

Thực hành Nguyên Lý Máy Học

Buổi 2: Giải thuật KNN & Bayes thơ ngây

Mục tiêu:

- Củng cố lý thuyết và cài đặt giải thuật KNN và Bayes thơ ngây
- Kiểm thử và đánh giá theo nghi thức **hold-out**

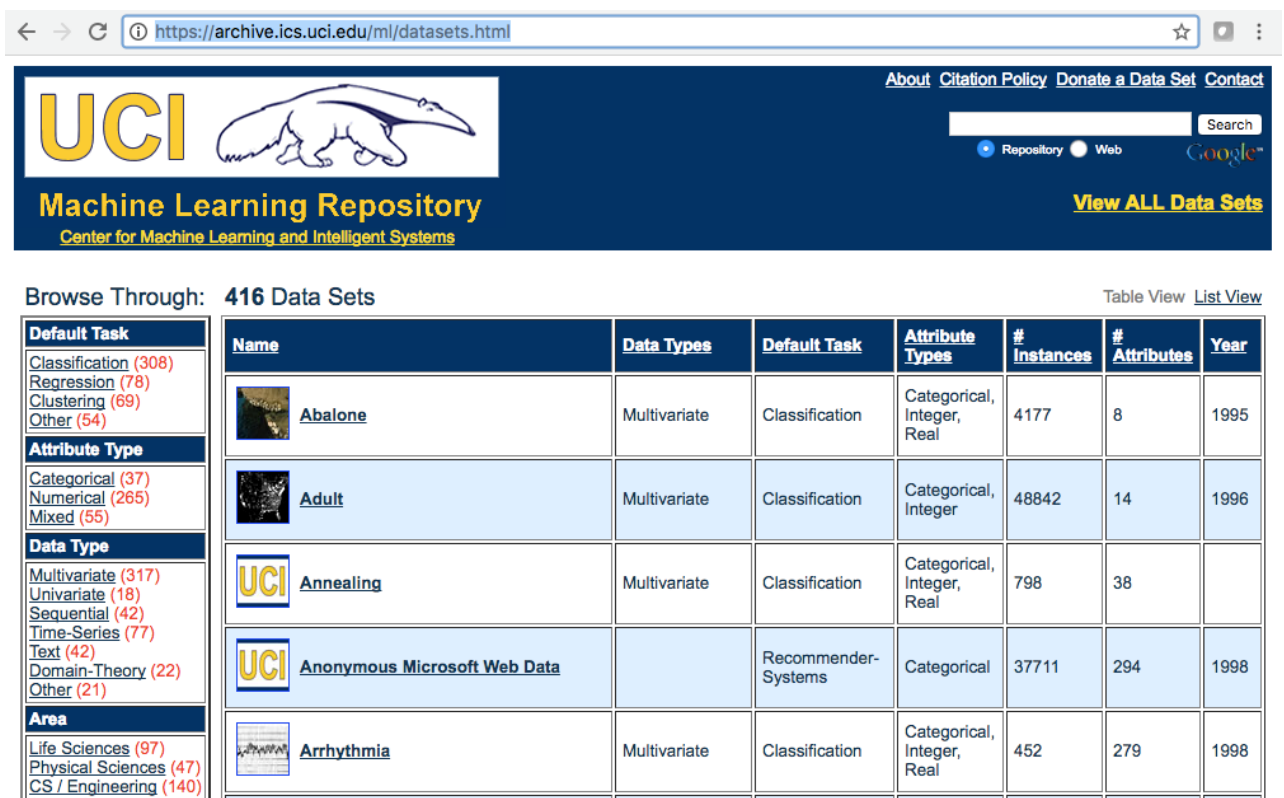
1. HƯỚNG DẪN THỰC HÀNH

Cách cài đặt một số thư viện cần thiết






- Cài đặt một số thư viện phục vụ cho bài thực hành: pandas, sklearn
 - pip install pandas // đọc file csv**
 - pip install sklearn**

Trang web lưu trữ các tập dữ liệu sử dụng trong quá trình thực hành

<https://archive.ics.uci.edu/ml/datasets.html>
<https://archive.ics.uci.edu/ml/datasets.php>



The screenshot shows the UCI Machine Learning Repository website. The header includes the UCI logo, a search bar, and links for About, Citation Policy, Donate a Data Set, and Contact. Below the header, there's a section titled "Browse Through: 416 Data Sets" with a "Table View" link. A table lists several datasets with columns: Name, Data Types, Default Task, Attribute Types, # Instances, # Attributes, and Year. The datasets listed are Abalone, Adult, Annealing, Anonymous Microsoft Web Data, and Arrhythmia. On the left side of the table, there are filters for Default Task, Attribute Type, Data Type, and Area, each with a list of categories and their counts.

Name	Data Types	Default Task	Attribute Types	# Instances	# Attributes	Year
 Abalone	Multivariate	Classification	Categorical, Integer, Real	4177	8	1995
 Adult	Multivariate	Classification	Categorical, Integer	48842	14	1996
 Annealing	Multivariate	Classification	Categorical, Integer, Real	798	38	
 Anonymous Microsoft Web Data		Recommender-Systems	Categorical	37711	294	1998
 Arrhythmia	Multivariate	Classification	Categorical, Integer, Real	452	279	1998

Default Task
Classification (308)
Regression (78)
Clustering (69)
Other (54)

Attribute Type
Categorical (37)
Numerical (265)
Mixed (55)

Data Type
Multivariate (317)
Univariate (18)
Sequential (42)
Time-Series (77)
Text (42)
Domain-Theory (22)
Other (21)

Area
Life Sciences (97)
Physical Sciences (47)
CS / Engineering (140)
Social Sciences (24)

Tập dữ liệu rượuvang sẽ sử dụng trong phần bài tập

Wine Quality Data Set

Download: [Data Folder](#), [Data Set Description](#)

Abstract: Two datasets are included, related to red and white vinho verde wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests (see [Cortez et al., 2009], [Web Link]).



Data Set Characteristics:	Multivariate	Number of Instances:	4898	Area:	Business
Attribute Characteristics:	Real	Number of Attributes:	12	Date Donated	2009-10-07
Associated Tasks:	Classification, Regression	Missing Values?	N/A	Number of Web Hits:	578954

Index of /ml/machine-learning-databases/wine-quality

Name	Last modified	Size	Description
Parent Directory		-	
winequality-red.csv	16-Oct-2009 14:36	82K	
winequality-white.csv	16-Oct-2009 14:36	258K	
winequality.names	21-Oct-2009 11:00	3.2K	

Apache/2.2.15 (CentOS) Server at archive.ics.uci.edu Port 443

• Tập dữ liệu Iris

Xét bài toán phân loại hoa IRIS dựa trên thông tin về kích thước của cánh hoa và đài hoa. Tập dữ liệu này có 150 phần tử, mỗi loại hoa có 50 phần tử. Dữ liệu có 4 thuộc tính (sepal length, sepal width, petal length, petal width) và 3 lớp (3 loại hoa Iris: Setosa, Versicolour, Virginica)



Tập dữ liệu này có thể download từ trang UCI (<https://archive.ics.uci.edu/ml/datasets/iris>) rồi đọc dữ liệu bằng lệnh `read_csv` của thư viện **Pandas** hoặc có thể nạp dữ liệu có sẵn bởi thư viện **Sklearn**

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

A. Giải thuật KNN

- Sử dụng tập dữ liệu có sẵn "iris"

```
#Lay file iris truc tiep tu sklearn
from sklearn.datasets import load_iris
iris_dt = load_iris()
iris_dt.data[1:5] # thuoc tinh cua tap iris
iris_dt.target[1:5] #gia tri cua nhan /class
```

- *Phân chia tập dữ liệu để xây dựng mô hình và kiểm tra theo nghi thức Hold-out*

```
from sklearn.cross_validation import train_test_split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(iris_dt.data, iris_dt.target, test_size=1/3.0,
random_state=5)

X_train[1:6]
X_train[1:6,1:3]
y_train[1:6]
X_test[6:10]
y_test[6:10]
```

- *Xây dựng mô hình K láng giềng KNN, với 5 láng giềng.*

```
# Xay dung mo hinh KNN
from sklearn.neighbors import KNeighborsClassifier
Mohinh_KNN = KNeighborsClassifier(n_neighbors=5)
Mohinh_KNN.fit(X_train,y_train)
```

- *Dự đoán nhãn cho các phần tử trong tập kiểm tra*

```
# du doan
y_pred = Mohinh_KNN.predict(X_test)
y_test
Mohinh_KNN.predict([[4, 4, 3, 3]])
```

- *Tính độ chính xác cho giá trị dự đoán của phần tử trong tập kiểm tra*

```
# tính độ chính xác
from sklearn.metrics import accuracy_score
print ("Accuracy is ", accuracy_score(y_test,y_pred)*100)
```

Kết quả thu được
Accuracy is 98.0

- Tính độ chính xác cho giá trị dự đoán thông qua ma trận con

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred, labels=[2,0,1])
```

Kết quả thu được

```
>>> confusion_matrix(y_test, y_pred, labels=[2,0,1])
array([[17,  0,  0],
       [ 0, 16,  0],
       [ 1,  0, 16]])
```

B. Giải thuật Bayes thơ ngây

- Dữ liệu kiểu số => giả sử các thuộc tính có phân phối *Gaussian*
Hàm mật độ xác suất được tính bởi công thức

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Trong đó:

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad \sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2$$

- Sklearn cung cấp sẵn hàm để tính mật độ xác suất theo phân phối Gaussian cho dữ liệu kiểu liên tục cũng như Multinomial (phân loại văn bản), BernoulliNB,...

```
from sklearn.naive_bayes import GaussianNB
from sklearn.naive_bayes import MultinomialNB
```

- Đọc dữ liệu từ file

```
import pandas as pd
dulieu = pd.read_csv(iris_data.csv")
X= dulieu.iloc[:,0:4]
y=dulieu.nhan
```

- Sử dụng hàm train_test_split() để phân chia dữ liệu, xây dựng mô hình theo phân phối Gaussian và so sánh kết quả dự đoán so với kết quả thực tế.

```

# Phân chia dữ liệu thành tập test và train
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
# Xây dựng mô hình dựa trên phân phối xác suất tuân theo Gaussian
model = GaussianNB()
model.fit(X_train, y_train)
print(model)
# dự đoán
thucte = y_test
dubao = model.predict(X_test)
thucte
dubao

```

- Sử dụng hàm `confusion_matrix()` để đánh giá giải thuật

```

from sklearn.metrics import confusion_matrix
cnf_matrix_gnb = confusion_matrix(thucte, dubao)
print(cnf_matrix_gnb)
[[16  0  0]
 [ 0 18  0]
 [ 0  0 11]]

```


BÀI TẬP KIỂM TRA BUỔI 2 – KNN & Bayes thơ ngây

- a. Đọc dữ liệu từ tập dữ liệu đánh giá chất lượng rượu vang **đỏ** trên trang UCI
<https://archive.ics.uci.edu/ml/datasets/wine+quality>
- b. Tập dữ liệu có bao nhiêu phần tử, Có bao nhiêu giá trị nhãn khác nhau? Ghi chú kết quả trong file code sau câu lệnh.
len(tên biến) : số lượng phần tử
import numpy as np
np.unique(tên biến) : liệt kê các giá trị khác nhau của biến
tên biến.value_counts() : liệt kê số lượng và giá trị khác nhau của biến
- c. Sử dụng **4 phần để** xây dựng mô hình và sử dụng **1 phần** để thực hiện đánh giá mô hình. Anh/chị ghi chú lại số lượng phần tử trong tập test và các giá trị nhãn khác nhau của các phần tử thuộc tập test trong file code.
- d. Xây dựng mô hình KNN với 9 láng giềng dựa trên tập dữ liệu học tạo ra ở bước c.
 - i. Đánh giá độ chính xác tổng thể và độ chính xác của từng lớp cho toàn bộ dữ liệu trong tập test
 - ii. In ra giá trị nhãn dự đoán và thực tế của 7 phần tử đầu tiên trong tập test.
Đánh giá độ chính xác tổng thể 7 phần tử đầu tiên trong tập test.
- e. Xây dựng mô hình Bayes thơ ngây, đánh giá độ chính xác tổng thể và độ chính xác cho từng phân lớp
- f. Với nghi thức hold-out (2/3 để học, 1/3 để kiểm tra), so sánh độ chính xác tổng thể của mô hình KNN và Bayes thơ ngây

BÀI TẬP KIỂM TRA BUỔI 2 – KNN & Bayes thơ ngây

- a. Đọc dữ liệu từ tập dữ liệu đánh giá chất lượng rượu vang **trắng** trên trang UCI
<https://archive.ics.uci.edu/ml/datasets/wine+quality>
- b. Tập dữ liệu có bao nhiêu phần tử, Có bao nhiêu nhãn? Ghi chú kết quả trong file code sau câu lệnh.
len(tên biến) : số lượng phần tử
import numpy as np
np.unique(tên biến) : liệt kê các giá trị khác nhau của biến
tên biến.value_counts() : liệt kê số lượng và giá trị khác nhau của biến
- c. Sử dụng **8 phần để** xây dựng mô hình và sử dụng **2 phần** để thực hiện đánh giá mô hình. Anh/chị ghi chú lại số lượng phần tử trong tập test và nhãn của các phần tử thuộc tập test trong file code
- d. Xây dựng mô hình KNN với 7 láng giềng dựa trên tập dữ liệu học tạo ra ở bước c.
 - i. Đánh giá độ chính xác tổng thể và độ chính xác của từng lớp cho toàn bộ dữ liệu trong tập test
 - ii. In ra giá trị nhãn dự đoán và thực tế của 8 phần tử đầu tiên trong tập test. Đánh giá độ chính xác tổng thể **8** phần tử đầu tiên trong tập test.
- e. Xây dựng mô hình Bayes thơ ngây, đánh giá độ chính xác tổng thể và độ chính xác cho từng phân lớp
- f. Với nghi thức hold-out (2/3 để học, 1/3 để kiểm tra), so sánh độ chính xác tổng thể của mô hình KNN và Bayes thơ ngây