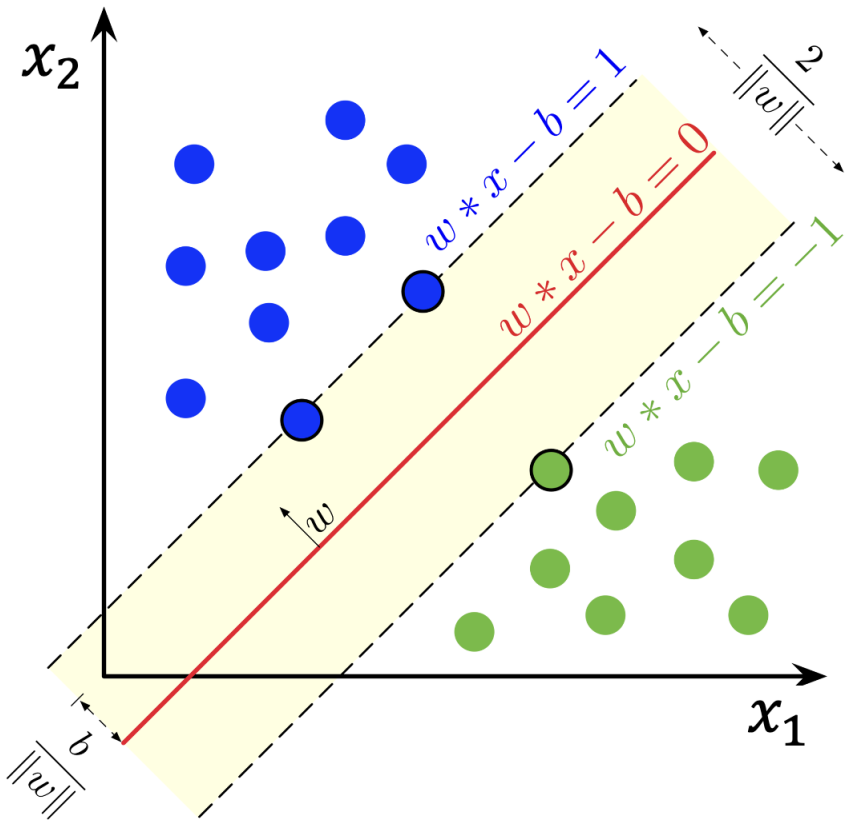
AI VIET NAM – COURSE 2023

Support Vector Machine - Exercise

Ngày 17 tháng 9 năm 2023

Phần I: Giới thiệu

Support Vector Machine (SVM) là một trong các thuật toán phổ biến nhất trong Machine Learning, có thể được sử dụng để thực hiện cả hai dạng bài toán là Regression và Classification. Theo đó, ý tưởng của SVM là tìm một mặt phẳng hoặc đường biên sao cho khoảng cách từ nó đến các điểm dữ liệu gần nhất (gọi là support vectors) là lớn nhất. Bằng cách này, SVM có thể phân chia các điểm dữ liệu trong tập dữ liệu training thành các nhóm (classes). Khi có một điểm dữ liệu mới, SVM sẽ dựa vào vị trí của nó so với mặt phẳng để xác định phân lớp tương ứng.



Trong bài tập này, chúng ta sẽ ôn tập các khái niệm cơ bản của SVM cũng như thực hành ứng dụng SVM vào một số bài toán khác nhau.

Phần II: Bài tập

A. Phần lập trình

Trong phần này, chúng ta sẽ thực hành ứng dụng SVM để xây dựng mô hình dự đoán trên hai tập dữ liệu về Regression và Classification sử dụng thư viện scikit-learn (sklearn) kèm một số thư viện hỗ trợ khác.

- **Support Vector Classifier (SVC):**

  1. **Tải bộ dữ liệu:** Các bạn tải bộ dữ liệu **breast-cancer.csv** tại đây.
  2. **Import các thư viện cần thiết:**

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.svm import SVC
from sklearn.preprocessing import (
    StandardScaler,
    LabelEncoder,
    OneHotEncoder,
    OrdinalEncoder
)
from sklearn.compose import ColumnTransformer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

  3. **Đọc dữ liệu:** Sử dụng pandas, ta đọc dữ liệu từ file .csv lên như sau:

```python
dataset_path = './breast-cancer.csv'
df = pd.read_csv(
    dataset_path,
    names=[
        'age',
        'meonpause',
        'tumor-size',
        'inv-nodes',
        'node-caps',
        'deg-malig',
        'breast',
        'breast-quad',
        'irradiat',
        'label'
    ]
)
```

     Sau khi đọc, DataFrame của chúng ta sẽ có dạng như sau:

	age	meonpause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	label
0	'40-49'	'premeno'	'15-19'	'0-2'	'yes'	'3'	'right'	'left_up'	'no'	'recurrence-events'
1	'50-59'	'ge40'	'15-19'	'0-2'	'no'	'1'	'right'	'central'	'no'	'no-recurrence-events'
2	'50-59'	'ge40'	'35-39'	'0-2'	'no'	'2'	'left'	'left_low'	'no'	'recurrence-events'
3	'40-49'	'premeno'	'35-39'	'0-2'	'yes'	'3'	'right'	'left_low'	'yes'	'no-recurrence-events'
4	'40-49'	'premeno'	'30-34'	'3-5'	'yes'	'2'	'left'	'right_up'	'no'	'recurrence-events'

Hình 1: Các hàng dữ liệu đầu tiên trong bộ dữ liệu breast-cacncer.csv

Bên cạnh đó, ta cũng có thể kiểm tra một số thông tin khác của bộ dữ liệu sử dụng `df.info()`, đạt kết quả như sau:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 286 entries, 0 to 285
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   age          286 non-null    object
 1   meonpause    286 non-null    object
 2   tumor-size   286 non-null    object
 3   inv-nodes    286 non-null    object
 4   node-caps    278 non-null    object
 5   deg-malig    286 non-null    object
 6   breast       286 non-null    object
 7   breast-quad  285 non-null    object
 8   irradiat     286 non-null    object
 9   label        286 non-null    object
dtypes: object(10)
memory usage: 22.5+ KB
```

Hình 2: Một số thông tin khác về bộ dữ liệu

4. **Tiền xử lý dữ liệu:** Quan sát bộ dữ liệu trên, có thể nhận ra bộ dữ liệu hiện tại có nhiều vấn đề cần chúng ta giải quyết trước khi đưa vào huấn luyện. Trong đó, dễ thấy chúng ta cần phải mã hóa các cột dữ liệu dạng categorical và giải quyết các missing values. Các bước thực hiện như sau:

   (a) **Filling missing values:** Từ kết quả `df.info()`, ta có thể thấy các cột thuộc tính **node-caps** và **breast-quad** đang không có đủ 286 non-null values, đồng nghĩa rằng hai cột này đang chứa missing values. Việc giải quyết vấn đề này sẽ có rất nhiều cách, song để đơn giản hóa vấn đề, chúng ta sẽ áp dụng chiến lược khỏa lấp các missing values bằng giá trị xuất hiện nhiều nhất trong cột tương ứng. Ta thực hiện như sau:

```
df['node-caps'] = df['node-caps'].fillna(df['node-caps'].mode()[0])
df['breast-quad'] = df['breast-quad'].fillna(df['breast-quad'].mode()[0])
```

   Ở đoạn code trên, ta sử dụng hàm mode() (các bạn có thể đọc thêm về hàm này tại đây) để tìm ra giá trị xuất hiện nhiều nhất, sau đó sử dụng hàm fillna() để gán giá trị này vào các ô missing.

   (b) **Encode categorical features:** Toàn bộ các cột thuộc tính của bộ dữ liệu đều ở dạng categorical. Vì vậy, ta cần mã hóa chúng thành dạng số trước khi đưa vào huấn luyện mô hình. Để thực hành các kiểu encode khác nhau, trong bài này chúng ta sẽ sử dụng cả

hai OneHotEncoder() và OrdinalEncoder() cho một số cột thuộc tính khác nhau. Đầu tiên, ta tách các cột features và label thành hai biến riêng:

```
y = df['label']
X = df.drop('label', axis=1)
```

Tiếp đến, ta xác định tên các cột sẽ sử dụng OrdinalEncoder() và các cột sẽ sử dụng OneHotEncoder():

```
non_rank_features = ['meonpause', 'node-caps', 'breast', 'breast-quad', '
    irradiat']
rank_features = ['age', 'tumor-size', 'inv-nodes', 'deg-malig']
```

Chúng ta sẽ sử dụng đồng thời hai Encoder này thông qua ColumnTransformer() của sklearn như sau:

```
transformer = ColumnTransformer(
    transformers=[
        ("OneHot", OneHotEncoder(drop='first'), non_rank_features),
        ("Ordinal", OrdinalEncoder(), rank_features)
    ],
    remainder='passthrough'
)
X_transformed = transformer.fit_transform(X)
```

Như vậy ta đã encode xong các thuộc tính. Để có thể quan sát dữ liệu sau khi encode một cách trực quan, ta sẽ biến đổi biến `X_transformed` thành DataFrame như sau:

```
onehot_features = transformer.named_transformers_['OneHot'].
    get_feature_names_out(non_rank_features)
all_features = onehot_features.tolist() + rank_features

X_encoded = pd.DataFrame(
    X_transformed,
    columns=all_features
)
```

Chúng ta có thể quan sát sự thay đổi của bộ dữ liệu thông qua ảnh sau:

	meonpause_'lt40'	meonpause_'premeno'	node-caps_'yes'	breast_'right'	breast-quad_'left_low'	breast-quad_'left_up'	breast-quad_'right_low'	breast-quad_'right_up'
0	0.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0
1	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
3	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0
4	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0

Hình 3: Một phần của bộ dữ liệu sau khi các categorical features đã được encode

(c) **Encode label:** Label của bộ dữ liệu gồm có hai giá trị (class) là **"recurrence-events"** và **"no-recurrence-events"**. Tương tự như categorical features, ta cũng cần phải đưa label này về dạng số. Trong sklearn, ta có LabelEncoder() chuyên dùng để encode các label. Cách sử dụng như sau:

```
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
```

(d) **Normalization:** Cuối cùng, để thuận tiện trong việc training, ta cũng áp dụng kỹ thuật chuẩn hóa dữ liệu vào các thuộc tính X như sau:

```
normalizer = StandardScaler()
X_normalized = normalizer.fit_transform(X_encoded)
```

5. **Chia tập dữ liệu train, val:** Sau khi hoàn tất tiền xử lý, ta bắt đầu phân chia tập dữ liệu ban đầu thành hai tập con. Một tập "train" dùng cho việc huấn luyện mô hình, một tập "val" dùng để đánh giá mô hình sau khi train. Tỉ lệ chia ở đây sẽ là 7:3.

```
test_size = 0.3
random_state = 1
is_shuffle = True
X_train, X_val, y_train, y_val = train_test_split(
    X_normalized, y_encoded,
    test_size=test_size,
    random_state=random_state,
    shuffle=is_shuffle
)
```

6. **Huấn luyện mô hình:** Ta huấn luyện mô hình SVM trên tập train:

```
classifier = SVC(
    random_state=random_state
)
classifier.fit(X_train, y_train)
```

7. **Đánh giá mô hình:** Với mô hình đã huấn luyện, ta sẽ kiểm tra hiệu năng của nó thông qua đánh giá trên tập val:

```
y_pred = classifier.predict(X_val)
scores = accuracy_score(y_pred, y_val)

print('Evaluation results on validation set:')
print(f'Accuracy: {scores}')
```

Kết quả in ra màn hình chính là độ chính xác của mô hình SVM đạt được trên tập val.

- **Support Vector Regression (SVR):**

1. **Tải bộ dữ liệu:** Các bạn tải bộ dữ liệu **auto-insurance.csv** tại đây.
2. **Import các thư viện cần thiết:**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.svm import SVR
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

3. **Đọc bộ dữ liệu:** Sử dụng pandas, các bạn đọc dữ liệu từ file .csv như sau:

```
dataset_path = './auto-insurance.csv'
df = pd.read_csv(
    dataset_path,
    names=[
        'n_claims',
        'total_payment'
    ]
)
```

Sau khi đọc, DataFrame sẽ có dạng như sau:

	n_claims	total_payment
0	108	392.5
1	19	46.2
2	13	15.7
3	124	422.2
4	40	119.4

Hình 4: Một số hàng đầu tiên của bộ dữ liệu auto-insurance.csv

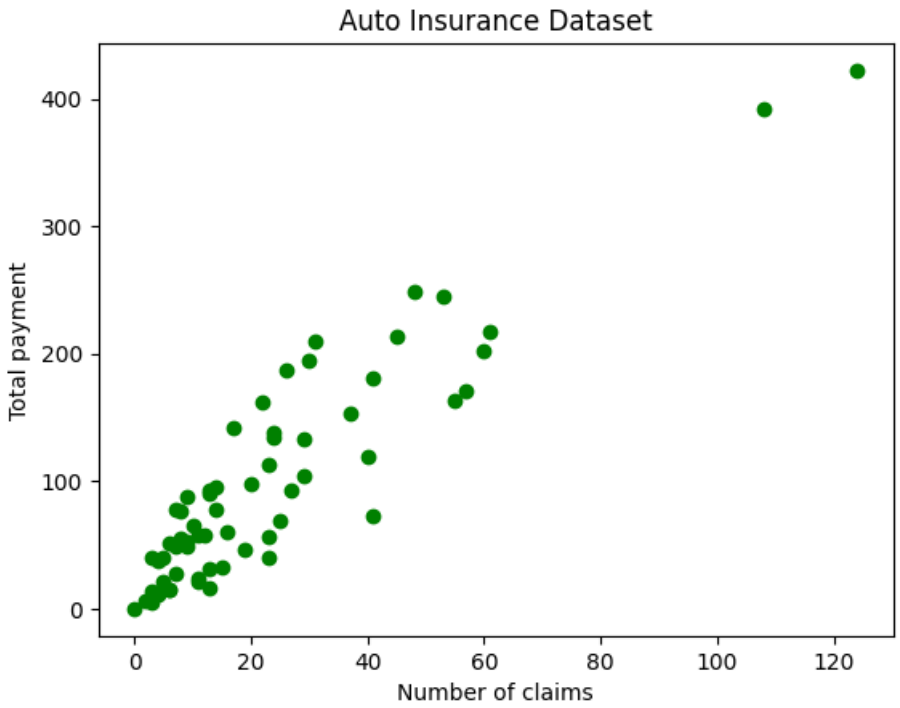
Ta cũng có thể coi một số thông tin khác của bộ dữ liệu thông qua hàm `df.info()`:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63 entries, 0 to 62
Data columns (total 2 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   n_claims       63 non-null     int64
 1   total_payment  63 non-null     float64
dtypes: float64(1), int64(1)
memory usage: 1.1 KB
```

Hình 5: Một số thông tin khác của bộ dữ liệu

Vì bộ dữ liệu chỉ có một thuộc tính, ta hoàn toàn có thể trực quan bộ dữ liệu này lên đồ thị như sau:

```
plt.scatter(df['n_claims'], df['total_payment'], color='green')
plt.title('Auto Insurance Dataset')
plt.xlabel('Number of claims')
plt.ylabel('Total payment')
plt.show()
```



Hình 6: Bộ dữ liệu auto-insurance.csv trên đồ thị

4. **Chuẩn hóa dữ liệu (Normalization):** Đối với bộ dữ liệu này, công việc tiền xử lý trở nên nhẹ nhàng hơn khi ta chỉ cần áp dụng bước chuẩn hóa. Như vậy, ta làm như sau:

```
normalizer = StandardScaler()
df_normalized = normalizer.fit_transform(df)
```

5. **Chia bộ dữ liệu train, val:** Ta tách bộ dữ liệu ban đầu thành hai tập con train và val như sau:

```
X, y = df_normalized[:, 0], df_normalized[:, 1]
X = X.reshape(-1, 1)

test_size = 0.3
random_state = 1
is_shuffle = True
X_train, X_val, y_train, y_val = train_test_split(
    X, y,
    test_size=test_size,
    random_state=random_state,
    shuffle=is_shuffle
)
```

6. **Huấn luyện mô hình:** Ta huấn luyện mô hình SVM trên tập train đã chia:

```
regressor = SVR()
regressor.fit(X_train, y_train)
```

7. **Đánh giá mô hình:** Để kiểm tra độ chính xác của mô hình đã huấn luyện, ta sẽ đánh giá nó trên tập val như sau:

```
y_pred = regressor.predict(X_val)
mae = mean_absolute_error(y_pred, y_val)
mse = mean_squared_error(y_pred, y_val)
```

```
4
5 print('Evaluation results on validation set:')
6 print(f'Mean Absolute Error: {mae}')
7 print(f'Mean Squared Error: {mse}')
```

Kết quả hiển thị trên màn hình là kết quả độ đo MAE và MSE của mô hình SVM trên tập val.

B. Phần trắc nghiệm

1. SVM là viết tắt của từ nào sau đây?

   (*a*) Supervised Vector Machine
   (*b*) Support Vector Machine
   (*c*) Simple Vector Model
   (*d*) Singular Value Method

2. Mô hình SVM thường được sử dụng để giải quyết một cách hiệu quả bài toán nào sau đây?

   (*a*) Speech Recognition
   (*b*) Image Classification
   (*c*) Object Segmentation
   (*d*) Machine Translation

3. Trong Machine Learning, mô hình SVM có thể giải quyết dạng bài toán nào sau đây?

   (*a*) Classification
   (*b*) Regression
   (*c*) Cả (a), (b) đều đúng
   (*d*) Cả (a), (b) đều sai

4. Trong Binary SVM, khái niệm "Decision Boundary" là?

   (*a*) Một đường ngăn cách các điểm dữ liệu thành hai lớp
   (*b*) Điểm dữ liệu có giá trị max
   (*c*) Điểm dữ liệu có giá trị min
   (*d*) Một đường đi qua các điểm dữ liệu

5. Trong SVM, decision boundary còn được gọi là?

   (*a*) Margin
   (*b*) Kernel
   (*c*) Hyperplane
   (*d*) Support Vector

6. Trong SVM, các điểm dữ liệu nằm gần với decision boundary nhất còn được gọi là?

   (*a*) Outliers
   (*b*) Nodes
   (*c*) Neighbors
   (*d*) Support Vectors

7. Với bộ dữ liệu mà các điểm dữ liệu có thể được phân biệt một cách tuyến tính (linearly separable), ta thường sử dụng loại kernel nào cho SVM?

   (*a*) Linear Kernel
   (*b*) Polynomial Kernel
   (*c*) RBF Kernel
   (*d*) Sigmoid Kernel

8. Trong SVM, khái niệm "Margin" là?

   (*a*) Độ chính xác của mô hình
   (*b*) Chiều rộng của decision boundary
   (*c*) Chiều dài của decision boundary
   (*d*) Khoảng cách giữa hai support vectors

9. Trong SVM, khi sử dụng linear kernel, phương trình nào sau đây biểu diễn Margin?

(a) $margin = \frac{1}{\|w\|}$
(b) $margin = \frac{1}{2\|w\|}$
(c) $margin = \frac{\|w\|}{2}$
(d) $margin = \frac{2}{\|w\|}$

10. Trong LinearSVM, phương trình nào sau đây biểu diễn decision boundary?

(a) $y = mx + b$
(b) $y = ax^2 + bx + c$
(c) $w^T x + b = 0$
(d) $e^x = y$

11. Khi tăng giá trị tham số C, mô hình SVM có thể bị rơi vào tình trạng nào sau đây?

(a) Overfitting
(b) Underfitting
(c) Low Bias
(d) Low Variance

12. Mệnh đề nào sau đây là đúng khi nói về điểm mạnh của SVM?

(a) Độ phức tạp tính toán cao
(b) Khả năng xử lý high-dimensional data
(c) Luôn đạt độ chính xác tuyệt đối
(d) Chỉ có thể xử lý linear separable data

13. Mệnh đề nào sau đây là đúng khi miêu tả về Hard SVM?

(a) Tối thiểu mức dự đoán sai
(b) Cho phép dự đoán sai tương đối
(c) Ưu tiên sử dụng trên dữ liệu nhiễu
(d) Đạt hiệu năng tối đa với dữ liệu lớn

14. Mệnh đề nào sau đây là đúng khi miêu tả về Soft SVM?

(a) Tối thiểu mức dự đoán sai
(b) Cho phép dự đoán sai tương đối
(c) Ưu tiên sử dụng trên dữ liệu nhiễu
(d) Đạt hiệu năng tối đa với dữ liệu lớn

15. Trong Hard SVM sử dụng linear kernel, chúng ta cần tối ưu hàm nào sau đây?

(a) $\min \sum_{i=1}^{n} \log(1 + e^{(-y_i(w \cdot x_i + b))})$
(b) $\min \frac{1}{2}\|w\|^2$
(c) $\min \frac{1}{n} \sum_{i=1}^{n} (y_i - (w \cdot x_i + b))^2$
(d) $\max \gamma \|x - x'\|^2$

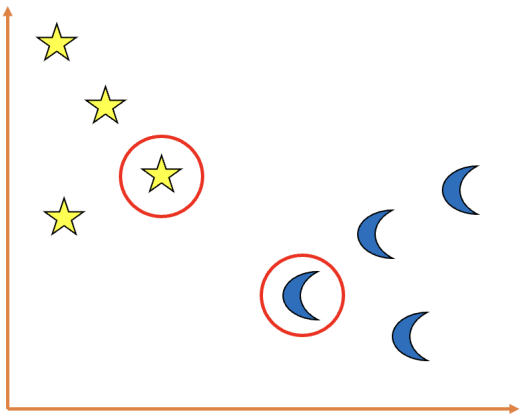
16. Độ chính xác của SVM phụ thuộc vào yếu tố nào sau đây?

(a) Tham số C
(b) Loại kernel
(c) Các tham số của kernel
(d) Tất cả phương án đều đúng

17. Trong Support Vector Regression (SVR), khái niệm "Margin" được gọi là?

(a) Decision Boundary
(b) ϵ-insensitive zone
(c) Hyperplane
(d) Bias zone

18. Cho bộ dữ liệu phân loại nhị phân có ảnh như hình dưới đây:

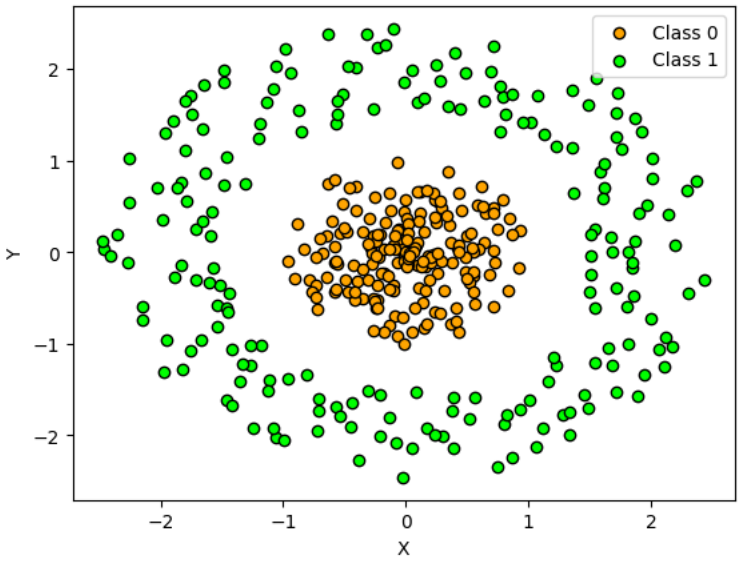


Khi loại bỏ một trong các điểm dữ liệu được khoanh tròn màu đỏ, decision boundary cho bộ dữ liệu trên có thay đổi hay không?

(*a*) Có

(*b*) Không

19. Cho bộ dữ liệu phân loại nhị phân có ảnh như hình dưới đây:



Với phân bố dữ liệu trên, sử dụng kernel nào sau đây là phù hợp nhất cho SVM?

(*a*) Linear Kernel

(*b*) Polynomial kernel

(*c*) RBF Kernel

(*d*) Sigmoid Kernel

20. Cho bộ dữ liệu có các điểm dữ liệu A(1, 2), B(4, 1), C(2, 6), D(3, 5) và một đường thẳng $x + y - 6 = 0$. Giả sử các điểm nằm bên dưới đường thẳng này được xác định thuộc class A. Theo đó, các điểm dữ liệu thuộc class A sẽ là?

(*a*) A, B
(*b*) C, D
(*c*) B, D
(*d*) A, C

21. Dựa vào đoạn code Support Vector Classifier ở phần II.A, độ chính xác mà mô hình sau khi huấn luyện đạt được trên tập val là (làm tròn đến hàng thập phân thứ 2):

(*a*) 0.45
(*b*) 0.54
(*c*) 0.69
(*d*) 0.72

22. Dựa vào đoạn code Support Vector Regression ở phần II.A, kết quả độ đo MSE mà mô hình sau khi huấn luyện đạt được trên tập val là (làm tròn đến hàng thập phân thứ 2):

(*a*) 0.45
(*b*) 0.54
(*c*) 0.69
(*d*) 0.72

- Hết -