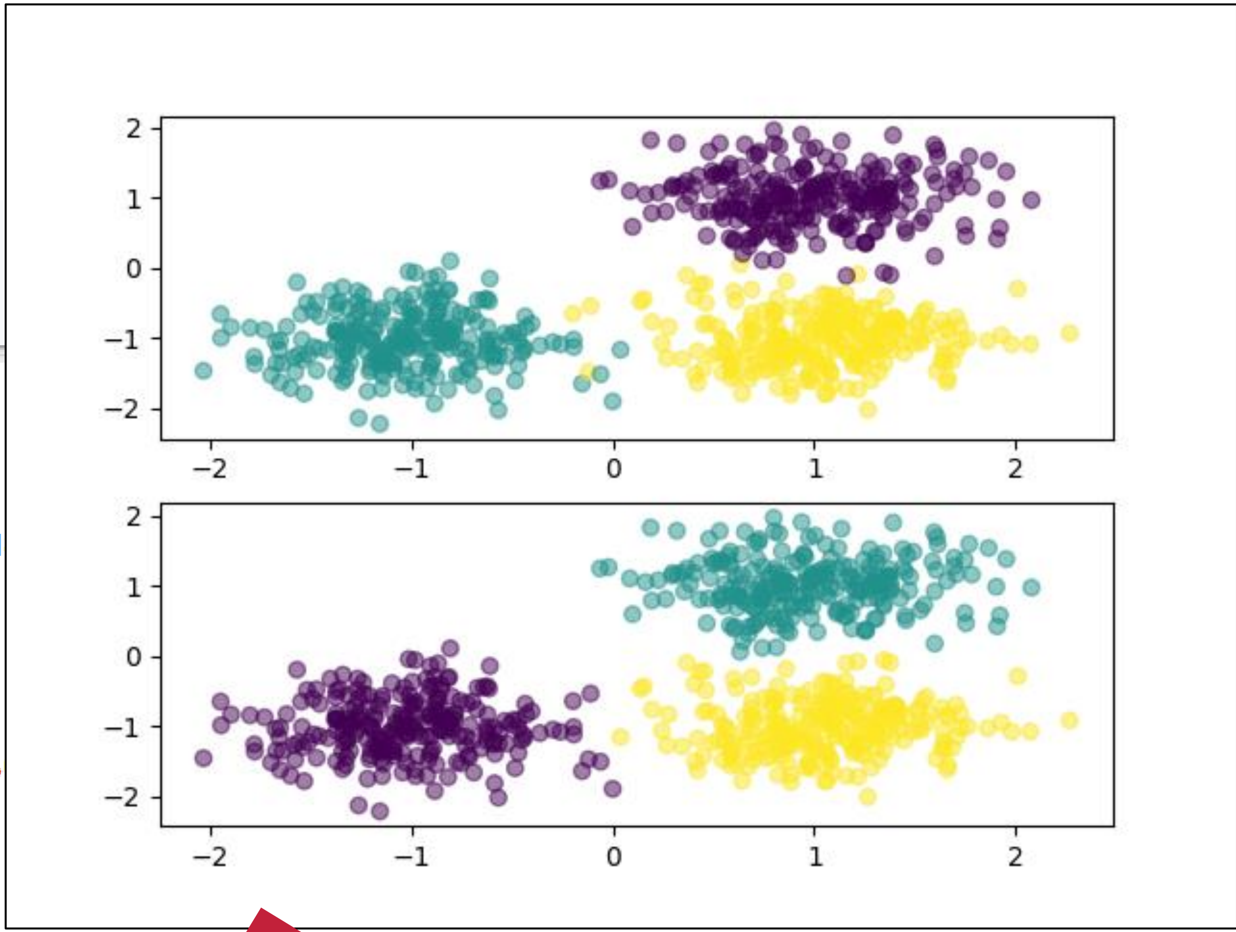
```
~/Downloads/Clustering » python3 ai.py
0.9445999199156632
```

k 均值

ai.py

```python
from matplotlib import pyplot
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
from sklearn.metrics.cluster import adjusted_mutu

# Clustering
c = [[1, 1], [-1, -1], [1, -1]]
X, y = make_blobs(n_samples=750, centers=c, clust
m = KMeans(n_clusters=len(c))
m.fit(X)
print(adjusted_mutual_info_score(y, m.labels_))
# Plot
fig, axs = pyplot.subplots(2)
axs[0].scatter(X[:, 0], X[:, 1], c=y, alpha=0.5)
axs[1].scatter(X[:, 0], X[:, 1], c=m.labels_, alpha=0.5)
pyplot.show()
```

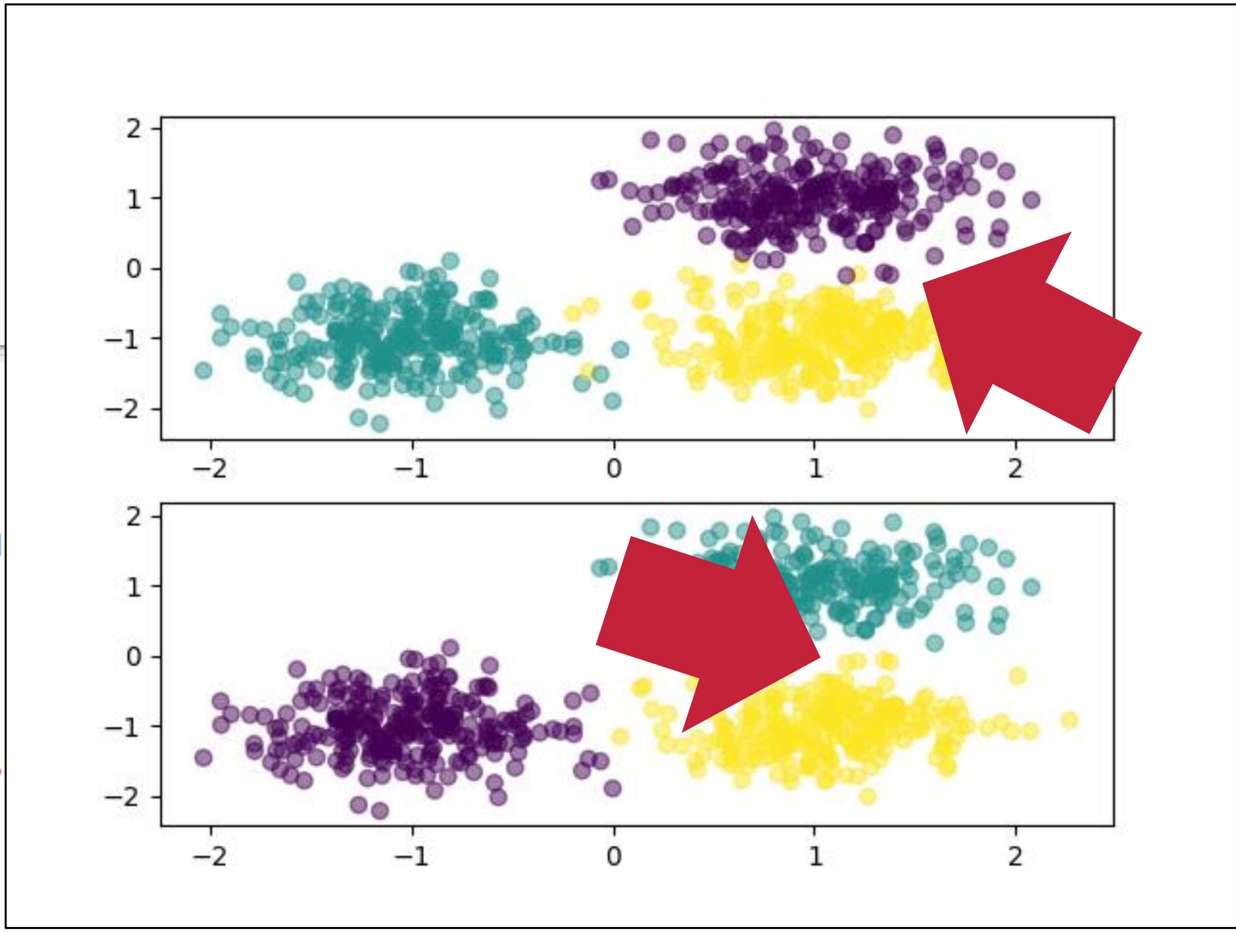


k 均值

ai.py

```python
from matplotlib import pyplot
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
from sklearn.metrics.cluster import adjusted_mutu

# Clustering
c = [[1, 1], [-1, -1], [1, -1]]
X, y = make_blobs(n_samples=750, centers=c, clust
m = KMeans(n_clusters=len(c))
m.fit(X)
print(adjusted_mutual_info_score(y, m.labels_))
# Plot
fig, axs = pyplot.subplots(2)
axs[0].scatter(X[:, 0], X[:, 1], c=y, alpha=0.5)
axs[1].scatter(X[:, 0], X[:, 1], c=m.labels_, alpha=0.5)
pyplot.show()
```



DBSCAN

- 密度聚类算法
- 它基于一组“邻域（Neighborhood）”参数 $(\epsilon, MinPts)$
- 来刻画样本分布的紧密程度

DBSCAN

ai.py

```python
from matplotlib import pyplot
from sklearn.cluster import DBSCAN
from sklearn.datasets import make_blobs
from sklearn.metrics.cluster import adjusted_mutual_info_score

# Clustering
c = [[1, 1], [-1, -1], [1, -1]]
X, y = make_blobs(n_sa[illegible]0, centers=c, cluster_std=0.4, random_state=0)
m = DBSCAN(eps=0.2)
m.fit(X)
print(adjusted_mutual_info_score(y, m.labels_))
# Plot
fig, axs = pyplot.subplots(2)
axs[0].scatter(X[:, 0], X[:, 1], c=y, alpha=0.5)
axs[1].scatter(X[:, 0], X[:, 1], c=m.labels_, alpha=0.5)
pyplot.show()
```

DBSCAN

ai.py

```python
from matplotlib import pyplot
from sklearn.cluster import DBSCAN
from sklearn.datasets import make_blobs
from sklearn.metrics.cluster import adjusted_mutual_info_score

# Clustering
c = [[1, 1], [-1, -1], [1, -1]]
X, y = make_blobs(n_samples=750, centers=c, cluster_std=0.4, random_state=0)
m = DBSCAN(eps=0.2)
m.fit(X)
print(adjusted_mutual_info_score(y, m.labels_))
# Plot
fig, axs = pyplot.subplots(2)
axs[0].scatter(X[:, 0], X[:, 1], c=y, alpha=0.5)
axs[1].scatter(X[:, 0], X[:, 1], c=m.labels_, alpha=0.5)
pyplot.show()
```

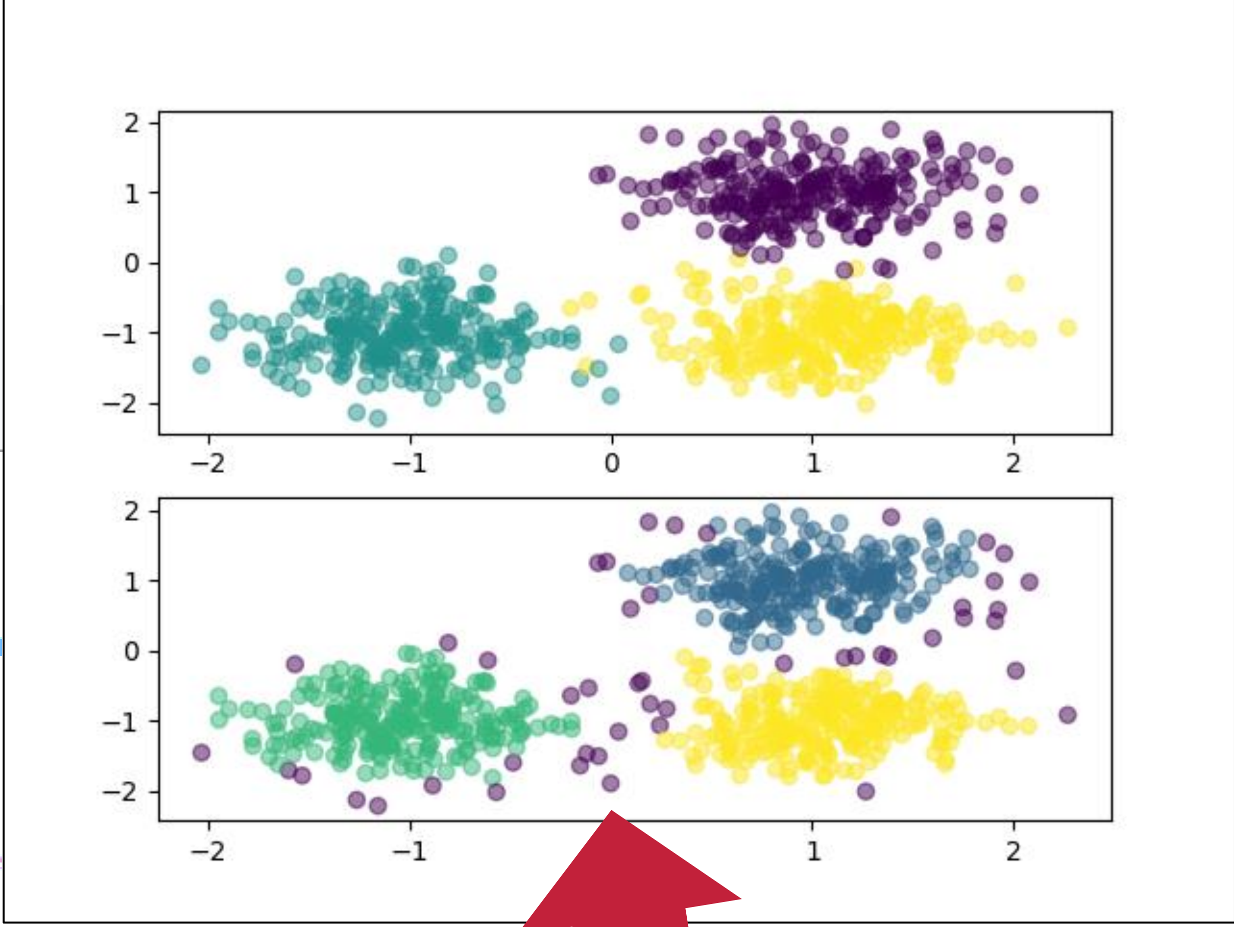
```
------------------------------------------------
~/Downloads/Clustering » python3 ai.py
0.8628736565341091
------------------------------------------------
```

DBSCAN

ai.py

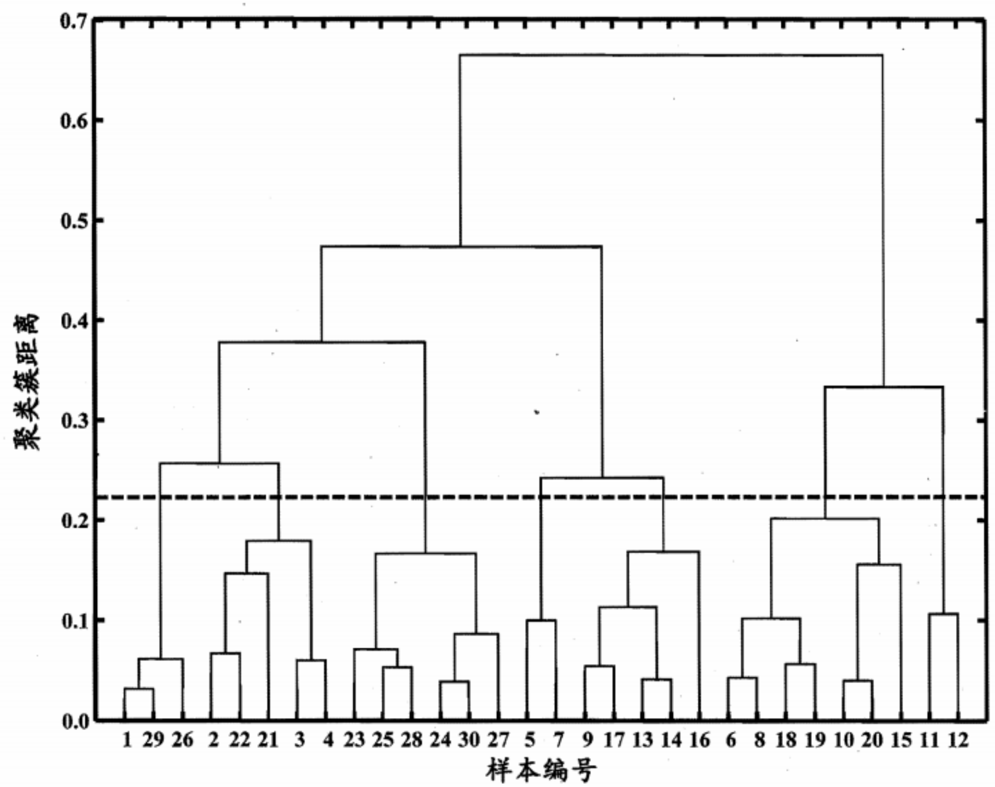
```python
from matplotlib import pyplot
from sklearn.cluster import DBSCAN
from sklearn.datasets import make_blobs
from sklearn.metrics.cluster import adjusted_mutua

# Clustering
c = [[1, 1], [-1, -1], [1, -1]]
X, y = make_blobs(n_samples=750, centers=c, cluste
m = DBSCAN(eps=0.2)
m.fit(X)
print(adjusted_mutual_info_score(y, m.labels_))
# Plot
fig, axs = pyplot.subplots(2)
axs[0].scatter(X[:, 0], X[:, 1], c=y, alpha=0.5)
axs[1].scatter(X[:, 0], X[:, 1], c=m.labels_, alpha=0.5)
pyplot.show()
```



AGNES

- ❖ AGNES（Agglomerative Nesting）
- ❖ 一种采用自底向上聚合策略的层次聚类算法
- ❖ 它先将数据集中的每个样本看作一个初始聚类簇
- ❖ 然后在算法运行的每一步中找出距离最近的两个聚类簇进行合并
- ❖ 该过程不断重复，直至达到预设的聚类簇个数



AGNES

ai.py

```python
from matplotlib import pyplot
from sklearn.cluster import AgglomerativeClustering
from sklearn.datasets import make_blobs
from sklearn.metrics.cluster import adjusted_mutual_info_score

# Clustering
c = [[1, 1], [-1, -1], [1, -1]]
X, y = make_blobs(n_samples=750, centers=c, clust      .4, random_state=0)
m = AgglomerativeClustering(n_clusters=len(c))
m.fit(X)
print(adjusted_mutual_info_score(y, m.labels_))
# Plot
fig, axs = pyplot.subplots(2)
axs[0].scatter(X[:, 0], X[:, 1], c=y, alpha=0.5)
axs[1].scatter(X[:, 0], X[:, 1], c=m.labels_, alpha=0.5)
pyplot.show()
```

AGNES

ai.py

```python
from matplotlib import pyplot
from sklearn.cluster import AgglomerativeClustering
from sklearn.datasets import make_blobs
from sklearn.metrics.cluster import adjusted_mutual_info_score

# Clustering
c = [[1, 1], [-1, -1], [1, -1]]
X, y = make_blobs(n_samples=750, centers=c, cluster_std=0.4, random_state=0)
m = AgglomerativeClustering(n_clusters=len(c))
m.fit(X)
print(adjusted_mutual_info_score(y, m.labels_))
# Plot
fig, axs = pyplot.subplots(2)
axs[0].scatter(X[:, 0], X[:, 1], c=y, alpha=0.5)
axs[1].scatter(X[:, 0], X[:, 1], c=m.labels_, alpha=0.5)
pyplot.show()
```

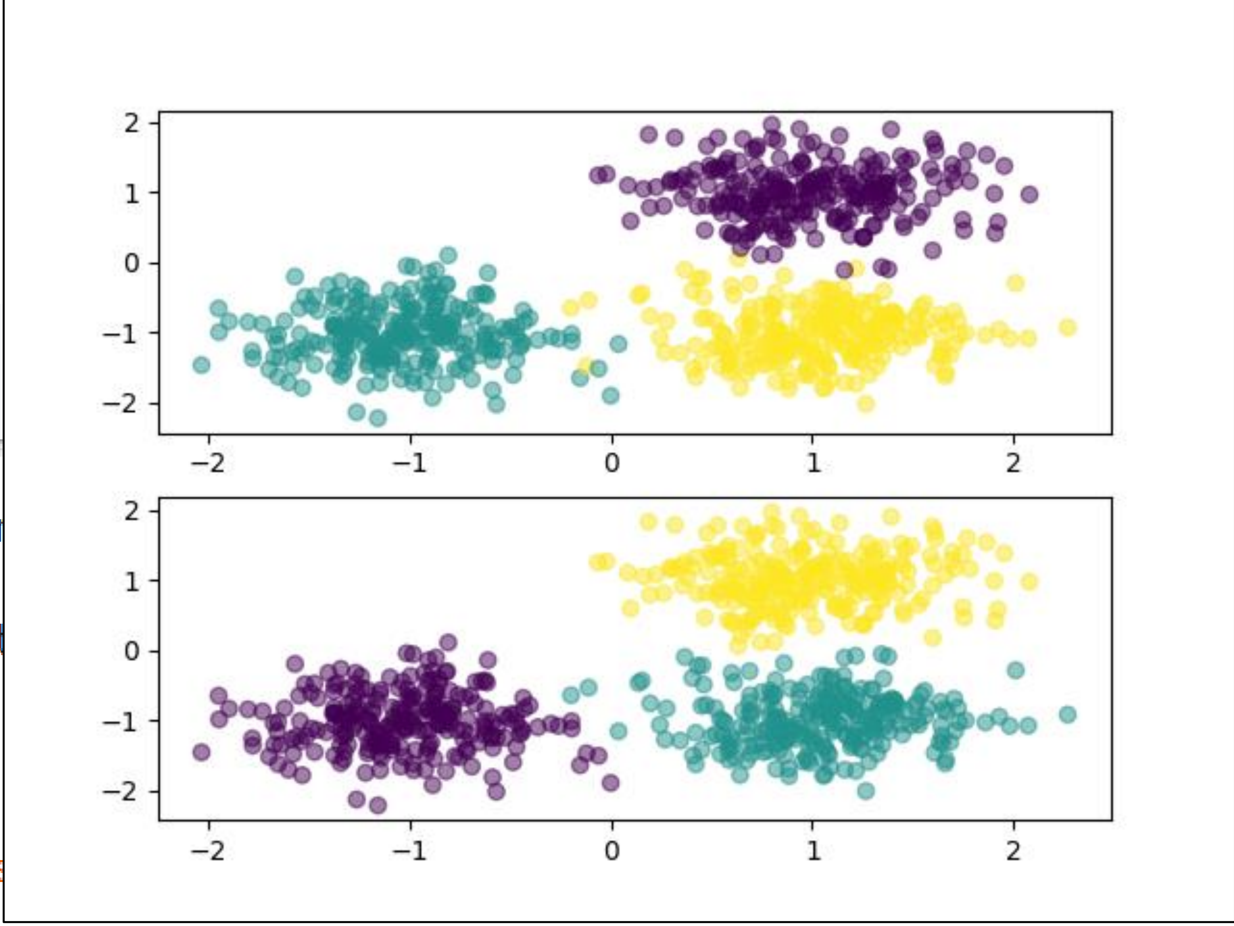
```
~/Downloads/Clustering » python3 ai.py
0.956429112266449
```

AGNES

ai.py

```python
from matplotlib import pyplot
from sklearn.cluster import AgglomerativeCluster
from sklearn.datasets import make_blobs
from sklearn.metrics.cluster import adjusted_mut

# Clustering
c = [[1, 1], [-1, -1], [1, -1]]
X, y = make_blobs(n_samples=750, centers=c, clus
m = AgglomerativeClustering(n_clusters=len(c))
m.fit(X)
print(adjusted_mutual_info_score(y, m.labels_))
# Plot
fig, axs = pyplot.subplots(2)
axs[0].scatter(X[:, 0], X[:, 1], c=y, alpha=0.5)
axs[1].scatter(X[:, 0], X[:, 1], c=m.labels_, alpha=0.5)
pyplot.show()
```



作业

- 精灵宝可梦（ポケットモンスタ、Pokémon）
- https://www.pokemon.co.jp/
- 一个跨媒体制作的作品系列
- 包括游戏、动画、漫画、卡片游戏及相关产品
- 游戏允许玩家捕获，收集，培育数百只“宝可梦”
- 通过与其他宝可梦对战
- 宝可梦能够提升等级甚至进化，成为更强大的宝可梦

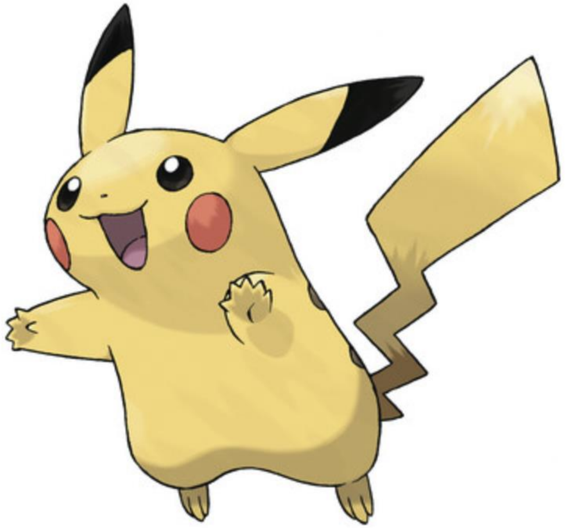
作业

- 宝可梦数据库
- https://pokemondb.net/

#006 Charizard	#248 Tyranitar
#025 Pikachu	#445 Garchomp
#094 Gengar	#448 Lucario
#130 Gyarados	#823 Corviknight
#133 Eevee	#849 Toxtricity
#149 Dragonite	#887 Dragapult

NORMAL	FIRE	WATER	ELECTRIC	GRASS
ICE	FIGHTING	POISON	GROUND	FLYING
PSYCHIC	BUG	ROCK	GHOST	DRAGON
	DARK	STEEL	FAIRY	

Pikachu | Partner Pikachu



Pokédex data

National №	025
Type	ELECTRIC
Species	Mouse Pokémon
Height	0.4 m (1′04″)
Weight	6.0 kg (13.2 lbs)
Abilities	1. Static Lightning Rod (hidden ability)
	025 (Yellow/Red/Blue) 022 (Gold/Silver/Crystal) 156 (Ruby/Sapphire/Emerald)

Base stats

HP	35	180	274
Attack	55	103	229
Defense	40	76	196
Sp. Atk	50	94	218
Sp. Def	50	94	218
Speed	90	166	306
Total	**320**	Min	Max

The ranges shown on the right are for a level 100 Pokémon. Maximum values are based on a beneficial nature, 252 EVs, 31 IVs; minimum values are based on a hindering nature, 0 EVs, 0 IVs.

作业

- 宝可梦数据库
- https://unicorn.org.cn/valency/src/pokemon-v0.5.27.csv

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False

作业

- ❖ 获取宝可梦数据
- ❖ 对 Type 1 / Type 2 / Legendary 进行聚类分析
- ❖ 使用 AMI（Adjusted Mutual Information）评估模型效果

- ❖ 提供完整的代码
- ❖ 提供完整的实验结果截图
- ❖ 尝试找到最好的模型，成绩与模型效果线性相关

作业

- 对 Type 1 / Type 2 / Legendary 进行聚类分析
- 使用 AMI（Adjusted Mutual Information）评估模型效果

```
------------------------------------------------------------
~/Downloads/Clustering » python3 demo.py
0.027631460747593192
0.011719382956574044
0.2207350772027686
------------------------------------------------------------
```

作业

- 在作业系统中下载并完成本实验课对应实验报告
- https://hw.dgut.edu.cn/
- 注意：所有标识为 * 的地方都需要填写
- 截止日期：2024-05-27 23:59:59

课程名称：人工智能概论

实验名称	回归模型		
姓名	***	学号	***
实验地点	***	实验日期	***

四、 实验作业及分析

4.1 实验过程

*** 请将详细实验过程的截图和相关说明填写在此处 ***

4.2 实验结果

*** 请将实验结果的截图和相关说明填写在此处 **

GOOD LUCK!