

Nguyễn Khắc Sơn – 21085691

Bài tập về nhà lần 4 (SVM)

Bài 1: Support Vector Machine (SVM) là một trong những thuật toán phân lớp hiệu quả và linh động. Dựa trên những kiến thức đã học trên lớp và đọc thêm tài liệu ở nhà, em hãy trình bày ngắn gọn những hiểu biết của mình về SVM.

Support Vector Machine (SVM)

- Là một thuật toán học máy thuộc nhóm học có giám sát, thường được sử dụng cho các bài toán phân loại và hồi quy.
- Mục tiêu chính của SVM là tìm ra đường thẳng (hoặc một mặt phẳng trong không gian nhiều chiều) để phân tách dữ liệu thuộc các lớp khác nhau sao cho khoảng cách từ các điểm gần nhất của các lớp đến đường phân tách này là lớn nhất.
- Đường phân tách này được gọi là siêu phẳng (*hyperplane*), và các điểm dữ liệu gần nhất gọi là vector hỗ trợ (*support vectors*).

Ví dụ minh họa:

Có 1 tập dữ liệu về chiều cao và cân nặng của hai nhóm người: **khỏe mạnh** và **không khỏe mạnh**. Các dữ liệu này được biểu diễn trên trục X-Y tương ứng với chiều cao – cân nặng.

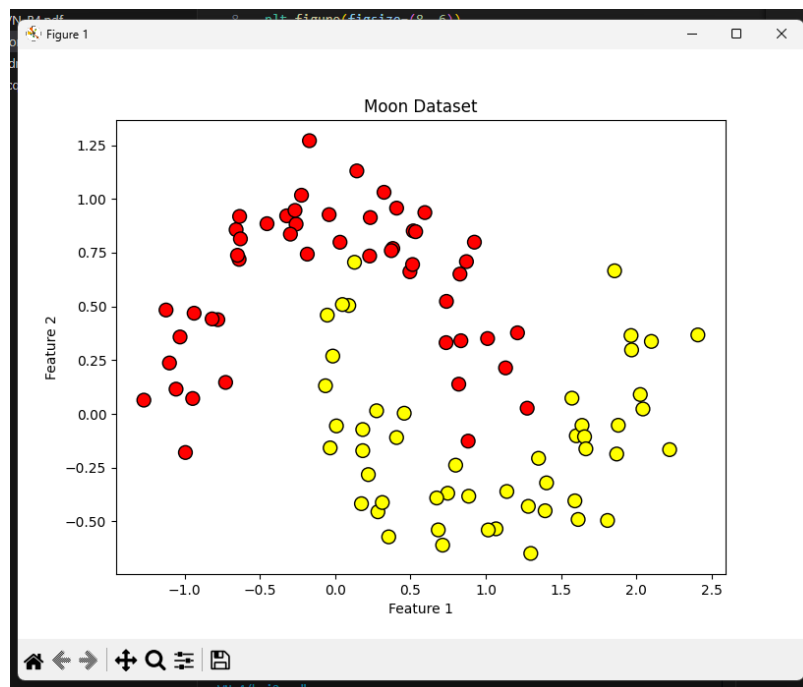
- Thuật toán SVM sẽ tìm ra đường thẳng tốt nhất để phân lớp 2 đối tượng này. Bằng cách tối đa hóa khoảng cách giữa các điểm gần nhất của 2 đối tượng đến đường thẳng phân lớp đó.
- Khi xuất hiện người mới với chiều cao và cân nặng đã đo được, SVM sẽ dùng đường thẳng phân lớp để xác định đối tượng mới thuộc 1 trong 2 đối tượng trên.

Bài 2¹: (Thực hành với thư viện scikit-learn)

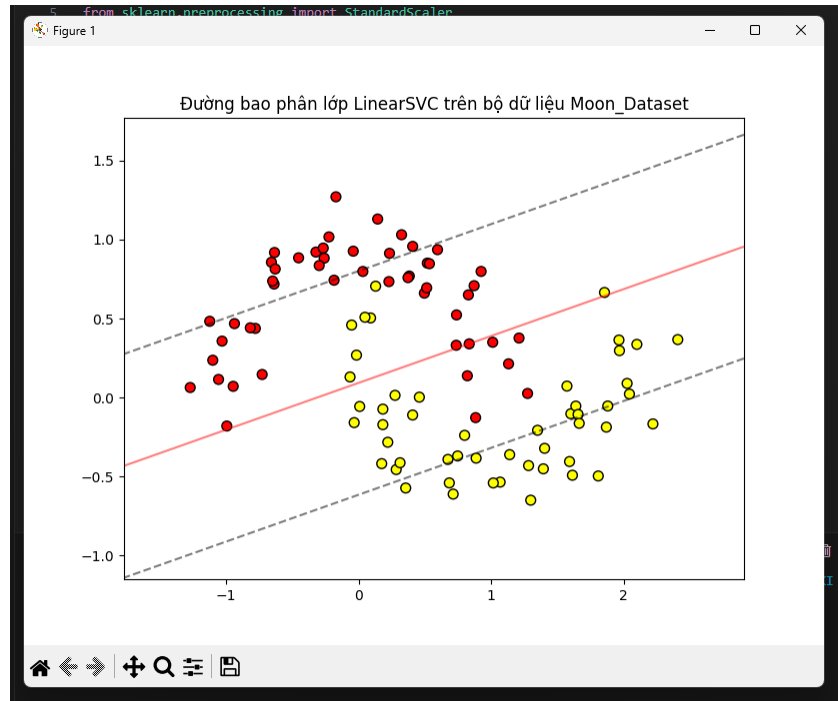
Thư viện scikit-learn cung cấp các lớp như SVC, NuSVC và LinearSVC có khả năng thực hiện phân loại nhị phân và đa lớp trên một tập dữ liệu. Moons dataset2 là một tập dữ liệu (toy dataset) thường được dùng để giúp trực quan hóa các thuật toán phân lớp. Em hãy lựa chọn một trong các lớp của thư viện scikit-learn để lập trình giải bài toán phân lớp sử dụng moon dataset. Đầu ra có thể vẽ đường bao phân lớp để giúp trực quan hóa việc phân lớp cũng như ưu điểm của thuật toán mà em đã thực hiện.

Giải:

Em chọn lớp LinearSVC để lập trình bài toán phân lớp với tập dữ liệu moon dataset.



Hình 1: Moon dataset (bộ dữ liệu có hình trăng khuyết)



Hình 2: Kẻ đường bao trên tập dữ liệu moon_dataset

Moon dataset:

Bộ dữ liệu gồm hai lớp, mỗi lớp có dạng một hình cung. Dữ liệu có thể được tạo ra với hoặc không có nhiễu, làm cho việc phân loại trở nên khó khăn hơn.

Đặc điểm: Không tuyến tính và phù hợp cho bài toán phân loại

Code thực hiện tạo bộ dữ liệu moon dataset và huấn luyện mô hình LinearSVC, sau đó vẽ đường bao phân lớp để trực quan hóa dữ liệu:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_moons
from sklearn.svm import LinearSVC
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline

# 1. Tạo bộ dữ liệu moon dataset
X, y = make_moons(n_samples=100, noise=0.15, random_state=42)

# 2. Tạo và huấn luyện mô hình LinearSVC
model = make_pipeline(StandardScaler(), LinearSVC(random_state=42))
model.fit(X, y)
```

```
# 3. Vẽ đường bao phân lớp
def plot_decision_boundary(model, X, y, ax=None, plot_support=True):
    if ax is None:
        ax = plt.gca()
    xlim = [X[:, 0].min() - 0.5, X[:, 0].max() + 0.5]
    ylim = [X[:, 1].min() - 0.5, X[:, 1].max() + 0.5]

    xx, yy = np.meshgrid(np.linspace(xlim[0], xlim[1], 500),
                          np.linspace(ylim[0], ylim[1], 500))
    Z = model.decision_function(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)

    ax.contour(xx, yy, Z, colors=['black', 'red', 'black'], levels=[-1, 0, 1], alpha=0.5,
               linestyle=['--', '-', '--'])

    ax.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn', edgecolors='k')

# 4. Vẽ đồ thị
plt.figure(figsize=(8, 6))
plot_decision_boundary(model, X, y)
plt.title('Đường bao phân lớp LinearSVC trên bộ dữ liệu Moon_Dataset')
plt.show()
```

Bài 3: (Giải bài toán bằng cách lập trình)

Sử dụng Python, lập một chương trình để giải quyết bài toán phân lớp trong ngôi nhà thông minh đã làm trên lớp với tập dữ liệu huấn luyện 14 mẫu.

Người dùng **nhập** vào giá trị một mẫu mới: (21°C, 52%). Chương trình **in** ra kết quả phân lớp và **vẽ** hình minh họa kết quả.

Trên hình minh họa kết quả, các em vẽ các điểm trong tập dữ liệu huấn luyện với:

- màu xanh nước biển + hình tròn cho **Bedroom**;
- màu đỏ + hình vuông cho **Living room**
- mẫu mới dùng màu xanh lá cây (+ hình vuông hoặc hình tròn, tùy thuộc vào kết quả phân lớp).

Giải:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```

import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier

# Dữ liệu mẫu [x, y] với x là nhiệt độ, y là độ ẩm
data = np.array([
    [22, 50], [24, 45], [23, 47], [21, 49], [25, 44],
    [19, 55], [20, 60], [18, 58], [17, 57], [20, 62],
    [23, 46], [24, 48], [19, 54], [18, 56]
])

# Nhãn tương ứng với các kiểu phòng
labels = np.array([
    "Living Room", "Living Room", "Living Room", "Living Room", "Living Room",
    "Bedroom", "Bedroom", "Bedroom", "Bedroom", "Bedroom",
    "Living Room", "Living Room", "Bedroom", "Bedroom"
])

# Tạo DataFrame cho Seaborn
df = pd.DataFrame(data, columns=["Nhiệt độ", "Độ ẩm"])
df["Loại phòng"] = labels

# Tạo mô hình phân loại KNN
X = df[["Nhiệt độ", "Độ ẩm"]]
y = df["Loại phòng"]
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X, y)

# Nhập mẫu mới
new_temp = float(input("Nhập nhiệt độ của mẫu mới (°C): "))
new_hum = float(input("Nhập độ ẩm của mẫu mới (%): "))
new_sample = np.array([new_temp, new_hum])

# Dự đoán kiểu phòng của mẫu mới
label_dudoan = knn.predict(new_sample)[0]
print(f"Kiểu phòng của mẫu mới (Nhiệt độ {new_temp}°C, Độ ẩm {new_hum}%): {label_dudoan}")

# Tạo biểu đồ scatter với Seaborn
plt.figure(figsize=(10, 10))
sns.scatterplot(x="Nhiệt độ", y="Độ ẩm", hue="Loại phòng", data=df, s=100,
    palette={"Bedroom": "blue", "Living Room": "red"}, style="Loại phòng",
    markers={"Bedroom": "o", "Living Room": "s"})

# Đánh dấu mẫu mới trên biểu đồ

```

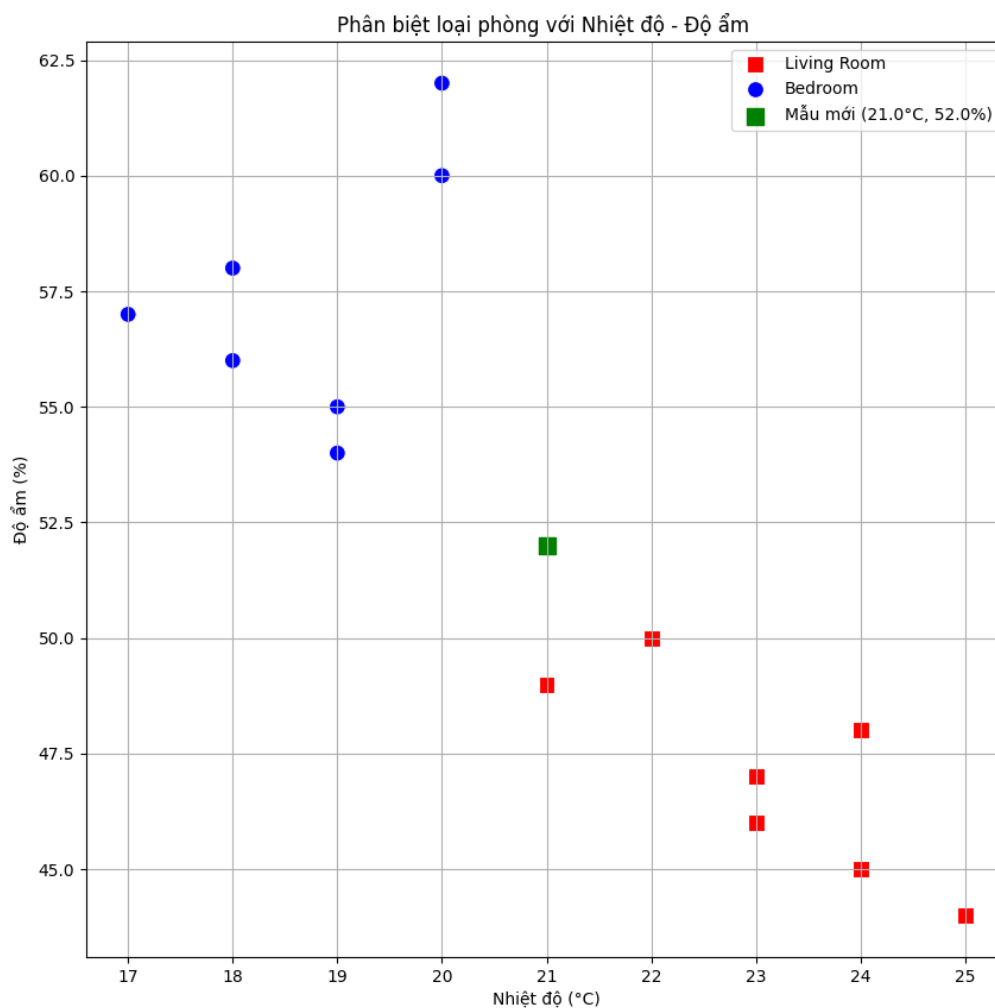
```

marker = 's' if label_dudoan == "Living Room" else 'o'
color = 'green'
plt.scatter(new_temp, new_hum, color=color, s=100, marker=marker, label=f"Mẫu mới
({new_temp}°C, {new_hum}%)")

# Cấu hình biểu đồ
plt.title("Phân biệt loại phòng với Nhiệt độ - Độ ẩm ")
plt.xlabel("Nhiệt độ (°C)")
plt.ylabel("Độ ẩm (%)")
plt.legend()
plt.grid(True)

# Hiển thị biểu đồ
plt.show()

```



Hình 3: Hình ảnh hiển thị dữ liệu mới, được phân vào loại phòng Living Room (Hình vuông, màu xanh lá)