Bài giảng môn học:

Khoa Học Dữ Liệu (7080509)

CHƯƠNG 4:
MỘT SỐ THƯ VIỆN PYTHON TRONG KHOA HỌC DỮ LIỆU
(Phần 03)

08/2019

Nội dung chương 4

4.1 Giới thiệu một số thư viện Python trong KHDL

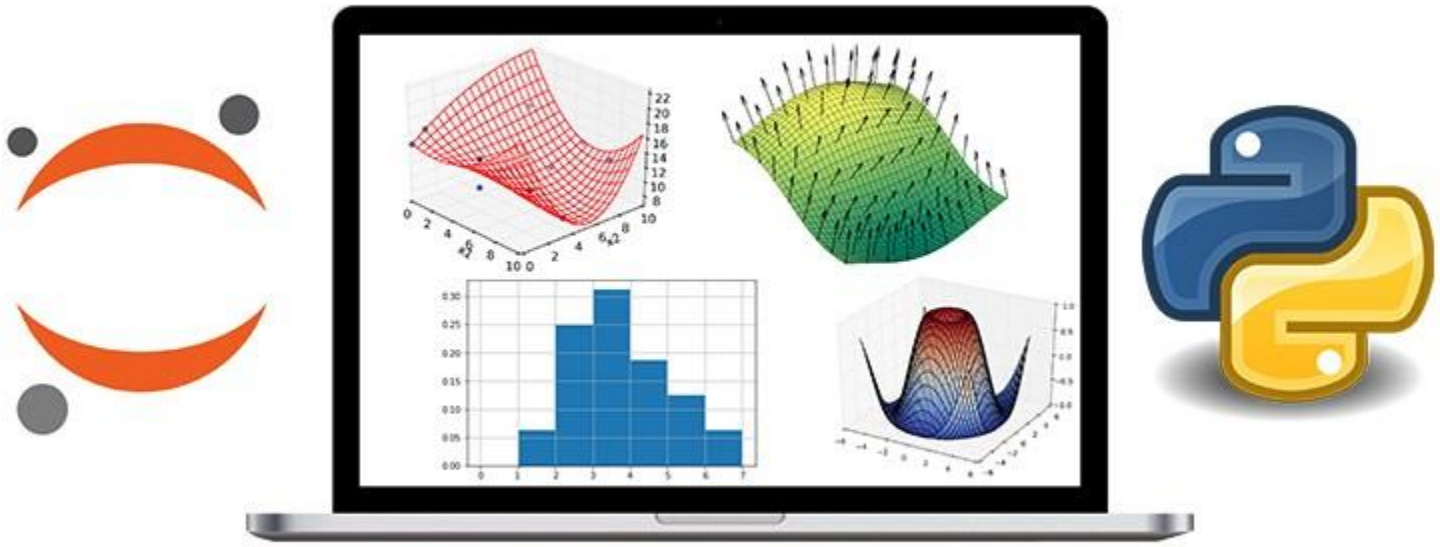
4.2 Thư viện Numpy *

4.3 Thư viện Pandas *

4.4 Thư viện Matplotlib*

4.5 Thư viện Scikit-learn

Chương 4: package Matplotlib



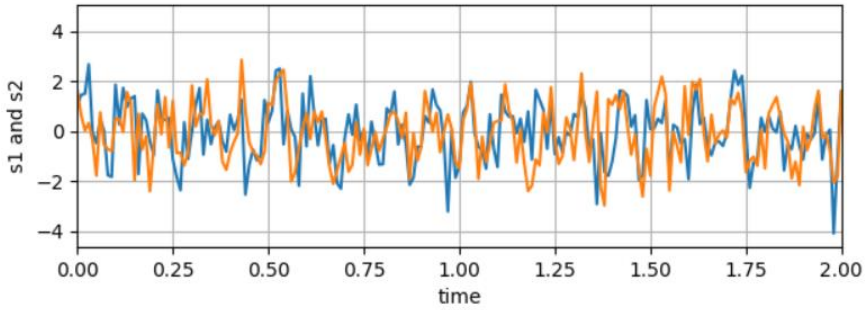
1. Giới thiệu Matplotlib

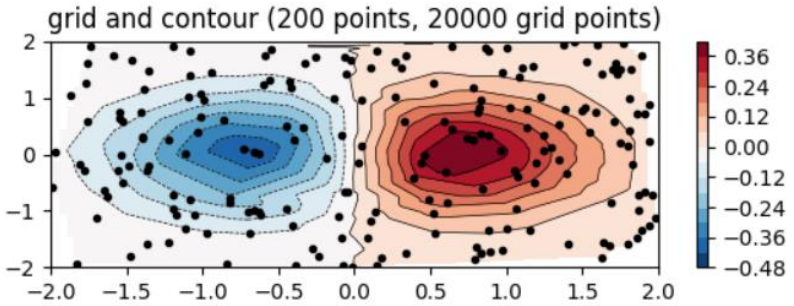
- **Matplotlib** là thư viện dùng để vẽ đồ thị rất mạnh mẽ, có cú pháp tương tự như Matlab.
- Hỗ trợ nhiều loại biểu đồ, đặt biệt là các loại được sử dụng trong nghiên cứu hoặc kinh tế như biểu đồng đường, cột, tần suất (histograms), tương quan, scatterplots…
- Cấu trúc của Matplotlib gồm nhiều phần, phục vụ cho các mục đích sử dụng khác nhau.
- Matplotlib miễn phí và mã nguồn mở.

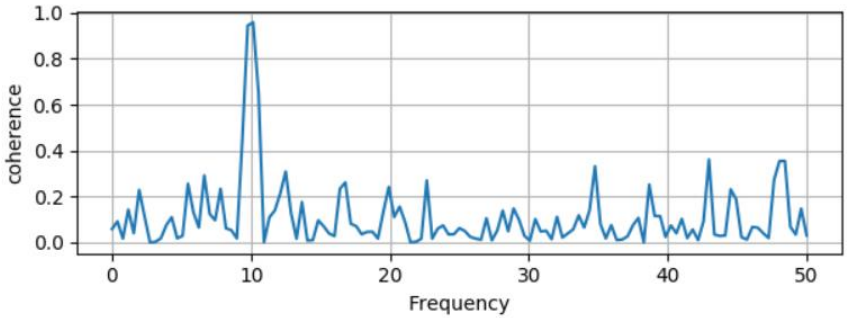
Tham khảo:

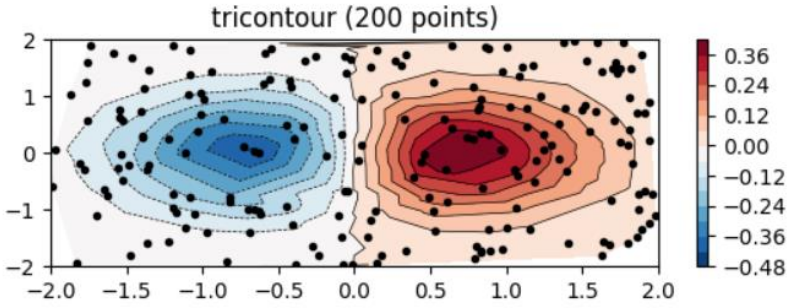
+ File: **CheatSheet-Matplotlib/Seaborn.pdf**
+ Link web: **Matplotlib package!**

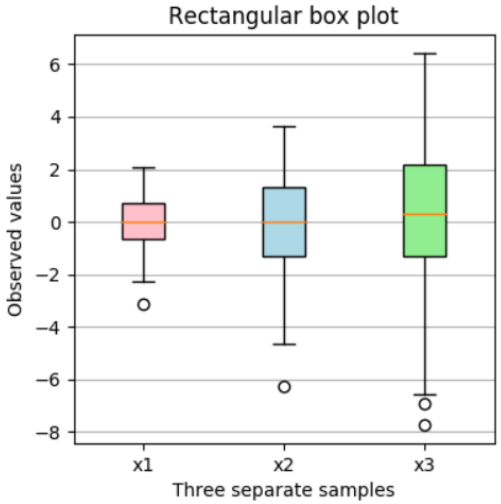
2. Một số biểu đồ vẽ bằng Matplotlib

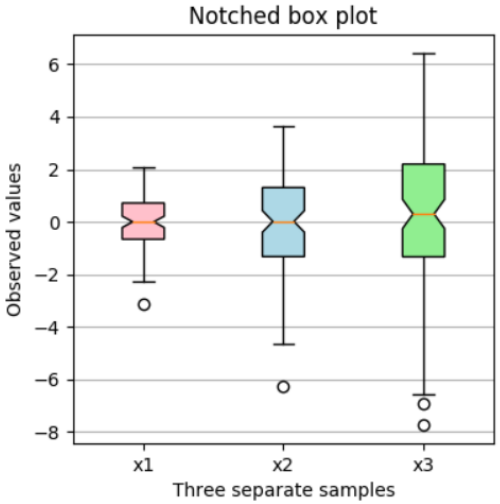


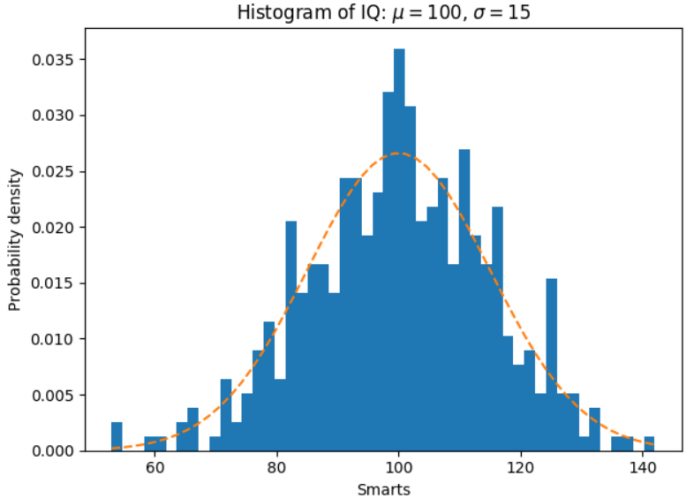












Đồ thị dạng đường (plot)

Cú pháp:

- **plot(x, y, 'go--', linewidth=2, markersize=12)**
- **plot(x, y, color='green', marker='o', linestyle='dashed', linewidth=2, markersize=12)**

Trong đó:

* X, Y – các tọa độ theo trục x và y;

Hàm pyplot.plot() còn có các tham số cơ bản sau:

* color = [‘b’ | ‘g’ | ‘r’ | ‘c’ | ‘m’ | ‘y’ | ‘k’ | ‘w’];
* linewidth = số thực - Độ rộng của đường đồ thị
* linestyle = [‘-‘ | ‘- -‘ | ‘-.’ | ‘:’ | ‘None’]; Kiểu đường đồ thị
* marker = [‘.’ | ‘,’ | ‘o’ | ‘+’ | ‘x’]
* markersize = float: Kích thước của điểm dữ liệu
* label = String;

Các tham số color, marker, linestyple có thể được biểu diễn ở dạng ‘[color][marker][linestyle]‘, ví dụ: ‘ro-‘ tương đương với color=’r’, marker=’o’, linestyle=’-‘.

Đồ thị dạng đường (plot)

Đọc lại file dữ liệu Data_Temp.csv để vẽ đồ thị

```
#ĐỌC DỮ LIỆU VÀ VẼ ĐỒ THỊ
#Khai báo sử dụng thư viện Pandas|Matplotlib
import pandas as pd
import matplotlib.pyplot as plt
#Xác định đường dẫn tới file dữ liệu
path ='Data_C4/Data_Temp.csv'
#Đọc file dữ liệu csv với pandas
data_df = pd.read_csv(path)

data_df.head(10)
```

	time	Ha Noi	Vinh	Da Nang	Nha Trang	Ho Chi Minh	Ca Mau
0	00 15-9-2019	25.65	24.79	24.01	25.06	25.48	24.97
1	01 15-9-2019	25.31	24.21	24.02	24.93	25.16	24.83
2	02 15-9-2019	25.05	23.73	23.89	24.79	24.80	24.55
3	03 15-9-2019	24.79	23.36	23.83	24.84	24.74	24.48
4	04 15-9-2019	24.59	23.05	23.69	24.82	24.80	24.38
5	05 15-9-2019	24.40	22.80	23.52	24.79	24.87	24.40
6	06 15-9-2019	24.38	22.79	23.68	25.10	24.71	24.41

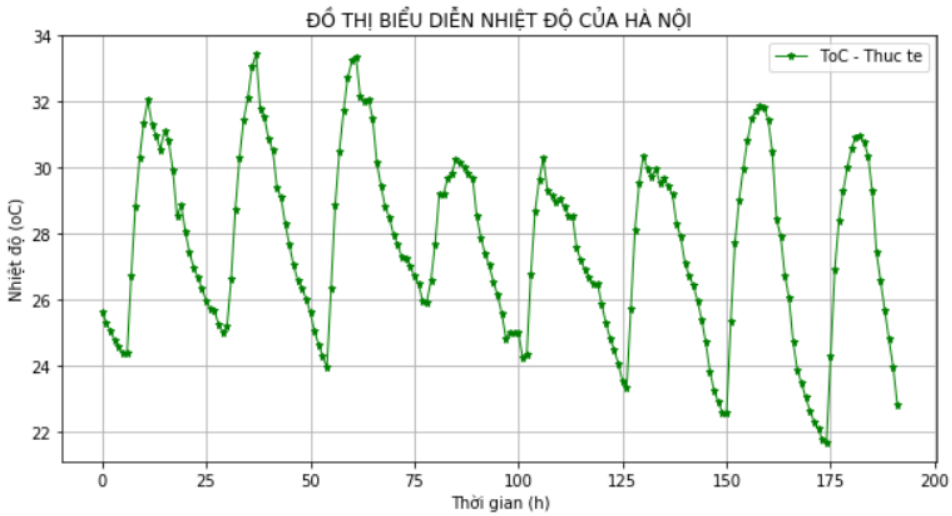
Đồ thị dạng đường (plot)

```python
#VẼ ĐỒ THỊ NHIỆT ĐỘ CỦA HÀ NỘI
y_hn = data_df['Ha Noi']

#-------------------------------------------------------------
fig,ax = plt.subplots(figsize=(10, 5))
#Vẽ biểu đồ dạng đường trục x: Thời gian | trục Y: Nhiệt độ
ax.plot(y_hn,color='green', marker='*',
        linestyle='-', linewidth=1,
        markersize=5, label="ToC - Thuc te")

#Hiển thị chú thích trong hình
ax.legend()
#Hiển thị lưới trong đồ thị
ax.grid(True)
#Thiết lập tiêu đề trục X, Y, và Tên hình
ax.set_ylabel('Nhiệt độ (oC)')
ax.set_xlabel('Thời gian (h)')
ax.set_title('ĐỒ THỊ BIỂU DIỄN NHIỆT ĐỘ CỦA HÀ NỘI')

plt.show()
```



Đồ thị dạng hình tròn (pie)

Cú pháp:

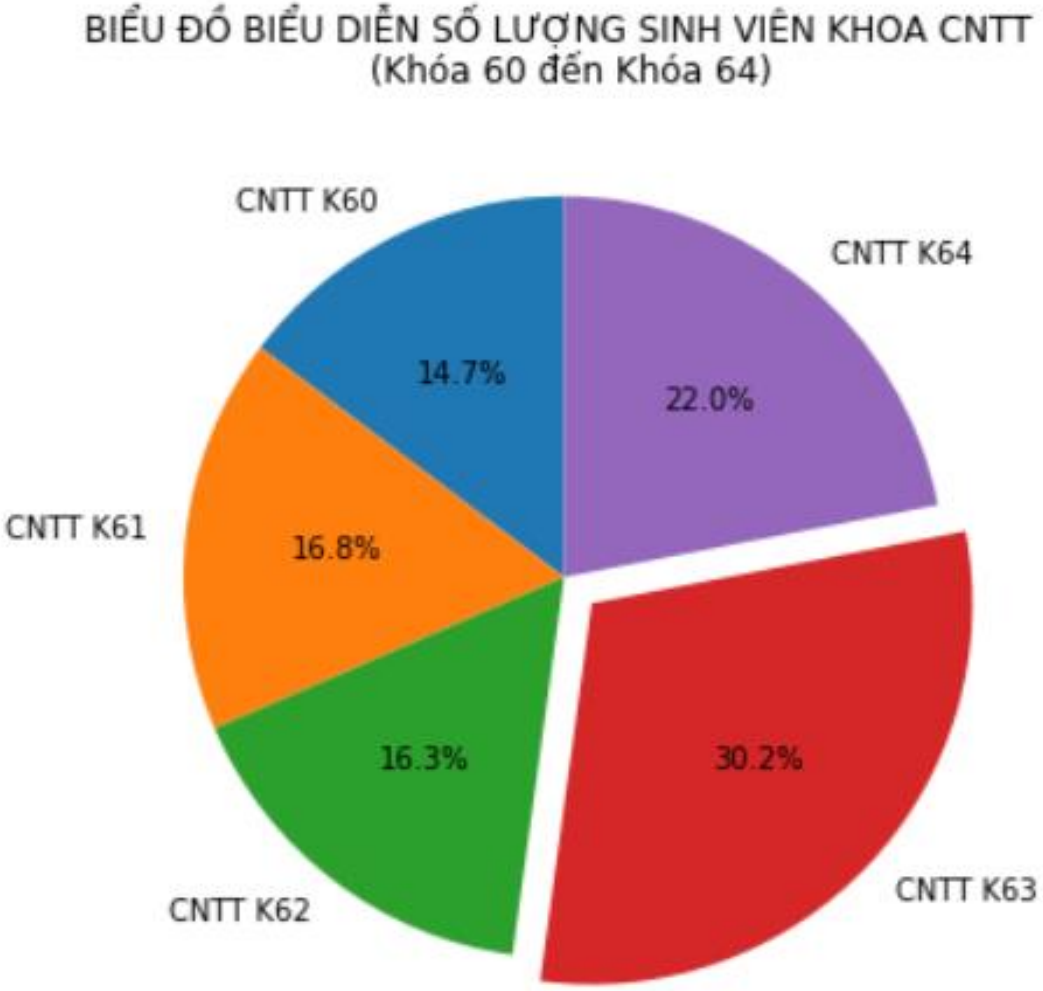
pie(x, explode=explode, labels=labels, autopct='%1.1f%%', startangle=90)

Các tham số cơ bản:

- x: Số liệu vẽ biểu đồ hình tròn (1 mảng, 1 list)
- explode: làm nổi bật một phần nào đó trong biểu đồ (>0 tách)
- labels: Nhãn của mỗi phần trong biểu đồ

```
1  #Khai báo thư viện
2  import matplotlib.pyplot as plt
3
4  # Khai báo dữ liệu vẽ biểu đồ hình tròn
5  ds_khoa = ['CNTT K60', 'CNTT K61', 'CNTT K62', 'CNTT K63', 'CNTT K64']
6  so_sv = [350, 400, 389, 719, 523]
7  #-------------------------------------------------------------------------------
8  fig1, ax1 = plt.subplots(figsize=(6, 6))
9  #Tách phần CNTT K63 làm nổi bật
10 explode = (0, 0, 0, 0.1, 0)
11 ax1.pie(so_sv, explode=explode, labels=ds_khoa,
12         autopct='%1.1f%%',  startangle=90)
13
14 #Thiết lập tiêu đề cho đồ thị
15 ax1.set_title('BIỂU ĐỒ BIỂU DIỄN SỐ LƯỢNG SINH VIÊN KHOA CNTT \n (Khóa 60 đến Khóa 64)')
16
17 plt.show()
```

Đồ thị dạng hình tròn (pie)



Đồ thị dạng cột (bar)

Cú pháp:

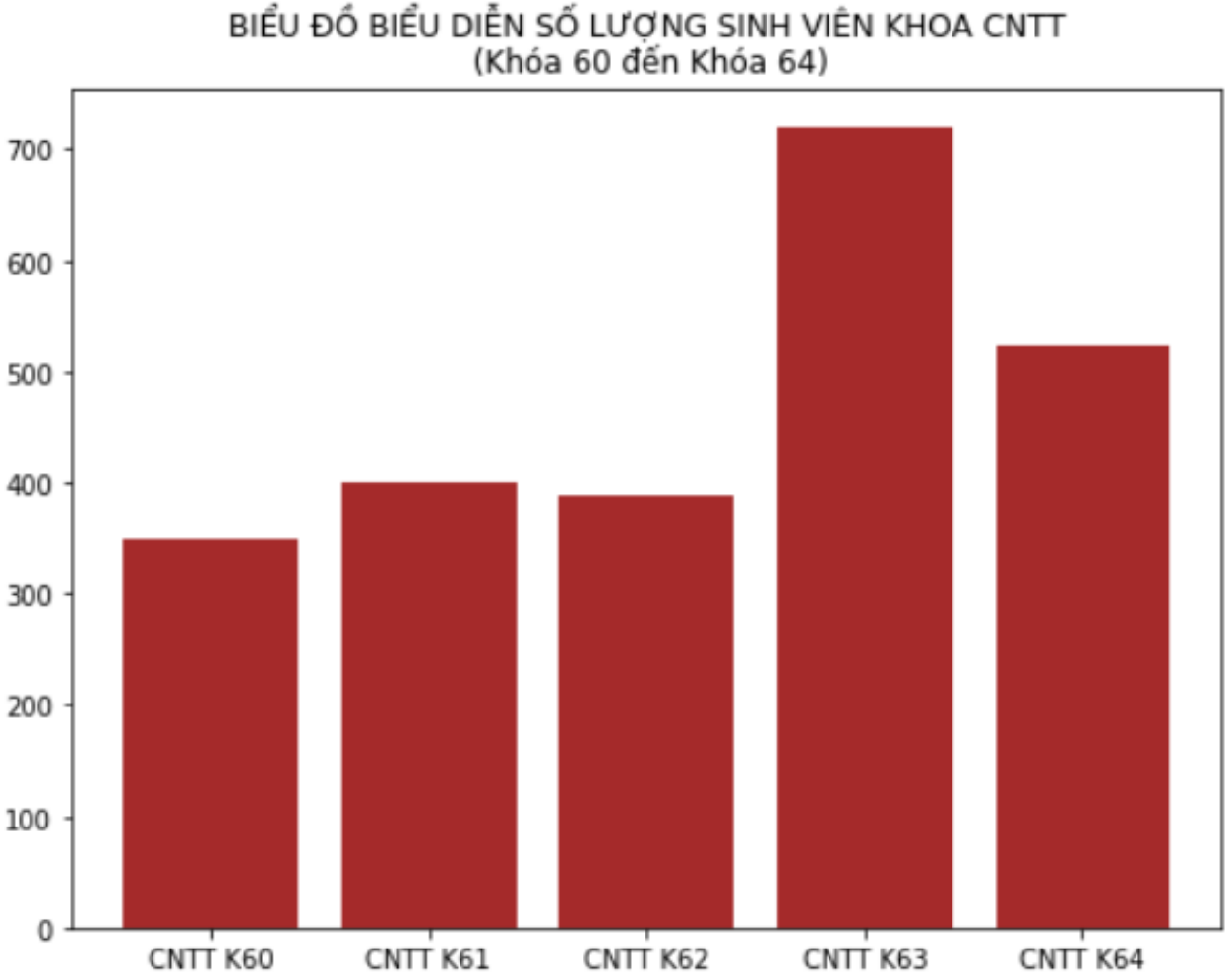
.bar(name, values)

Các tham số cơ bản:

- name: Tên cột
- Values: Giá trị của dữ liệu muốn vẽ đồ thị

```
1  #Khai báo thư viện
2  import matplotlib.pyplot as plt
3
4  # Khai báo dữ liệu vẽ biểu đồ
5  ds_khoa = ['CNTT K60', 'CNTT K61', 'CNTT K62', 'CNTT K63', 'CNTT K64']
6  so_sv = [350, 400, 389, 719, 523]
7  #------------------------------------------------------------
8  fig2,ax2 = plt.subplots(figsize=(8,6))
9  #Vẽ biểu đồ hình cột
10 ax2.bar(ds_khoa, so_sv,color='brown')
11
12 #Thiết lập tiêu đề cho đồ thị
13 ax2.set_title('BIỂU ĐỒ BIỂU DIỄN SỐ LƯỢNG SINH VIÊN KHOA CNTT\n (Khóa 60 đến Khóa 64)')
14
15 plt.show()
```

Đồ thị dạng cột (bar)



Đồ thị dạng điểm (scatter)

Cú pháp:

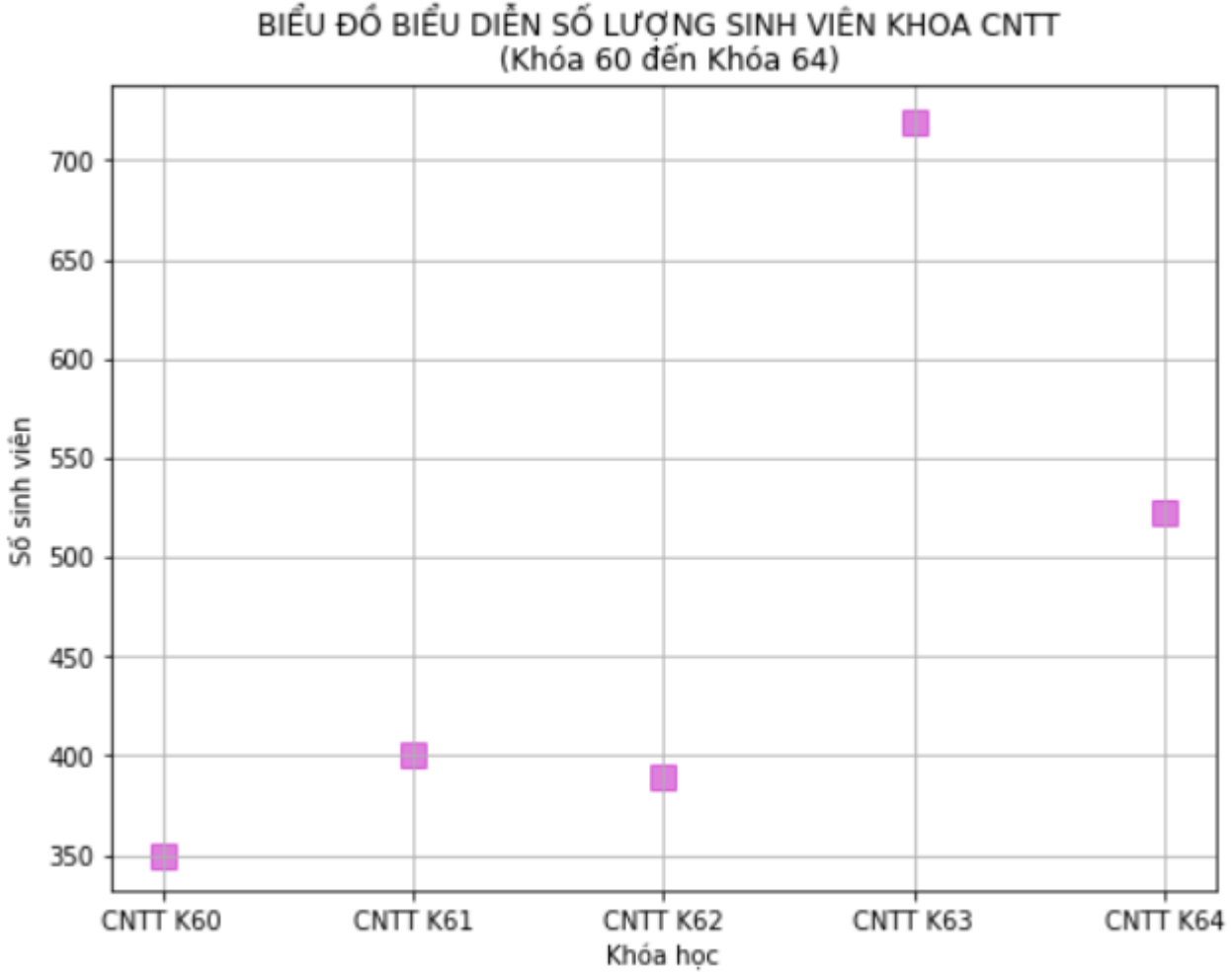
.scatter(name, values, maker, s, color, alpha)

Các tham số cơ bản sau:

- marker – kiểu điểm, ['.' | ',' | 'o' | '+' | 'x']
- s – kích thước của điểm
- color = ['b' | 'g' | 'r' | 'c' | 'm' | 'y' | 'k' | 'w']
- alpha – độ mờ đục, trong khoảng [0.0, 1.0].

```
1  #Khai báo thư viện
2  import matplotlib.pyplot as plt
3
4  # Khai báo dữ liệu vẽ biểu đồ
5  ds_khoa = ['CNTT K60', 'CNTT K61', 'CNTT K62', 'CNTT K63', 'CNTT K64']
6  so_sv = [350, 400, 389, 719, 523]
7  #-------------------------------------------------------------------------------
8  #Vẽ biểu đồ dạng điểm
9  fig, ax = plt.subplots(figsize=(8, 6))
10 plt.scatter(ds_khoa, so_sv, marker='s', s=100, color='m',alpha=0.5 )
11
12 ax.grid(True)
13 #Thiết lập tiêu đề trục X, Y, và Tên hình
14 ax.set_ylabel('Số sinh viên')
15 ax.set_xlabel('Khóa học')
16 ax.set_title('BIỂU ĐỒ BIỂU DIỄN SỐ LƯỢNG SINH VIÊN KHOA CNTT \n (Khóa 60 đến Khóa 64)')
17
18 plt.show()
```

Đồ thị dạng điểm (scatter)



Thực hành

Bài 20: Làm việc với Matplotlib

Sử dụng kết quả thống kê được ở Bài tập 18 về nhiệt độ max, min, mean của 6 thành phố:

	Hà Nội	Vinh	Đà Nẵng	Nha Trang	Hồ Chí Minh	Cà Mau
Max	33.45	32.57	29.88	28.68	31.06	31.37
Mean	27.71	26.72	25.52	26.17	26.16	26.73
Min	21.68	22.60	20.93	24.50	23.22	23.99

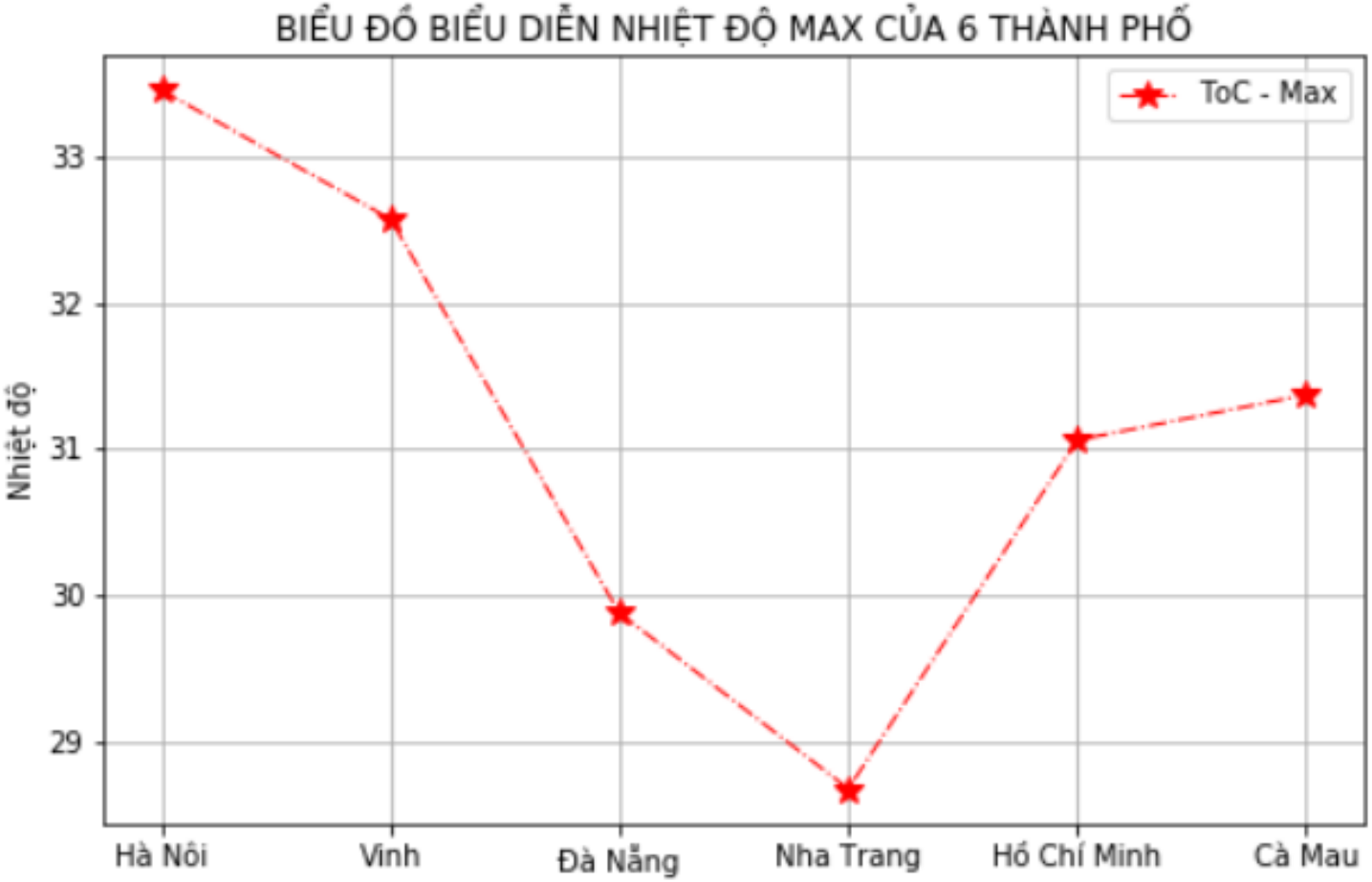
Yêu cầu:

1) **Vẽ biểu đồ dạng đường biểu diễn nhiệt độ Max của 6 thành phố**
2) **Vẽ biểu đồ dạng cột biểu diễn nhiệt độ Mean của 6 thành phố**
3) **Vẽ biểu đồ dạng điểm biểu diễn nhiệt độ Min của 6 thành phố**

(Thiết lập các tham số để có được kết quả như hình minh họa)

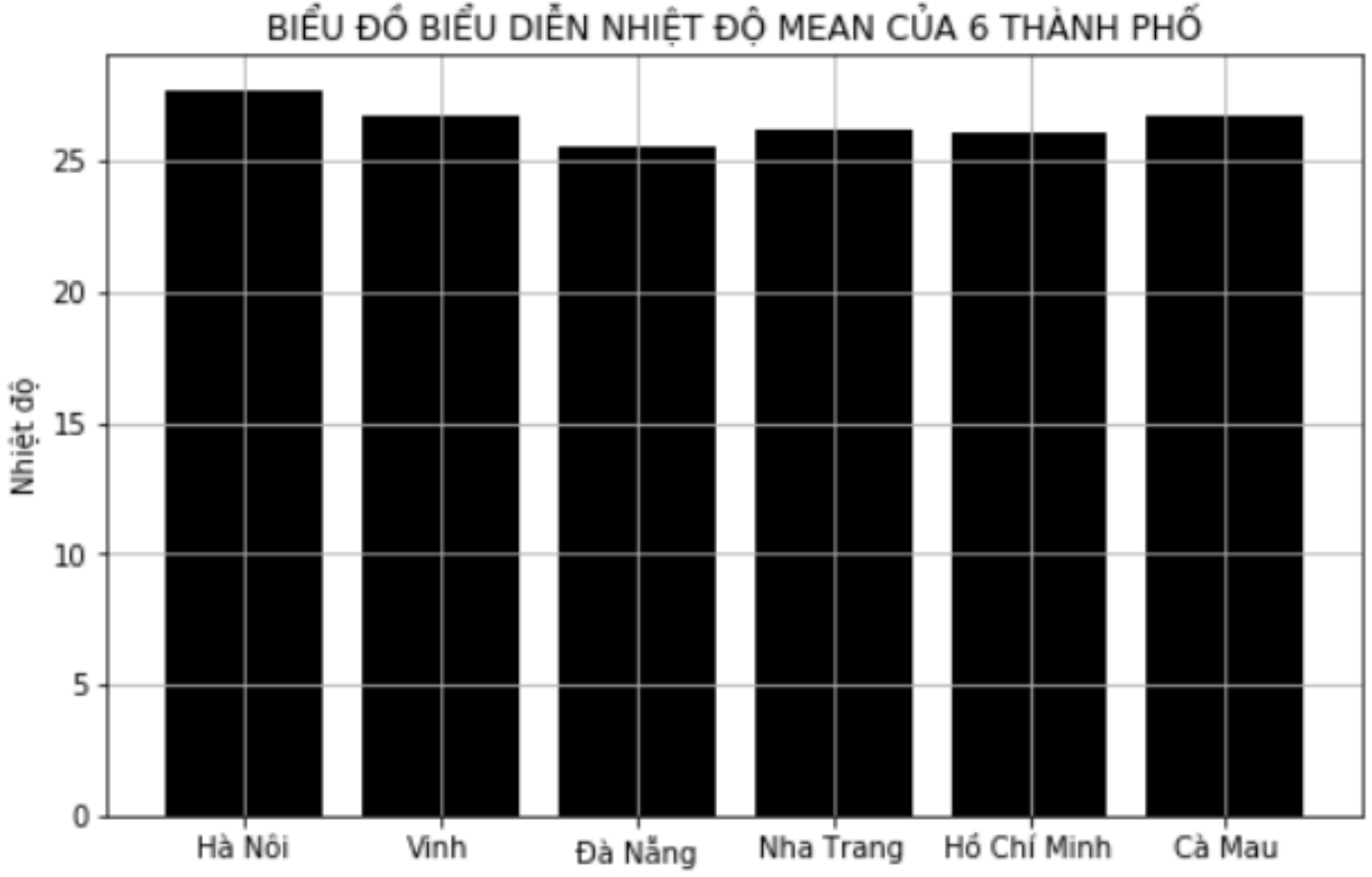
Bài 20: Làm việc với Matplotlib

1) Vẽ biểu đồ dạng đường biểu diễn nhiệt độ Max của 6 thành phố



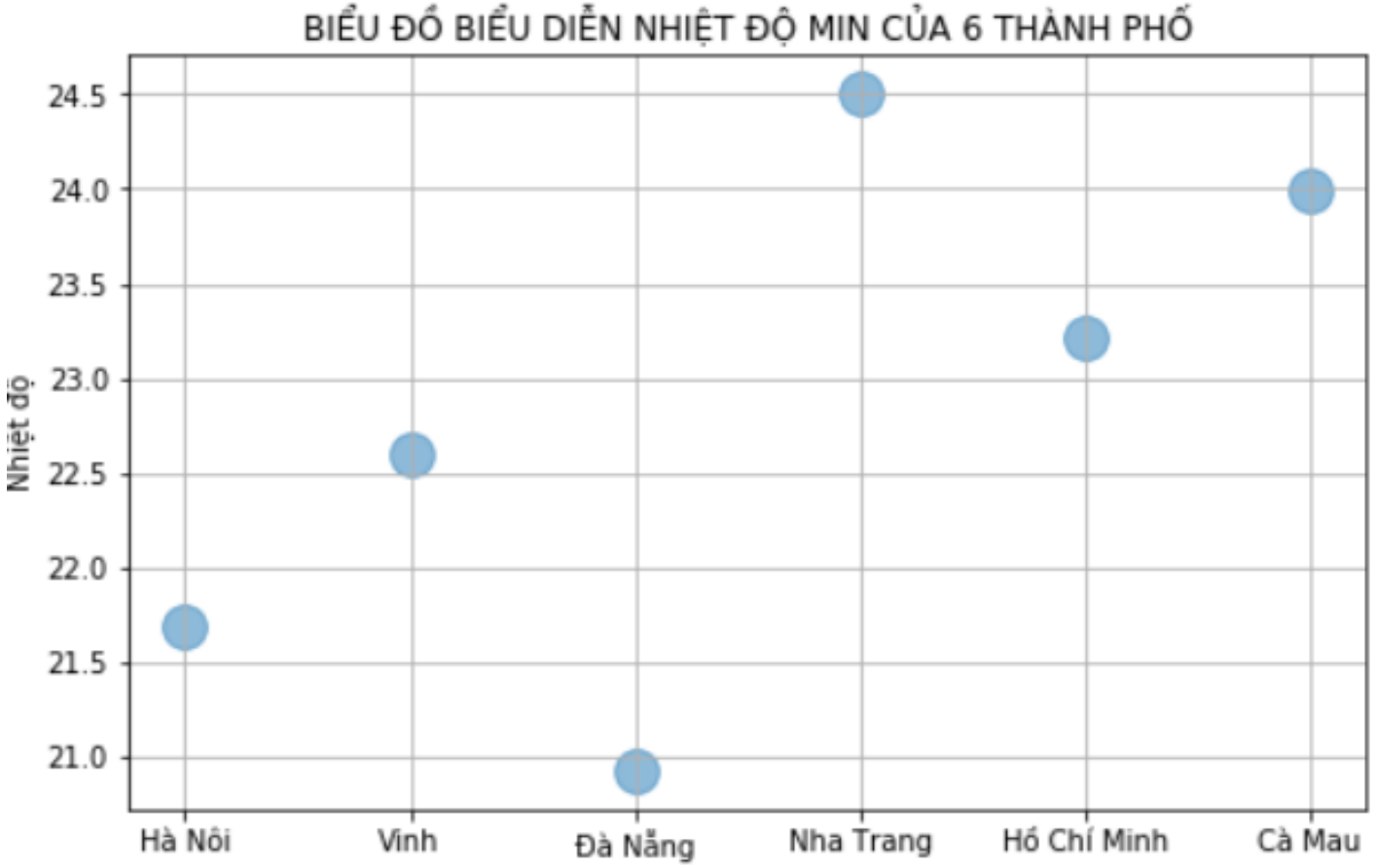
Bài 20: Làm việc với Matplotlib

2) Vẽ biểu đồ dạng cột biểu diễn nhiệt độ Mean của 6 thành phố

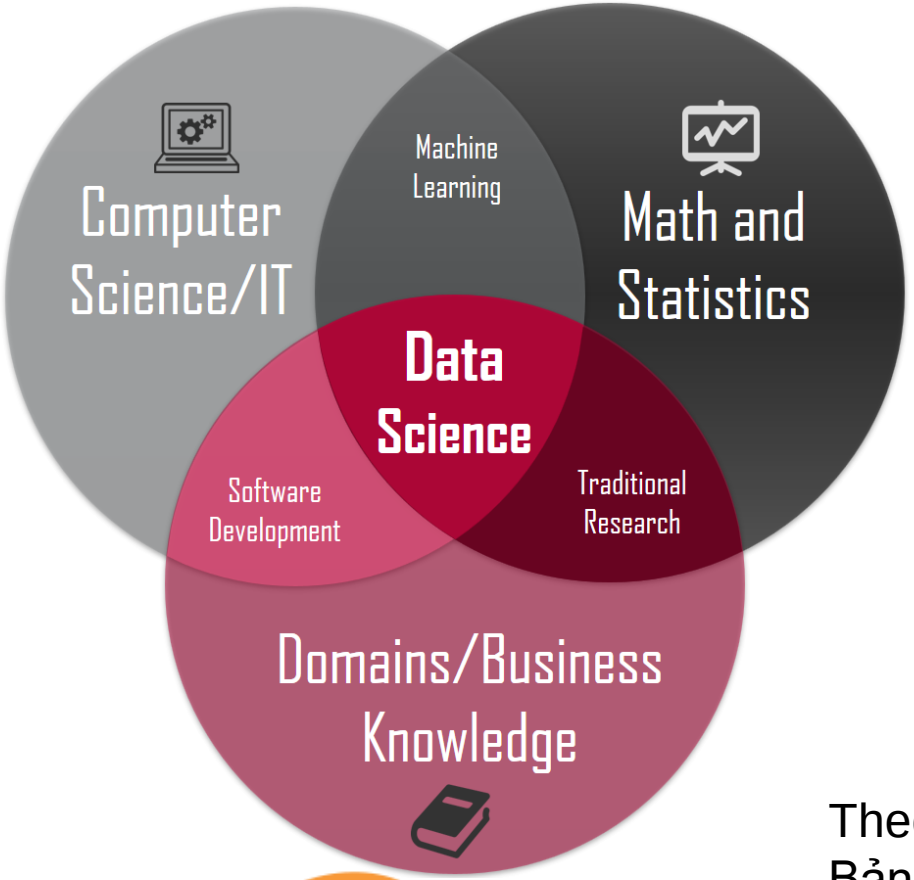


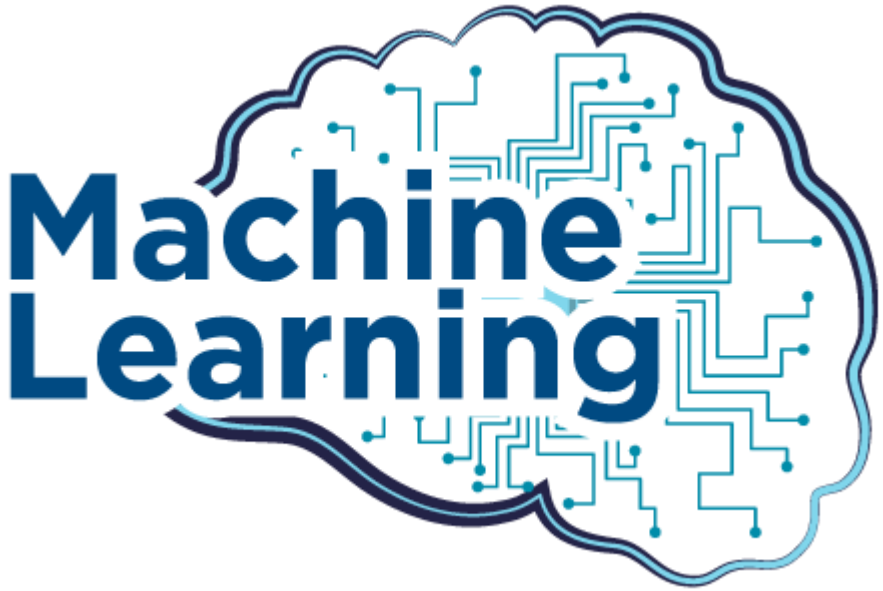
Bài 20: Làm việc với Matplotlib

3) Vẽ biểu đồ dạng điểm biểu diễn nhiệt độ Min của 6 thành phố



Chương 4: package Scikits-learn (SINH VIÊN TÌM HIỂU THÊM)



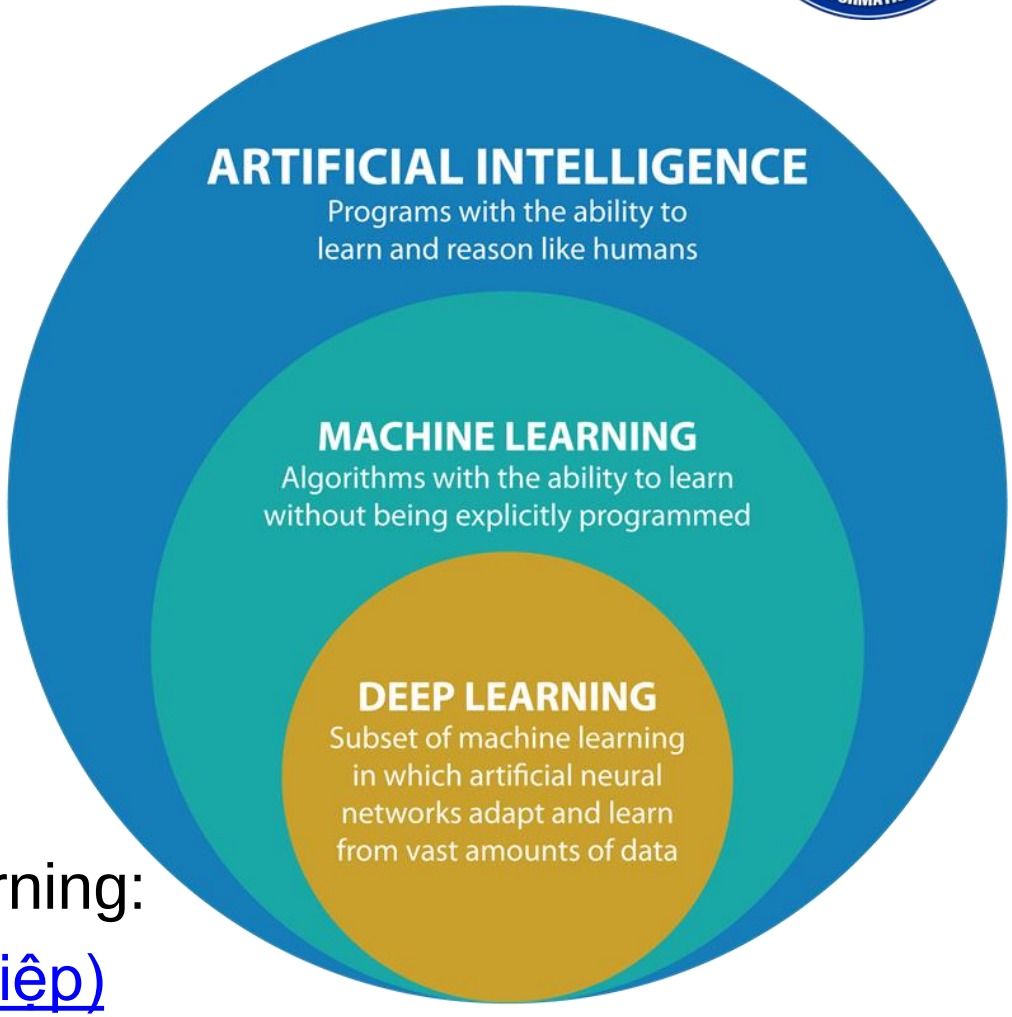


Theo GS.TSKH Hồ Tú Bảo – Viện JAIST, Nhật Bản: "KHDL là khoa học dựa trên sự kết hợp của toán học (tiêu biểu là thống kê) và công nghệ thông tin (tiêu biểu là machine learning)"

scikits learn
machine learning in Python

Học máy

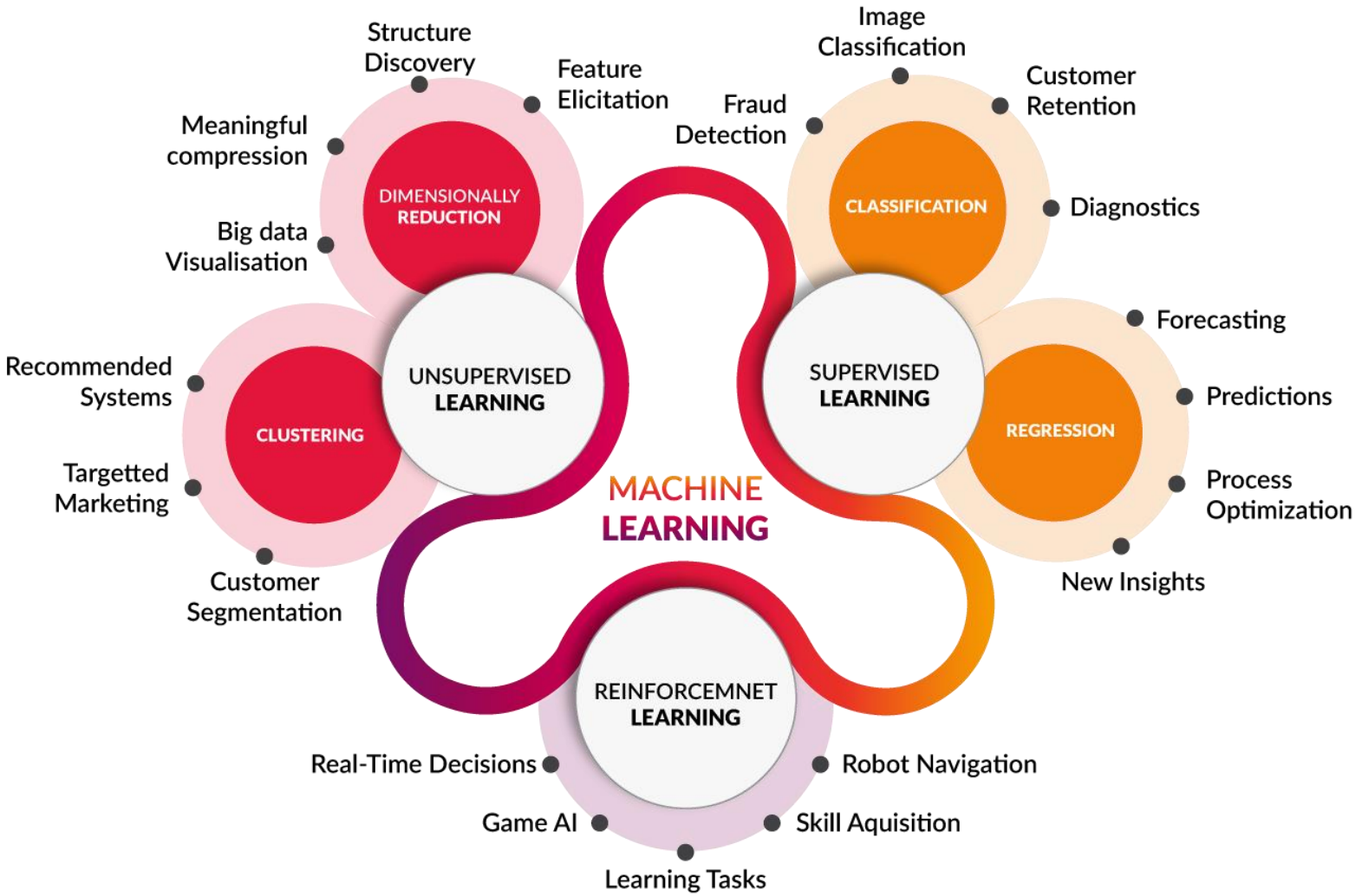
Machine Learning là một tập con của AI. Nói đơn giản, Machine Learning là một lĩnh vực nhỏ của Khoa Học Máy Tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể



Link web hay về Machine Learning:

+ Học máy cơ bản (Vũ Hữu Tiệp)
+ Machine Learning cho người bắt đầu (Ông Xuân Hồng)

Phân loại và các ứng dụng cơ bản của ML



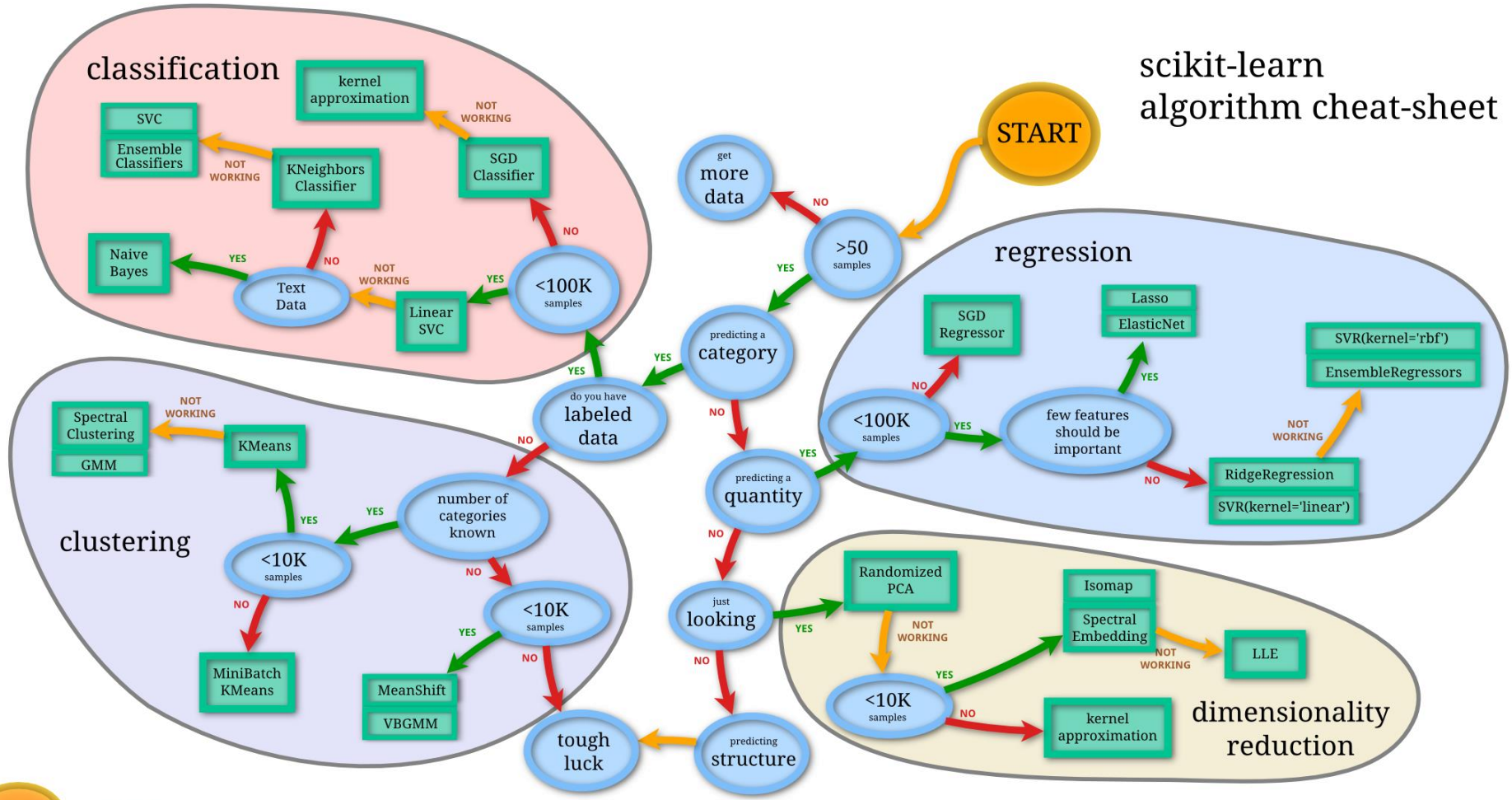
1. Giới thiệu Scikit-learn

- **Scikit-learn** (viết tắt là sklearn) xuất phát là một dự án trong một cuộc thi lập trình của Google vào năm 2007.
- Sau đó nhiều viện nghiên cứu và các nhóm ra nhập để phát triển Sklearn.
- Là một thư viện mã nguồn mở dành cho học máy, rất mạnh mẽ và thông dụng được viết bằng ngôn ngữ Python.
- Scikit-learn chứa hầu hết các thuật toán machine learning hiện đại nhất, đi kèm với tài liệu rất chi tiết và luôn được cập nhật.

Tham khảo:

+ File: **CheatSheet-Scikit-learn.pdf**
+ Link web: **https://scikit-learn.org/stable/**

Các thuật toán cơ bản trong sklearn



Back

Các bước thực hiện Sklearn

- **Bước 1:** Tải tập dữ liệu đã chuẩn bị để đưa vào model
- **Bước 2:** Tách tập dữ liệu X_Train, Y_Train
- **Bước 3:** Tạo model phù hợp với bài toán

Supervised Learning Estimators

Linear Regression

```
>>> from sklearn.linear_model import LinearRegression
>>> lr = LinearRegression(normalize=True)
```

Support Vector Machines (SVM)

```
>>> from sklearn.svm import SVC
>>> svc = SVC(kernel='linear')
```

Naive Bayes

```
>>> from sklearn.naive_bayes import GaussianNB
>>> gnb = GaussianNB()
```

KNN

```
>>> from sklearn import neighbors
>>> knn = neighbors.KNeighborsClassifier(n_neighbors=5)
```

- **Mỗi một model có rất nhiều tham số khác nhau:**
  - **Tìm hiểu các bước thực hiện cơ bản của thuật toán tương ứng.**
  - **Các tham số cần thiết, thiết lập giá trị cho các tham số.**

Các bước thực hiện Sklearn

- **Bước 4:** Huấn luyện Model với dữ liệu Train

Model Fitting

Supervised learning

```
>>> lr.fit(X, y)                          Fit the model to the data
>>> knn.fit(X_train, y_train)
>>> svc.fit(X_train, y_train)
```

Unsupervised Learning

```
>>> k_means.fit(X_train)                  Fit the model to the data
>>> pca_model = pca.fit_transform(X_train)  Fit to data, then transform it
```

- **Bước 5:** Dự đoán với tập Test trên model xây dựng được

Prediction

Supervised Estimators

```
>>> y_pred = svc.predict(np.random.random((2,5)))   Predict labels
>>> y_pred = lr.predict(X_test)                     Predict labels
>>> y_pred = knn.predict_proba(X_test)              Estimate probability of a label
```

Unsupervised Estimators

```
>>> y_pred = k_means.predict(X_test)      Predict labels in clustering algos
```

Các bước thực hiện Sklearn

- **Bước 6:** Đánh giá độ chính xác của Model

Classification Metrics

Accuracy Score

```
>>> knn.score(X_test, y_test)
```

Estimator score method

```
>>> from sklearn.metrics import accuracy_score
>>> accuracy_score(y_test, y_pred)
```

Metric scoring functions

Classification Report

```
>>> from sklearn.metrics import classification_report
>>> print(classification_report(y_test, y_pred))
```

Precision, recall, f1-score and support

Confusion Matrix

```
>>> from sklearn.metrics import confusion_matrix
>>> print(confusion_matrix(y_test, y_pred))
```

Các bước thực hiện Sklearn

- **Bước 7:** Lựa chọn tham số tối ưu của Model

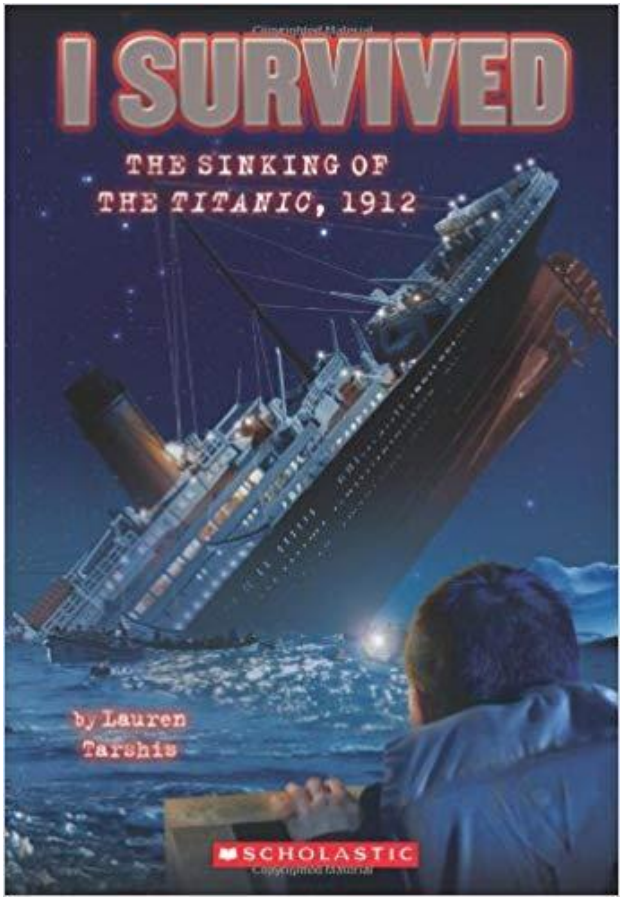
Randomized Parameter Optimization

```
>>> from sklearn.grid_search import RandomizedSearchCV
>>> params = {"n_neighbors": range(1,5),
              "weights": ["uniform", "distance"]}
>>> rsearch = RandomizedSearchCV(estimator=knn,
                                 param_distributions=params,
                                 cv=4,
                                 n_iter=8,
                                 random_state=5)
>>> rsearch.fit(X_train, y_train)
>>> print(rsearch.best_score_)
```

Bài toán 1: Supervised Learning/Classification

DỰ ĐOÁN KHẢ NĂNG SỐNG SÓT CỦA HÀNH KHÁCH

- Đề cập tới nhiều Kỹ thuật quan trọng trong Cleaning data:
  - Missing data
  - LabelEncoder
  - Recale data
  - Drop, birth data
- Giải quyết bài toán phân lớp trong Machine learning học có giám sát (Supervised)
- Các bước xây dựng model, huấn luyện…với thuật toán KNN



- **Link mã nguồn trên Colab:** [Các bước giải bài toán 1]
- **Link tải file dữ liệu:** [All file data]

Bài toán 2: Unsupervised Learning/Clustering

KHÁM PHÁ TỪ DỮ LIỆU CHI TIÊU TẠI CỬA HÀNG BÁN BUÔN

- Đề cập tới 2 Kỹ thuật quan trọng trong Cleaning data:
  - Fetures Scaling
  - Outlier detection
- Giải quyết bài toán phân cụm trong lớp bài toán Machine learning học không giám sát (Unsupervised)
- Các bước xây dựng model, huấn luyện…với thuật toán Kmean

- **Mô tả và mã nguồn trên Colab:** Lời giải bài toán 2
- **Link tải file dữ liệu:** All file data