Bài Đánh Giá Năng Lực về Máy Học Cơ Bản

Thời gian làm bài: 10 tiếng

Dương Trường Bình, Nguyễn Anh Khôi, Nguyễn Đăng Nhã và Đinh Quang Vinh
System và Latex: Nguyễn Dương Thuận và Nguyễn Đình Tiềm

Các bạn có thể dùng internet để tìm kiếm nhanh thông tin như cách dùng một hàm nào đó.
Các bạn cũng có thể dùng Python editor (như Colab) để thử nhanh hay xác nhận cách dùng một hàm nào đó.
Tất cả có 47 câu và làm trong 10 tiếng.
(Quan trọng) các bạn nhớ lưu và submit bài làm trước deadline.
Chúc các bạn làm bài tốt!

---

1 Text Classification - KNN

Bài toán 1: Phân loại đánh giá phim
Trong phần này, chúng ta sẽ xử lý một tập hợp các câu đánh giá phim và chuyển đổi nó thành định dạng vector bằng 'CountVectorizer'. Sự chuyển đổi này rất cần thiết để chuẩn bị dữ liệu văn bản cho các mô hình Machine Learning. Sau đây là quy trình từng bước chuyển đổi:
Bộ dữ liệu text - corpus
Chúng ta bắt đầu với một bộ data nhỏ các câu đánh giá phim, mà chúng ta gọi là **corpus**. Corpus chứa 6 câu đánh giá khác nhau:

```
import sklearn
from sklearn.neighbors import KNeighborsClassifier
from sklearn.feature_extraction.text import CountVectorizer
import numpy as np
import pandas as pd

corpus = ["a very good movie",
          "this movie is excellent",
          "very good and excellent",
          "the movie is bad",
          "poor and boring plot",
          "bad and boring movie"]
```

Các văn bản	Nhãn
a very good movie	1
this movie is excellent	1
very good and excellent	1
the movie is bad	0
poor and boring plot	0
bad and boring movie	0

Bảng 1: Danh sách văn bản và nhãn của chúng

Tạo đặc trưng

```
# 1. Tạo tập dữ liệu văn bản
corpus = ["a very good movie",
          "this movie is excellent",
          "very good and excellent",
          "the movie is bad",
          "poor and boring plot",
          "very boring and bad plot"]
```

Vectorization

Chúng ta sử dụng CountVectorizer để chuyển đổi dữ liệu văn bản thành một ma trận, trong đó mỗi cột tương ứng với một từ trong bộ từ vựng và mỗi hàng đại diện cho một văn bản. Mỗi giá trị trong ma trận thể hiện số lần từ đó xuất hiện trong văn bản tương ứng.

Ví dụ: Câu đầu tiên là: "a very good movie"được thể hiện bằng vector [0 0 0 0 1 0 1 0 0 0 0 1] tức là đối với câu thứ nhất, có 1 từ ở index 4, 6, 11 trong bộ từ vựng vocab.

```
# 2. Biến đổi văn bản thành dạng ma trận tần suất từ
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(corpus)

# 3. Tạo nhãn dữ liệu
y_data = np.array([1, 1, 1, 0, 0, 0])
```

Câu hỏi 1: Biết bộ từ điển được sắp xếp theo thứ tự như sau:

and	bad	boring	excellent	good	is	movie	plot	poor	the	this	very

Vector đặc trưng khi nhận đầu vào là "movie is very bad"là?

A. `[0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0]`

B. `[0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1]`

C. `[0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1]`

D. `[0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1]`

Đáp án đúng: D

Câu hỏi 2: Hãy tính khoảng cách Euclid giữa câu *"movie is very bad"* (bạn vừa thực hiện ở câu trước) và câu *"this movie is excellent"* trong tập văn bản đã được biến đổi thành ma trận đặc trưng. Khoảng cách Euclid giữa hai câu này bằng?

A. 1.73

B. 2.0

C. 2.45

D. 1.41

Đáp án đúng: B

Câu hỏi 3: Top 3 câu có khoảng cách gần nhất với câu *"movie is very bad"* là?

A. *"a very good movie", "this movie is excellent", "very good and excellent"*

B. *"poor and bad movie", "poor and boring plot", "bad and boring movie"*

C. *"this movie is excellent", "a very good movie", "the movie is bad"*

D. *"poor and boring plot", "the movie is bad", "poor and bad movie"*

Đáp án đúng: C

Câu hỏi 4: Nếu sử dụng KNN với k = 5, với đầu vào là *"movie is very bad"* thì KNN sẽ cho kết quả là?

A. 1

B. 0

C. 1.41

D. 3

Đáp án đúng: A

Bài toán 2: Breast Cancer Wisconsin Diagnostic - Tập Dữ Liệu Chẩn Đoán Ung Thư Vú

Tập dữ liệu này chứa thông tin về các đặc trưng liên quan đến khối u ung thư vú, cùng với chẩn đoán của chúng. (Đây là dữ liệu đã được cắt vớt và chuẩn hóa)

Mô Tả Dữ Liệu

Chỉ số	perimeter_mean	area_mean	compactness_mean	diagnosis
0	0.32	0.20	0.10	B
1	0.13	0.06	0.20	B
2	0.31	0.19	0.10	B
4	0.58	0.43	0.47	M
5	0.56	0.36	0.73	M
6	0.49	0.36	0.16	M

Các Đặc Trưng

- **perimeter_mean**: Trung bình chu vi của các khối u.
- **area_mean**: Trung bình diện tích của các khối u.
- **compactness_mean**: Trung bình tính chất đặc của các khối u, được tính bằng (chu vi^2/diện tích - 1.0).
- **diagnosis**: Nhãn lớp cho biết chẩn đoán của khối u, trong đó 'B' đại diện cho lành tính và 'M' đại diện cho ác tính.

```
!gdown 1-2NZ1j0P8AOZ0WWOn9XHAoSlGW9KFrO-
df = pd.read_csv('/content/final_dataset.csv')
X_train = df[["perimeter_mean", "area_mean", "compactness_mean"]].values.tolist()
labels = df['diagnosis'].values.tolist()
```

Câu hỏi 5: Đoạn mã bên dưới chuyển đổi lớp 'B' và lớp 'M' lần lượt thành?

```
y_train = []

for label in labels:
  if label == 'B':
    y_train.append(0)
  else:
    y_train.append(1)

print(type(y_train))
y_train
```

A. Lớp 'B' = 1 và Lớp 'M' = 0

B. Lớp 'B' = 0 và Lớp 'M' = 1

C. Lớp 'B' = [0, 0, 0, 0] và Lớp 'M' = [1, 1, 1, 1]

D. Lớp 'B' = [1, 1, 1, 1] và Lớp 'M' = [0, 0, 0, 0]

Đáp án: B

Câu hỏi 6: Kết quả của mã nguồn bên dưới được hiểu như thế nào?

```
train_data = zip(X_train, y_train)
train_data = list(train_data)
train_data



%aivietnam
```

```
================= Output =================
Vector đại diện cho bộ corpus sau khi
    vectorization:
[([0.32, 0.2, 0.1], 0),
 ([0.13, 0.06, 0.2], 0),
 ([0.31, 0.19, 0.1], 0),
 ([0.24, 0.14, 0.06], 0),
 ([0.21, 0.12, 0.13], 0),
 ([0.58, 0.43, 0.47], 1),
 ([0.56, 0.36, 0.73], 1),
 ([0.49, 0.36, 0.16], 1),
 ([0.6, 0.44, 0.43], 1),
 ([0.54, 0.4, 0.31], 1)]
==========================================
```

A. (Khoảng cách, Lớp)

B. (Lớp, Khoảng cách)

C. ([perimeter_mean, area_mean, compactness_mean], Lớp)

D. (Lớp, [perimeter_mean, area_mean, compactness_mean])

Đáp án: C

Câu hỏi 7: Cho input x=[0.3, 0.3, 0.3]. Tính khoảng cách Manhattan từ x tới 3 điểm dữ liệu đầu tiên (3 dòng đầu trong bộ dữ liệu train) là?

A. 0, 0, 0

B. 0.32, 0.46, 0.44

C. 0.51, 0.32, 0.46

D. 0.32, 0.51, 0.32

Đáp án: D

Câu hỏi 8: Sắp xếp theo thứ tự tăng dần khoảng cách, nếu K=1 thì KNN sẽ trả về kết quả là ?

A. 0.32

B. 1

C. 0

D. 0.75

Đáp án: A

Câu hỏi 9: Sắp xếp theo thứ tự tăng dần khoảng cách, nếu K=5, có tổng cộng bao nhiêu Class 0 và 1?

A. Class 0: 1 và Class 1: 4

B. Class 0: 2 và Class 1: 3

C. Class 0: 3 và Class 1: 2

D. Class 0: 4 và Class 1: 1

Đáp án: C

2 Text Classification - KMean

Bài toán 1: Phân cụm đánh giá phim
Trong phần này, chúng ta sẽ xử lý một tập hợp các câu đánh giá phim và chuyển đổi nó thành định dạng vector bằng `CountVectorizer`. Sự chuyển đổi này rất cần thiết để chuẩn bị dữ liệu văn bản cho các mô hình Machine Learning. Sau đây là quy trình từng bước chuyển đổi:
Bộ dữ liệu text ban đầu - corpus
Chúng ta bắt đầu với một bộ data nhỏ các câu đánh giá phim, mà chúng ta gọi là **corpus**. Corpus chứa 6 câu đánh giá khác nhau:

```
from sklearn.feature_extraction.text import CountVectorizer
import numpy as np

corpus = ["a very good movie",
          "this movie is excellent",
          "very good and excellent",
          "the movie is bad",
          "poor and boring plot",
          "bad and boring movie"]
vectorizer = CountVectorizer()
```

Vectorization

Chúng ta sử dụng `CountVectorizer` để chuyển đổi dữ liệu văn bản thành một ma trận, trong đó mỗi cột tương ứng với một từ trong bộ từ vựng và mỗi hàng đại diện cho một văn bản. Mỗi giá trị trong ma trận thể hiện số lần từ đó xuất hiện trong văn bản tương ứng.

Ví dụ: Câu đầu tiên là: "a very good movie"được thể hiện bằng vector `[0 0 0 0 1 0 1 0 0 0 0 1]` tức là đối với câu thứ nhất, có 1 từ ở index 4, 6, 11 trong bộ từ vựng vocab.

```
X = vectorizer.fit_transform(corpus)
print(f"Bộ từ vựng xây dựng từ corpus: {dict(sorted(vectorizer.vocabulary_.items()))}"
    )
```

```
======================================== Output ========================================
Bộ từ vựng xây dựng từ corpus: {'and': 0, 'bad': 1, 'boring': 2, 'excellent': 3, 'good
    ': 4, 'is': 5, 'movie': 6, 'plot': 7, 'poor': 8, 'the': 9, 'this': 10, 'very': 11}

========================================================================================
```

Chú ý: Ở đây, bộ từ vựng không hề có từ "a", hãy xem rằng từ "a"không mang lại nhiều ý nghĩa ở trong các câu trên.

```
X = X.toarray()
print("Vector đại diện cho bộ corpus sau
    khi vectorization: ")
print(X)




%aivietnam
```

```
================= Output =================
Vector đại diện cho bộ corpus sau khi
    vectorization:
[[0 0 0 0 1 0 1 0 0 0 0 1]
 [0 0 0 1 0 1 1 0 0 0 1 0]
 [1 0 0 1 1 0 0 0 0 0 0 1]
 [0 1 0 0 0 1 1 0 0 1 0 0]
 [1 0 1 0 0 0 0 1 1 0 0 0]
 [1 1 1 0 0 0 1 0 0 0 0 0]]
==========================================
```

Đề bài trắc nghiệm với KMean

Cho tâm cụm là 2 tâm cụm ngẫu nhiên:

- C1 [1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1]: very good and excellent
- C2 [0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0]: the movie is bad

Xác định khoảng cách Euclidean của mỗi tâm cụm tới các sample còn lại (Iteration đầu tiên):

```
C1 = X[2]
C2 = X[3]
C1, C2


%aivietnam
```

```
================= Output =================
(array([1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
    1]),
 array([0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0,
    0]))
==========================================
```

Câu hỏi 10: Khoảng cách Euclidean của C1 tới các sample 0, 1, 4, 5 (làm tròn tới 3 chữ số thập phân) lần lượt là?

A. 1.732, 1.732, 2.499, 2.499

B. 1.732, 1.732, 1.732, 2.499

C. 1.732, 2.499, 2.499, 2.499

D. 2.499, 2.499, 1.732, 1.732

Đáp án: C

Câu hỏi 11: Khoảng cách Euclidean của C2 tới các sample 0, 1, 4, 5 (làm tròn tới 3 chữ số thập phân) lần lượt là?

A. 2.000, 2.236, 2.828, 2.236

B. 2.236, 2.000, 2.828, 2.000

C. 2.236, 2.828, 2.000, 2.828

D. 2.236, 2.828, 2.000, 2.236

Đáp án: B

Câu hỏi 12: Xác định các điểm dữ liệu thuộc cụm 1 và cụm 2: Vector biểu diễn phân cụm của bộ data X là? Ví dụ: Nếu câu đầu tiên thuộc cụm C1, câu thứ 2 thuộc cụm C2 thì vector biểu diễn phân cụm sẽ là `allocation = [C1, C2, ...]`

A. `[C1, C1, C2, C1, C2, C1]`

B. `[C1, C1, C2, C1, C1, C2]`

C. `[C1, C2, C1, C2, C2, C1]`

D. `[C1, C2, C1, C2, C1, C2]`

Đáp án: D

Câu hỏi 13: Cập nhật tâm cụm: Cập nhật tọa độ của tâm cụm thứ 1 và thứ 2 dựa trên các câu được gán cho mỗi tâm cụm (Chọn đáp án gần bằng, làm tròn 1 chữ số thập phân):

A.
```
Cụm C1: [0.7, 0.0 , 0.3, 0.3, 0.6, 0.0 , 0.3, 0.3, 0.3, 0.0 , 0.0 , 0.7]
Cụm C2: [0.3, 0.7, 0.3, 0.3, 0.0 , 0.7, 1.0 , 0.0 , 0.0 , 0.3, 0.3, 0.0 ]
```

B.
```
Cụm C1: [0.7, 0.0, 0.4, 0.4, 0.7, 0.0, 0.3, 0.3, 0.3, 0.1, 0.0, 0.6]
Cụm C2: [0.4, 0.7, 0.2, 0.3, 0.0, 0.6, 0.9, 0.1, 0.0, 0.2, 0.4, 0.0]
```

C.
```
Cụm C1: [0.5, 0.0, 0.4, 0.3, 0.6, 0.0, 0.2, 0.4, 0.3, 0.0, 0.1, 0.7]
Cụm C2: [0.3, 0.5, 0.4, 0.3, 0.0, 0.7, 1.0, 0.0, 0.0, 0.4, 0.2, 0.0]
```

D.
```
Cụm C1: [0.8, 0.0, 0.3, 0.3, 0.5, 0.0, 0.4, 0.2, 0.3, 0.0, 0.0, 0.7]
Cụm C2: [0.3, 0.6, 0.4, 0.2, 0.0, 0.5, 0.8, 0.0, 0.1, 0.3, 0.4, 0.0]
```

Đáp án: A

Câu hỏi 14: Xác định các điểm dữ liệu thuộc cụm 1 và cụm 2 sau khi cập nhật tâm cụm (Theo vector biểu diễn)

A. `[C1, C2, C1, C2, C1, C2]`

B. `[C2, C1, C1, C2, C1, C2]`

C. `[C1, C2, C2, C1, C2, C1]`

D. `[C1, C2, C2, C1, C2, C1]`

Đáp án: A

Bài tập 2 - Tập Dữ Liệu Chẩn Đoán Ung Thư Vú

Tập dữ liệu này chứa thông tin về các đặc trưng liên quan đến khối u ung thư vú, cùng với chẩn đoán của chúng. (Đây là dữ liệu đã được cắt vớt và chuẩn hóa)

Mô Tả Dữ Liệu

Chỉ số	perimeter_mean	area_mean	compactness_mean	diagnosis
0	0.32	0.20	0.10	B
1	0.13	0.06	0.20	B
2	0.31	0.19	0.10	B
3	0.24	0.14	0.06	B
4	0.21	0.12	0.13	B
5	0.58	0.43	0.47	M
6	0.56	0.36	0.73	M
7	0.49	0.36	0.16	M
8	0.60	0.44	0.43	M
9	0.54	0.40	0.31	M

Các Đặc Trưng

- **perimeter_mean**: Trung bình chu vi của các khối u.
- **area_mean**: Trung bình diện tích của các khối u.
- **compactness_mean**: Trung bình tính chất đặc của các khối u, được tính bằng (chu vi^2/diện tích - 1.0).
- **diagnosis**: Nhãn lớp cho biết chẩn đoán của khối u, trong đó 'B' đại diện cho lành tính và 'M' đại diện cho ác tính.

```python
import pandas as pd
import numpy as np

!gdown --id 1-2NZ1j0P8AOZ0WWOn9XHAoSlGW9KFrO-

df = pd.read_csv('/content/final_dataset.csv')
train_data = df[["perimeter_mean", "area_mean", "compactness_mean"]].values
print(train_data)
```

```
================= Output =================
[[0.32 0.2  0.1 ]
 [0.13 0.06 0.2 ]
 [0.31 0.19 0.1 ]
 [0.24 0.14 0.06]
 [0.21 0.12 0.13]
 [0.58 0.43 0.47]
 [0.56 0.36 0.73]
 [0.49 0.36 0.16]
 [0.6  0.44 0.43]
 [0.54 0.4  0.31]]
==========================================
```

Chọn hai tâm cụm:

- C1: [0.24, 0.14, 0.06] (Index 3)
- C2: [0.56, 0.36, 0.73] (Index 6)

```
centroid_values = [train_data[3],
    train_data[6]]
centroid_values
```

```
================= Output =================
[array([0.24, 0.14, 0.06]),
array([0.56, 0.36, 0.73])]
==========================================
```

Hàm tính khoảng cahcs Euclidiean:

```
def euclidean_distance(x1, x2):
    return np.sqrt(np.sum(np.power(x1 - x2, 2)))
```

Câu hỏi 15: Tính tổng khoảng cách của các điểm dữ liệu tới tâm cụm thứ nhất C1: [0.24, 0.14, 0.06] (Đáp án làm tròn tới 2 chữ số thập phân)

A. 2.53

B. 3.46

C. 3.27 Đáp án: C

D. 2.89

Câu hỏi 16: Tính tổng khoảng cách của các điểm dữ liệu tới tâm cụm thứ hai C2: [0.56, 0.36, 0.73] (Đáp án làm tròn tới 2 chữ số thập phân)

A. 5.32

B. 5.23 Đáp án: B

C. 4.57

D. 4.75

Câu hỏi 17: Xác định các điểm dữ liệu thuộc cụm 1 và cụm 2: Vector biểu diễn phân cụm của bộ data_train là? Ví dụ: C1 là 0, C2 là 1, nếu điểm 0 thuộc cụm 2, điểm 1 thuộc cụm 1 thì allocation = [1, 0, ...] allocation = [[0], [1], [2], [3], [4], [5], [6], [7], [8], [9]]

A. `[0, 0, 0, 0, 0, 1, 1, 0, 1, 1]` Đáp án: A

B. `[0, 1, 0, 0, 0, 1, 1, 0, 1, 1]`

C. `[0, 0, 1, 0, 0, 0, 1, 0, 1, 0]`

D. `[0, 1, 0, 1, 0, 1, 0, 0, 1, 1]`

Câu hỏi 18: Cập nhật tọa độ của tâm cụm thứ 1 và thứ 2 dựa trên các điểm dữ liệu được gán cho mỗi tâm cụm trước đó (Chọn đáp án gần bằng, làm tròn 2 chữ số thập phân):

A. C1: $[0.48, 0.32, 0.27]$, C2: $[0.64, 0.23, 0.19]$

B. C1: $[0.23, 0.12, 0.54]$, C2: $[0.18, 0.37, 0.34]$

C. C1: $[0.32, 0.18, 0.32]$, C2: $[0.17, 0.24, 0.12]$

D. C1: $[0.28, 0.18, 0.13]$, C2: $[0.57, 0.41, 0.49]$

Đáp án: D

Câu hỏi 19: Xác định các điểm dữ liệu thuộc cụm 1 và cụm 2 sau khi cập nhật tâm cụm (Theo vector biểu diễn)

A. $[0, 1, 0, 0, 0, 1, 1, 0, 1, 1]$

B. $[0, 0, 1, 0, 0, 0, 1, 0, 1, 0]$

C. $[0, 0, 0, 0, 0, 1, 1, 0, 1, 1]$

D. $[0, 1, 0, 1, 0, 1, 0, 0, 1, 1]$

Đáp án: C

3 Decision Tree

Tập Dữ Liệu Dự Đoán Quyết Định Chơi Tennis
Tập dữ liệu này chứa thông tin về các đặc trưng thời tiết liên quan đến quyết định chơi tennis (PlayTennis). Bộ dữ liệu gồm 4 thuộc tính:

- **Outlook**: trạng thái thời tiết (Sunny, Overcast, Rain)
- **Temperature**: nhiệt độ (Hot, Mild, Cool)
- **Humidity**: độ ẩm (High, Normal)
- **Wind**: gió (Weak, Strong)
- **PlayTennis**: quyết định chơi tennis (Yes, No)

Bộ dữ liệu để xây dựng cây quyết định:

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes

Bộ dữ liệu kiểm tra:

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

```
import numpy as np
import pandas as pd
from collections import Counter
from math import log2

!gdown --id 1rHVX3bVdaiXec3Dbts1wW8CU8wSKOvxE
data = pd.read_csv('/content/tennis.csv')
```

Xác định đặc trưng và nhãn của dữ liệu

```
X = data.iloc[:, :-1].values  # All columns except the last one are features
y = data.iloc[:, -1].values  # The last column is the label
feature_names = data.columns[:-1].tolist()
print(f"Các đặc trưng: {feature_names}")
```

```
================= Output =================
Các đặc trưng: ['Outlook', 'Temperature', 'Humidity', 'Wind']
==========================================
```

Tách dữ liệu thành 2 phần:

- Dữ liệu để xây dựng cây quyết định (10 dòng đầu tiên)
- Dữ liệu kiểm tra (4 dòng còn lại)

```
X_train, y_train = X[:10], y[:10]
X_test, y_test = X[10:], y[10:]
```

Các bước xây dựng cây quyết định (khi sử dụng Entropy):

- Bắt đầu với toàn bộ tập dữ liệu
- Chọn thuộc tính chia nhánh tốt nhất dựa trên độ đo Entropy và Information Gain
- Chia dữ liệu thành các nhánh con dựa trên thuộc tính đó
- Lặp lại quá trình đến khi:
  - Tất cả các mẫu thuộc cùng một lớp
  - Không còn thuộc tính nào để chia nhánh
  - Đạt đến điều kiện dừng (số lượng mẫu nhỏ nhất hoặc độ sâu tối đa)

Câu hỏi 20: Entropy của tập dữ liệu ban đầu là bao nhiêu (làm tròn đến 4 chữ số thập phân)?

A. 0.7219

B. 0.9710

C. 0.8113

D. 0.9183

Đáp án: B

Câu hỏi 21: Với tập dữ liệu huấn luyện ban đầu, Information Gain của thuộc tính Temperature là bao nhiêu (làm tròn đến 4 chữ số thập phân)?

A. 0.0913

B. 0.0955

C. 0.8113

D. 0.9183

Đáp án: B

Câu hỏi 22: Thuộc tính nào sẽ được chọn để chia nhánh đầu tiên?

A. Outlook

B. Temperature

C. Humidity

D. Wind

Đáp án: A

Câu hỏi 23: Cây quyết định hoàn chỉnh sẽ có bao nhiêu lá?

A. 4

B. 5

C. 6

D. 7

Đáp án: C

Câu hỏi 24: Dùng cây quyết định đã xây dựng trên dữ liệu huấn luyện, dự đoán kết quả của dòng dữ liệu cuối cùng trong tập kiểm tra.

A. Strong

B. Cool

C. Yes

D. No

Đáp án: D

Câu hỏi 25: Dùng cây quyết định đã xây dựng, dự đoán kết quả cho mẫu dữ liệu sau: Outlook = Overcast, Wind = Weak.

A. Yes

B. No

C. Thiếu giá trị của thuộc tính Temperature nên không dự đoán được

D. Thiếu giá trị của thuộc tính Humidity nên không dự đoán được

Đáp án: A

Câu hỏi 26: Dùng cây quyết định đã xây dựng, dự đoán kết quả cho mẫu dữ liệu sau: Temperature = Hot, Outlook = Rainy.

A. Yes

B. No

C. Thiếu giá trị của thuộc tính Wind nên không dự đoán được

D. Thiếu giá trị của thuộc tính Humidity nên không dự đoán được

Đáp án: C

Câu hỏi 27: Độ chính xác của mô hình cây trên tập kiểm tra là bao nhiêu?

A. 50%

B. 75%

C. 100%

D. 25%

Đáp án: B

Câu hỏi 28: Nếu một thuộc tính liên tục được chọn làm thuộc tính phân chia, bước tiếp theo sẽ là gì?

A. Chuyển đổi thuộc tính liên tục thành thuộc tính phân loại

B. Chọn điểm cắt để chia thuộc tính thành hai miền

C. Bỏ qua thuộc tính liên tục đó

D. Chia thuộc tính thành nhiều phân khúc dựa trên các giá trị trung bình

Đáp án: B

Câu hỏi 29: Một thuộc tính có giá trị Information Gain bằng 0 cho thấy điều gì?

A. Thuộc tính không mang lại thông tin để phân biệt các lớp

B. Thuộc tính là quan trọng nhất

C. Thuộc tính này có thể loại bỏ trong quá trình huấn luyện

D. Thuộc tính gây overfitting

Đáp án: A

4 Random Forest

Mô tả bài toán cho cả 4 thuật toán RF, XGBoost, Ada Boost, Gradient Boost
Trong bài toán này, chúng ta sẽ được cho một bộ dataset mô tả thông tin về nhân viên trong một công ty, bao gồm các features liên quan đến nhân viên và mức lương của họ. Nhiệm vụ của chúng ta là phân tích, xử lí bộ data dưới đây và trả lời các câu hỏi yêu cầu người làm phải thực hiện coding.
https://drive.google.com/file/d/1WkVx5tinsxuU3SnIwLNiYU7X2Sqo3vgG/view?usp=sharing
Thực hiện các yêu cầu sau đây

1. **Đọc dữ liệu**
   Sử dụng pandas, đọc file csv được cung cấp, sau đó hiển thị ra màn hình để hiểu các trường dữ liệu.
2. **Label Encoding**
   Chuyển đổi các cột dữ liệu dạng chữ (cụ thể là cột "Gender"và "Position") sang dạng số bằng cách sử dụng `LabelEncoder` từ thư viện `sklearn`.
3. **Tách dữ liệu thành bộ feature (X) và label (y)**
   - Sử dụng các cột "Gender", "Experience (Years)"và "Position"làm features đầu vào (`X`).
   - Sử dụng cột "Salary"làm biến đầu ra (`y`).
4. **Tách tập dữ liệu thành tập train và test**
   - Chia dữ liệu thành tập huấn luyện (`X_train`, `y_train`) và tập kiểm tra (`X_test`, `y_test`) với tỷ lệ 80:20.
   - Đảm bảo rằng việc chia tách dữ liệu là ngẫu nhiên nhưng tái lập được với `random_state=42`.

Câu hỏi 30: Khái niệm "bagging"trong Random Forest có ý nghĩa gì đối với sự đa dạng của các cây trong rừng?

A. Tăng độ sâu của từng cây để tăng đa dạng

B. Sử dụng các tập dữ liệu huấn luyện khác nhau cho mỗi cây để tạo ra sự đa dạng

C. Thay đổi số lượng đặc trưng tại mỗi nút phân chia

D. Áp dụng các thuật toán tối ưu hóa khác nhau cho mỗi cây

Đáp án: B

Câu hỏi 31: Hãy sắp xếp lại các bước thực hiện thuật toán Random Forest theo thứ tự đúng để hoàn thiện quy trình.

1. Tạo N bootstrapped dataset từ dữ liệu gốc.
2. Xây dựng một cây quyết định cho mỗi bootstrapped dataset.
3. Chọn ngẫu nhiên 2 feature từ dataset.
4. Tính entropy hoặc Gini để chọn feature tốt nhất.
5. Loại bỏ feature đã được chọn ra khỏi bảng dữ liệu.
6. Lặp lại quá trình chọn feature tốt nhất và loại cột đã chọn cho đến khi không còn feature nào.

A. $6 \rightarrow 5 \rightarrow 4 \rightarrow 1 \rightarrow 3 \rightarrow 2$

B. $4 \rightarrow 6 \rightarrow 5 \rightarrow 1 \rightarrow 3 \rightarrow 2$

C. $6 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 3 \rightarrow 2$

D. $5 \rightarrow 6 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 2$

Đáp án đúng: C

Câu hỏi 32: Random Forest có khả năng xử lý dữ liệu thiếu (missing data) như thế nào?

A. Random Forest Không thể xử lý dữ liệu thiếu

B. Sử dụng trung bình để điền dữ liệu thiếu

C. Tự động bỏ qua các mẫu có dữ liệu thiếu

D. Sử dụng các kỹ thuật nội suy trong quá trình huấn luyện cây

Đáp án: D

Câu hỏi 33: So với các phương pháp ensemble khác như Gradient Boosting, Random Forest thường có ưu điểm gì trong việc xử lý dữ liệu lớn và đa dạng?

A. Random Forest khó bị overfitting hơn

B. Random Forest có thời gian huấn luyện nhanh hơn

C. Random Forest có thể song song hóa dễ dàng hơn do các cây độc lập

D. Random Forest đạt được độ chính xác cao hơn trong mọi trường hợp

Đáp án: C

Câu hỏi 34: Sử dụng các hàm `mean_squared_error` và `r2_score` từ thư viện `sklearn.metrics`, hãy tính toán giá trị MSE và R^2 của mô hình Random Forest trên tập kiểm tra (sử dụng các tham số `n_estimators=50` và `random_state=42`). Giá trị MSE và R^2 là bao nhiêu?

A. MSE: 781254527.5, R^2: 0.5572

B. MSE: 827087272.8, R^2: 0.6572

C. MSE: 781254527.5, R^2: 0.6572

D. MSE: 827087272.8, R^2: 0.5572

Đáp án: D

Câu hỏi 35: Hãy thử nghiệm số lượng cây của mô hình Random Forest, với các giá trị `n_estimators` khác nhau (10, 20, 50, 100), số lượng cây nào đem lại MSE (Mean Squared Error) nhỏ nhất? (`random_state=42`)

A. 10 cây

B. 20 cây

C. 50 cây

D. 100 cây

Đáp án đúng: B

Câu hỏi 36: Hãy thử nghiệm các giá trị độ sâu của cây `max_depth` từ 1 đến 10, độ sâu nào mang lại MSE (Mean Squared Error) nhỏ nhất? (`random_state=42`)

A. Độ sâu 1

B. Độ sâu 3

C. Độ sâu 5

D. Độ sâu 10

Đáp án đúng: C

5 Ada/Gra Boost

Mô tả bài toán cho cả 4 thuật toán RF, XGBoost, Ada Boost, Gradient Boost Trong bài toán này, chúng ta sẽ được cho một bộ dataset mô tả thông tin về nhân viên trong một công ty, bao gồm các features liên quan đến nhân viên và mức lương của họ. Nhiệm vụ của chúng ta là phân tích, xử lí bộ data dưới đây và trả lời các câu hỏi yêu cầu người làm phải thực hiện coding. employee_data.csv
Tham khảo Link code gốc Ada+Gradient
Thực hiện các yêu cầu sau đây

1. **Đọc dữ liệu**
   Sử dụng pandas, đọc file csv được cung cấp, sau đó hiển thị ra màn hình để hiểu các trường dữ liệu.

2. **Label Encoding**
   Chuyển đổi các cột dữ liệu dạng chữ (cụ thể là cột "Gender"và "Position") sang dạng số bằng cách sử dụng `LabelEncoder` từ thư viện `sklearn`.

3. **Tách dữ liệu thành bộ feature (X) và label (y)**
   - Sử dụng các cột "Gender", "Experience (Years)"và "Position"làm features đầu vào (`X`).
   - Sử dụng cột "Salary"làm biến đầu ra (`y`).
4. **Tách tập dữ liệu thành tập train và test**
   - Chia dữ liệu thành tập huấn luyện (`X_train`, `y_train`) và tập kiểm tra (`X_test`, `y_test`) với tỷ lệ `80:20`.
   - Đảm bảo rằng việc chia tách dữ liệu là ngẫu nhiên nhưng tái lập (reproducibility) được với `random_state=42`.

Nhập các thư viện cần thiết và tải dữ liệu.

```
from sklearn.metrics import mean_squared_error, r2_score
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import AdaBoostRegressor, GradientBoostingRegressor
from sklearn.metrics import mean_squared_error
import numpy as np

#Tải và đọc dữ liệu
!gdown 1WkVx5tinsxuU3SnIwLNiYU7X2Sqo3vgG
data = pd.read_csv('/content/employee_data.csv')
```

Câu hỏi 37: Điểm khác biệt chính giữa AdaBoost và Gradient Boosting trong cách mà chúng cải thiện mô hình là gì?

A. AdaBoost tập trung vào việc sửa lỗi của các mẫu dữ liệu có lỗi cao nhất, còn Gradient Boosting tập trung vào giảm thiểu giá trị lỗi toàn bộ bằng cách sử dụng đạo hàm.

B. AdaBoost sử dụng các mô hình con yếu, trong khi Gradient Boosting chỉ sử dụng mô hình con mạnh.

C. AdaBoost không thể dẫn đến overfitting, trong khi Gradient Boosting dễ bị overfitting.

D. AdaBoost và Gradient Boosting có cùng cách tiếp cận trong việc cải thiện mô hình qua các bước lặp.

Đáp án: A

Câu hỏi 38: Điều gì xảy ra khi bạn tăng số lượng mô hình con (estimators) trong AdaBoost hoặc Gradient Boosting? (Thực hiện thay đổi tham số estimators để kiểm tra)

A. Hiệu suất mô hình luôn tăng khi tăng số lượng mô hình con.

B. Hiệu suất có thể tăng, nhưng nếu quá cao sẽ gây overfitting.

C. Hiệu suất giảm dần khi tăng số lượng mô hình con.

D. Hiệu suất không bị ảnh hưởng bởi số lượng mô hình con.

Đáp án: B

Huấn luyện mô hình

```
# Adaboost
ada_regressor = AdaBoostRegressor(n_estimators=50, random_state=42)
ada_regressor.fit(X_train, y_train)

# Gradient Boost
gb_regressor = GradientBoostingRegressor(n_estimators=50, random_state=42)
gb_regressor.fit(X_train, y_train)
```

Câu hỏi 39: Khi nào overfitting có thể xảy ra trong AdaBoost và Gradient Boosting? (Hãy thử nghiệm với code các trường hợp trên và đưa ra kết luận)

A. Khi sử dụng quá ít mô hình con.

B. Khi sử dụng giá trị learning rate quá cao và số lượng mô hình con quá nhiều.

C. Overfitting không xảy ra trong Gradient Boosting.

D. Khi mô hình không có đủ dữ liệu để huấn luyện.

Đáp án: B

Câu hỏi 40: AdaBoost và Gradient Boosting cho phép đánh giá tầm quan trọng của các đặc trưng. Tầm quan trọng của đặc trưng nào sẽ có khả năng cao nhất trong bài toán dự đoán lương nhân viên? (Dùng phương thức feature_importances_ có sẵn trong model)

A. Gender.

B. Experience (Years).

C. Position.

D. ID.

Đáp án: B

Câu hỏi 41: Sử dụng các hàm mean_squared_error, r2_score của thư viện sklearn.metrics để tính toán giá trị MSE và R^2 (sử dụng tham số của 2 mô hình là n_estimators=50, random_state=42), trả lời câu hỏi: Dựa trên 2 giá trị trên, hiệu suất của 2 mô hình Ada & Gradient Boost như thế nào?

A. Với giá trị R^2 từ 0.7-0.8, mô hình đang hoạt động khá tốt, giá trị MSE khá tương đối không cao không thấp.

B. Với giá trị R^2 từ 0.6-0.7, mô hình đang hoạt động ở mức khá, bao quát được một phần nhưng cũng nhiều sai sót, giá trị MSE khá cao.

C. Với giá trị R^2 từ 0.4-0.5, mô hình đang hoạt động ở mức tệ, không nắm bắt được phương sai trong dữ liệu. Giá trị MSE lớn.

D. Với giá trị R^2 từ 0.1-0.2, mô hình dường như không học được từ dữ liệu. Giá trị MSE siêu lớn thể hiện sự sai sót khi mô hình không thể học.

Đáp án: B

Câu hỏi 42: (Cho cả 3 phần) Hãy xem đây như là một bài toán thực nghiệm, thay thế từ mô hình Random Forest, XGBoost, AdaBoost, Gradient Boost vào dữ liệu trên. Sau đó đánh giá bộ dữ liệu trên tập test và đưa ra mô hình tốt nhất được chọn tối ưu cho bộ dữ liệu này. (Thực hiện với các siêu tham số chung: n_estimators = 50, random_state=42)

A. Random Forest

B. XGBoost

C. AdaBoost

D. Gradient Boost

Đáp án: D

6 XGBoost

Cài đặt và nhập các thư viện cần thiết

```
# Cài đặt thư viện XGBoost
!pip install xgboost

# Nhập các thư viện cần thiết
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import xgboost as xgb
from xgboost import XGBRegressor

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error, r2_score

from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
```

Tải và đọc dữ liệu

```
# Tải dữ liệu
# https://drive.google.com/file/d/1WkVx5tinsxuU3SnIwLNiYU7X2Sqo3vgG/view?usp=sharing
!gdown 1WkVx5tinsxuU3SnIwLNiYU7X2Sqo3vgG

# Đọc dữ liệu
data = pd.read_csv('/content/employee_data.csv')
data.head()
```

Xử lí dữ liệu

```
# Mã hóa giới tính và vị trí công việc
label_encoder_gender = LabelEncoder()
label_encoder_position = LabelEncoder()

data['Gender'] = label_encoder_gender.fit_transform(data['Gender'])
data['Position'] = label_encoder_position.fit_transform(data['Position'])

X = data.drop(columns=['ID', 'Salary'])
y = data['Salary']
```

```
# Chia tập dữ liệu thành tập train và test với tỉ lệ (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state
    =42)
X_train.shape, y_train.shape, X_test.shape, y_test.shape
```

Khởi tạo tham số của mô hình XGBoost

```
# Định nghĩa các tham số cho mô hình XGBoost
params = {
    'n_estimators': 50,
    'random_state': 42
}

# Khởi tạo mô hình XGBoost với các tham số đã định nghĩa
xgb_model = XGBRegressor(**params)
```

Câu hỏi 43: Dựa vào đoạn mã được cung cấp để trả lời câu hỏi, Mô hình XGBoost được khởi tạo đúng với mô tả nào sau đây?

A. Mô hình dùng cho bài toán phân loại, có 50 cây và đặt *random_state* để điều chỉnh số lượng cây trong mô hình.

B. Mô hình dùng cho bài toán phân loại, có 42 cây và *random_state* để giúp đảm bảo kết quả có thể tái hiện lại.

C. Mô hình dùng cho bài toán hồi quy, có 50 cây và *random_state* để điều chỉnh số lượng cây trong mô hình.

D. Mô hình dùng cho bài toán hồi quy, có 50 cây và *random_state* để giúp đảm bảo kết quả có thể tái hiện lại.

Đáp án đúng: D

```
# Huấn luyện mô hình
xgb_model.fit(X_train, y_train)
```

Câu hỏi 44: Sau khi huấn luyện mô hình trên. Đâu là đặc trưng được sử dụng nhiều nhất để phân nhánh? Hãy tìm hiểu và sử dụng hàm `.get_score()` với tham số `importance_type='weight'`. Tài liệu tham khảo

A. Gender

B. Experience

C. Position

D. Salary

Đáp án đúng: B

Câu hỏi 45: Hãy in ra cây đầu tiên (0). Điều kiện để phân nhánh lần thứ 2 (Root Node tính là 1 lần) được hiển thị trên hình là? Hãy tìm hiểu và sử dụng hàm `.plot_tree()` với tham số bắt buộc phải có là `num_trees=0`.

A. Position < 5

B. Position > 5

C. Experience < 8

D. Experience > 8

Đáp án đúng: A

Câu hỏi 46: Hãy in ra cây đầu tiên (0). Nếu một input có Experience (Year) = 4 và Position = 4, cây thứ 0 này sẽ trả về kết quả là? Hãy tìm hiểu và sử dụng hàm `.plot_tree()` với tham số bắt buộc phải có là `num_trees=0`.

A. -14229.6719

B. -9139.3916

C. 4251.66406

D. -2619.49707

Đáp án đúng: D

Câu hỏi 47: Giá trị đánh giá *Mean Square Error* và R^2 *Score* trên tập Test cho mô hình XGBoost trên xấp xỉ là? Hãy sử dụng hàm `mean_squared_error()` và `r2_score()`.

A. MSE = 1182725453, R2 Score = 0.3668

B. MSE = 1325373291, R2 Score = 0.2904

C. MSE = 1150167002, R2 Score = 0.3842

D. MSE = 1150167002, R2 Score = 0.2904

Đáp án đúng: A