Bộ môn An toàn Thông tin – Khoa Mạng máy tính và Truyền thông
Trường Đại học Công nghệ Thông tin – Đại học Quốc gia thành phố Hồ Chí Minh

1

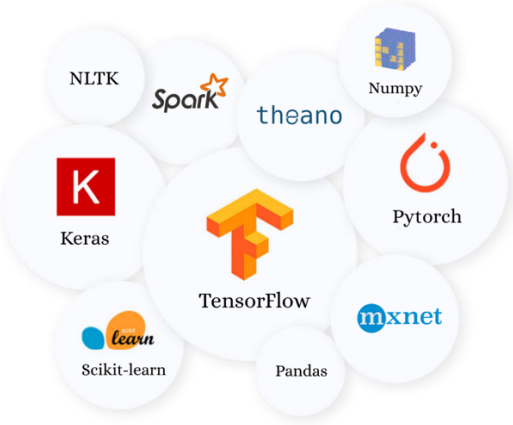
Lab

PHỤC VỤ MỤC ĐÍCH GIÁO DỤC
FOR EDUCATIONAL PURPOSE ONLY

Setting Up Your Machine Learning for Cybersecurity Arsenal

Python

Thực hành môn Phương pháp học máy trong an toàn thông tin



Tháng 3/2025

Lưu hành nội bộ

<Nghiêm cấm đ ăng tải trên internet dưới mọi hình thức>

BEE SECURE
INSECLAB

Mọi góp ý về tài liệu, vui lòng gửi về email inseclab@uit.edu.vn

A. TỔNG QUAN

1. Mục tiêu

- Giới thiệu môi trường,
- Các xử lí ma trận numpy, sklearn, Pytorch/Keras,
- Biểu diễn dữ liệu Matplotlib, trích xuất thuộc tính, Anaconda/Jupyter Notebook
- Đánh giá mô hình theo các tiêu chí cơ bản

2. Thời gian thực hành

- Thực hành tại lớp: **5** tiết tại phòng thực hành.
- Hoàn thành báo cáo kết quả thực hành: tối đa **7** ngày.

3. Môi trường thực hành

Google Colab (https://research.google.com/colaboratory/) và thư mục tải Datasets

B. THỰC HÀNH

1. Numpy

Numpy là thư viện quan trọng trong máy học. Nhờ API từ thư viện ta có thể xây dựng các thuật toán và công cụ cho máy học từ đầu

a) Mảng đa chiều numpy

Numpy hỗ trợ cá phép tínhh như đại số tuyến tính và ma trận. Mảng đa chiều trong numpy gọi là ndarrays, sử lý của ndarrays nhanh gấp 25 so với dùng vòng lặp for.
Ví dụ về định nghĩa Numpy object:

```
import numpy as np
np_array = np.array( [0, 1, 2, 3] )
# Creating an array with ten elements initialized as zero
np_zero_array = np.zeros(10)
```

b) Cách thức hoạt động ma trân numpy

Thư viện numpy cung cấp hàm dot () để tính tích của hai ma trận

```
import numpy as np

a = np.array([-8, 15])

X = np.array([[1, 5],
              [3, 4],
              [2, 3]])
```

```
y = np.dot(X, a)
```

® **Bài tập (yêu cầu làm)**

1. *Sinh viên cho ví vụ về phép cộng, trừ hai ma trận numpy.*

c) Triển khai một công cụ dự đoán đơn giản bằng numpy

Để hiểu về việc sử dụng phương thức dot() trong numpy cho phép nhân hai ma trận. Nên ta thử triển khai một công cụ dự đoán các giá trị trong tương lai từ một tập nhiều đầu vào và trên trọng số tương đối, bằng cách ứng dụng tích giữa ma trận và vector.

```
import numpy as np

def predict(data, w):
    return data.dot(w)


# w is the vector of weights
w = np.array([0.1, 0.2, 0.3])

# matrices as input datasets
data1 = np.array([0.3, 1.5, 2.8])

data2 = np.array([0.5, 0.4, 0.9])

data3 = np.array([2.3, 3.1, 0.5])


data_in = np.array([data1[0],data2[0],data3[0]])

print('Predicted value: $%.2f' %  predict(data_in, w) )
```

2. Scikit-learn

Thư viện scikit-learning cung cấp một loạt các mô hình và thuật toán có thể dễ dàng sử dụng lại trong quá trình phát triển các giải pháp tùy chỉnh, sử dụng các phương pháp và dự đoán, bao gồm những điều sau:

- Phân loại - Classification
- Hồi quy - Regression
- Giảm kích thước - Dimensionality reduction
- Phân cụm – Clustering

scikit-learning cũng cung cấp các mô-đun sẵn có sử dụng cho phép thực hiện các tác vụ sau:

- Data preprocessing – xử lý dữ liệu
- Feature extraction – trích xuất thuộc tính
- Hyperparameter optimization - tối ưu siêu tham số
- Model evaluation – đánh giá mô hình

Ví dụ: Sử dụng các mẫu phân tích dự đoán có sẵn trong scikit-learn, dữ liệu được huấn luyện (ma trận X), sử dụng mô hình linear regression để dự đoán dựa trên vector trọng số y.

Sử dụng phương thức fit() và predict() được triển khai trong class LinearRegression

```
import numpy as np
from sklearn.linear_model import LinearRegression

# X is a matrix that represents the training dataset
# y is a vector of weights, to be associated with input dataset

X = np.array([[3], [5], [7], [9], [11]]).reshape(-1, 1)
y = [8.0, 9.1, 10.3, 11.4, 12.6]

lreg_model = LinearRegression()
lreg_model.fit(X, y)

# New data (unseen before)
new_data = np.array([[13]])

print('Model Prediction for new data: $%.2f' %
lreg_model.predict(new_data)[0]  )
```

3. Matplotlib

Thư viện hỗ trợ biểu diễn dữ liệu bằng đồ họ giúp có cách nhìn dữ liệu trực quan. Matplotlib là công cụ vẽ biểu đồ dữ liệu lấy cảm hứng từ MATLAB.

Ví dụ: sử dụng phương thức plot() đầu vào thu được bằng phương thức arange() của thư viện numpy

```
import numpy as np

import matplotlib.pyplot as plt

plt.plot(np.arange(15), np.arange(15))

plt.show()
```

4. Pandas

Thư viên pandas giúp xử lý cấu trúc dữ liệuu DataFrame – dạng bảng, các cột đại diện cho thuộc tính, và hang đại diện có giá trị.

```
import pandas as pd
from sklearn import datasets

iris = datasets.load_iris()
iris_df = pd.DataFrame(iris.data, columns = iris.feature_names)

iris_df.head()
iris_df.describe()
```

Sinh viên tự tạo tập tin CSV có cấu trúc như sau:

	X	Y	Z
1	78	84	86
2	85	94	97
3	80	83	73
4	96	94	96
5	86	86	83

® Bài tập (yêu cầu làm)

2. *Sinh viên sử dụng pandas xử lý các yêu cầu sau:*
   - *Đọc CSV thành Dataframe và hiển thị*
   - *Hãy chuyển index mặc định thành giá trị cột id*
   - *Sắp xếp dữ liễu theo nhiều cột (sort)*
   - *Chọn một cột cụ thể và hiển thị nó*
   - *Chọn 2 hàng đầu tiền và hiển thị chúng*
   - *Hãy chọ một hàng dựa trên một điều kiện giá trị của cột*
   - *Thay đổi một vài giá trị thành **NaN** ở CSV, sau đó đọc lên thành Dataframe và thay thế chúng bằng giá trị 0*
   - *Ở cột Z chuyển giá trị lớn hơn 90 là True và nhỏ hơn là False trong Dataframe*
   - *Chuyển Dataframe trên thành 2 Dataframe d1 và d2; d1 chứ cột X và Y, d2 chứa cột Z; cuối cùng d3 là thành quả của nối 2 Dataframe d1 và d2*
   - *Dùng tính năng thống kê hãy hiển thị kết quả thông kể các giá trị thuộc tính của Dataframe*

Bài tập về nhà (yêu cầu làm)

3. *Sinh viên tự tìm hiểu thực hiện lại ví dụ dùng mô hình Linear Regression trong thư viện* scikit-learning bằng các thư viện sau:
   - TensorFlow
   - Keras
   - PyTorch

   Cho biết cảm nghĩ về việc dùng 4 thư viện này

5. Phát hiện các mối đe dọa an ninh mạng email bằng ML

d) Phát hiện spam với Perceptrons

Việc áp dụng Neural Networks (NNs) phát hiện spam là cách đơn giản nhật.
Hầu hến dữ liệu tấn công bằng email rất lớn cần có các công cụ như máy học để phát hiện đâu là ham đâu là spam.
Sử dụng thư viện scikit-learn để tạo bộ lọc spam mail dựa trên Perceptron.
Tập dữ liệu spam mail chúng ta sử dụng là: https://archive.ics.uci.edu/ml/machine-learning-databases/00228/smsspamcollection.zip.

Tập dữ liệu định dạng CSV, ta tiến hành xử lý dữ liệu, chuyển chúng về giá trị số để Perceptron có hiểu được. Lưu ý chỉ chọn các tin nhắn có từ khoá "buy" và "sex", số lần từ khoá xuất tin trong tin nhắn để chỉ ra là spam hoặc ham.
Thành phần quả trình trên là: sms_spam_perceptron.csv
Tiến hành tải tập tin trên, thông qua thư viện pandas trích xuất Dataframe các giá trị tương ứng thông qua *iloc().*

```
import pandas as pd
import numpy as np
df = pd.read_csv('sms_spam_perceptron.csv')
y = df.iloc[:, 0].values
y = np.where(y == 'spam', -1, 1)
X = df.iloc[:, [1, 2]].values
```

Ta gán nhãn ham và spam cho biến *y* ở cột đầu tiền bằng phương thức *iloc()*, đồng thời chuyển các nhãn thành số -1 (spam) và 1 (ham) bằng cách sử dụng phương thức *where().*
Biến ma trận *X* bao gồm các giá trị hai cột *sex* và *buy*.
Trước khi đưa vào Perceptron, ta chia dữ liệu thành train và test.

```
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)
```

Sử dụng phương thức *train_test_split(),* đưa vào biến *X* và *y,* chia dữ liệu thành 2 phần, 30% dữ liệu gốc (tham số *test_size = 0.3*) là để test, 70% còn lại là dữ liệu train
Khởi tạo lớp Perceptron của gói sklearn.linear_model:

```
from sklearn.linear_model import Perceptron

p = Perceptron(max_iter=40, eta0=0.1, random_state=0)
p.fit(X_train, y_train)
```

Khi khởi tạo biến *p*, ta chọn số lần lập tối đa là 40 (*max_iter = 40*), tốc độ học learning rate (*eta0 = 0.1*). Cuối cùng dung phương fit để đưa dữ liệu vào huấn luyện cho *p*.
Bây giờ ta có thể đánh giá các giá trị của dữ liệu test bằng phương thức *predict().*

```
y_pred = p.predict(X_test)
```

Ta có thể xác minh các đánh giá trên do Perceptron trả về.

```
from sklearn.metrics import accuracy_score
print('Misclassified samples: %d' % (y_test != y_pred).sum())
print('Accuracy: %.2f' % accuracy_score(y_test, y_pred))
```

e) Phát hiện spam với SVMs

SVM là một ví dụ về các thuật toán được giám sát supervised algorithms (cũng như Perceptron).

Tải và phân chia dữ liệu như Perceptron.

Khởi tạo SVM bằng các import class SVC (support vector classifier) từ gói *sklearn.svm*, chọn phân loại linear classifier (kernel = 'linear'), sau đó đào tạo mô hình bằng phương thức *fit()* và đánh giá dữ liệu test bằng phương thức *predict().*

```
from sklearn.svm import SVC

svm = SVC(kernel='linear', C=1.0, random_state=0)
svm.fit(X_train, y_train)
y_pred = svm.predict(X_test)
```

Cuối cùng xác minh các đánh giá như Perceptron.

Bài tập (yêu cầu làm)

4. *Sinh viên hoàn thành code phát hiện spam với* ***SVMs*** *và* ***Linear regression***

6. Phát hiện Phishing với ML

f) Logistic regession

Ta sử dụng tập dữ liệu có sẵn trên UCI machine learning
https://archive.ics.uci.edu/ml/machine-learning-databases/00327/

Sau đó chuyển đổi định dạng từ .arff sang CSV bằng kỹ thuật one-hot encoding (https://en.wikipedia.org/wiki/One-hot) , có chứa 30 thuộc tính phishing websites.

```
import pandas as pd
import numpy as np
from sklearn import *
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

```
phishing_dataset = np.genfromtxt('phishing_dataset.csv', delimiter=',',
dtype=np.int32)
samples = phishing_dataset[:,:-1]
targets = phishing_dataset[:, -1]

from sklearn.model_selection import train_test_split

training_samples, testing_samples, training_targets, testing_targets =
train_test_split(
         samples, targets, test_size=0.2, random_state=0)

log_classifier = LogisticRegression()
log_classifier.fit(training_samples, training_targets)
predictions = log_classifier.predict(testing_samples)

accuracy = 100.0 * accuracy_score(testing_targets, predictions)
print ("Logistic Regression accuracy: " + str(accuracy))
```

Bài tập (yêu cầu làm)

5. *Sinh viên cho biết chức năng của phương thức* ***genfromtxt()*** trong thư viện numpy.

g) Decision trees

```
…
from sklearn import tree tree_classifier = tree.DecisionTreeClassifier()
tree_classifier.fit(training_samples, training_targets)
…
```

Bài tập (yêu cầu làm)

6. *Sinh viên hoàn thiện code Decision trees trên và đánh giá kết quả nhận được so với phương pháp Logistic regression.*

Bài tập về nhà (yêu cầu làm)

7. *Sinh viên thực hiện code phát hiện phising website bằng mô hình học máy Logistic regression và Decision trees với train và test trên tập dữ liệu https://www.kaggle.com/shashwatwork/phishing-dataset-for-machine-learning*
8. *Sinh viên thực hiện code phát hiện phising website bằng mô hình học máy Logistic regression hoặc Decision trees với train và test trên tập dữ liệu phishtank. Tham khảo cách xử lý và trích xuất thuộc tích https://github.com/surajr/URL-Classification*

C. YÊU CẦU & ĐÁNH GIÁ

1. Yêu cầu

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.. Đăng ký nhóm cố định từ buổi 1.
- Sinh viên báo cáo kết quả thực hiện và nộp bài bằng **1 trong 2 hình thức**:

h) Hình thức 1 - Báo cáo chi tiết:

Báo cáo cụ thể quá trình thực hành (có ảnh minh họa các bước) và giải thích các vấn đề kèm theo. Trình bày trong file PDF theo mẫu có sẵn tại website môn học.

i) Hình thức 2 - Video trình bày chi tiết:

Quay lại quá trình thực hiện Lab của sinh viên kèm thuyết minh trực tiếp mô tả và giải thích quá trình thực hành. Upload lên **Youtube** và chèn link vào đầu báo cáo theo mẫu. **Lưu ý:** *Không chia sẻ ở chế độ Public trên Youtube.*

Đặt tên file báo cáo theo định dạng như mẫu:

[Mã lớp]-LabX-TeamX_MSSV1 _MSSV2

Ví dụ: *[NT522.P21.ANTT.1]-Lab1-TeamX_23520000_23520999.*

- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Nộp báo cáo trên theo thời gian đã thống nhất tại website môn học.

2. Đánh giá:

- Sinh viên hiểu và tự thực hiện được bài thực hành, đóng góp tích cực tại lớp.
- Báo cáo trình bày chi tiết, giải thích các bước thực hiện và chứng minh được do nhóm sinh viên thực hiện.

- Hoàn tất nội dung cơ bản và có thực hiện nội dung *mở rộng – cộng điểm* (với lớp **ANTN**).

Kết quả thực hành cũng được đánh giá bằng kiểm tra kết quả trực tiếp tại lớp vào cuối buổi thực hành hoặc vào buổi thực hành thứ 2.

Lưu ý: Bài sao chép, nộp trễ, "*gánh team*", … sẽ được xử lý tùy mức độ.

HẾT

Chúc các bạn hoàn thành tốt!