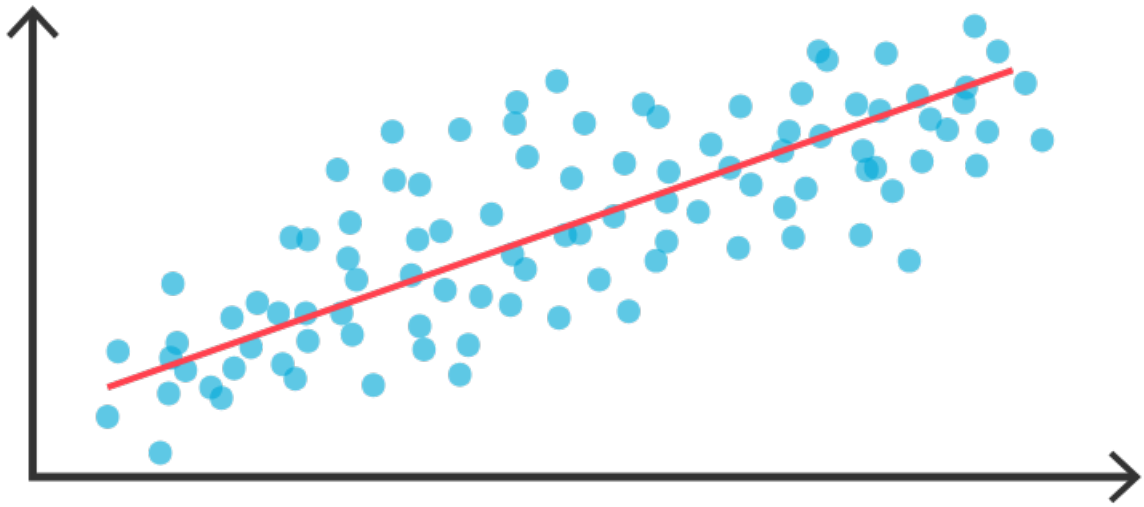
AI VIET NAM – COURSE 2024

Ensemble Learning - Exercise

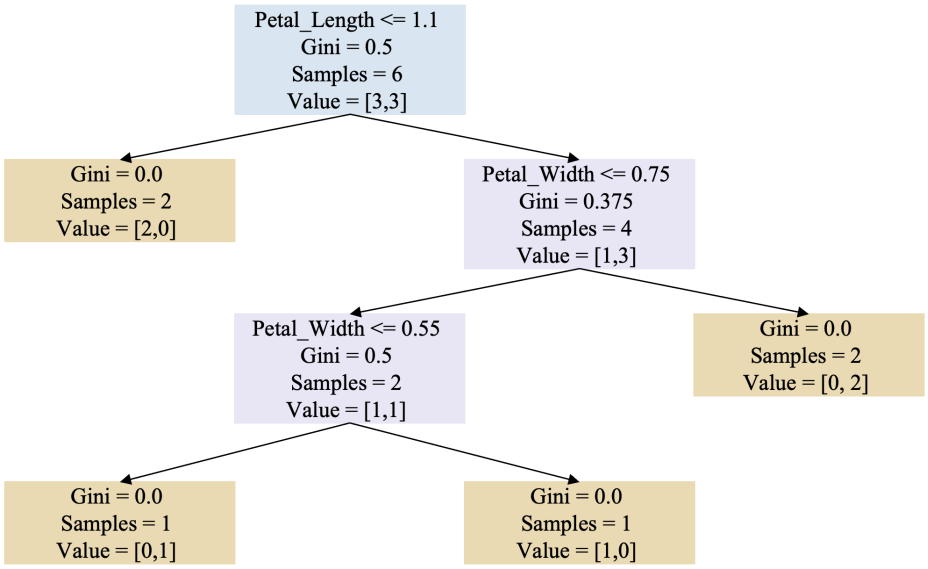
Ngày 8 tháng 9 năm 2024

Phần I: Giới thiệu

Trong bài tập này, chúng ta sẽ thực hành các nội dung liên quan đến Decision Tree, Random Forest, AdaBoost và Gradient Boosting áp dụng cho bài toán Regression. Trong Machine Learning, bài toán regression là dạng bài toán mà kết quả đầu ra của mô hình là một giá trị liên tục.

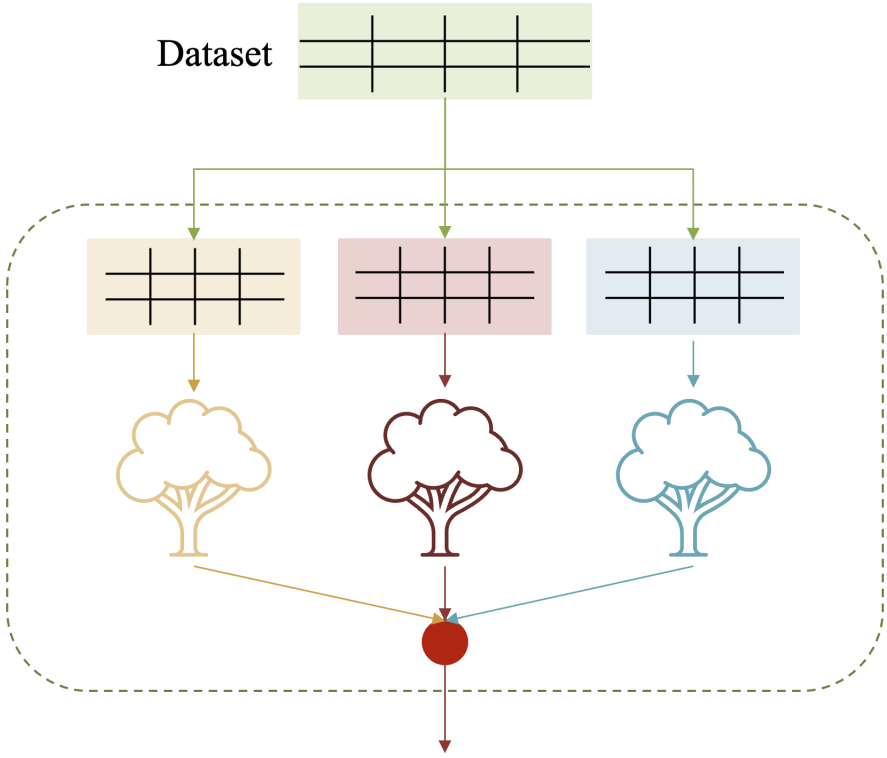


Decision Tree (DT) là một thuật toán Machine Learning theo kiểu Supervised Learning có thể dùng để giải quyết cho hai dạng bài toán là Regression và Classification. DT xây dựng một cấu trúc dạng cây, với các node đại diện cho quyết định dựa trên một điều kiện của đặc trưng nào đó. Sau khi trải qua một loạt quyết định, kết quả dự đoán cuối cùng chính là giá trị mà node lá nắm giữ.



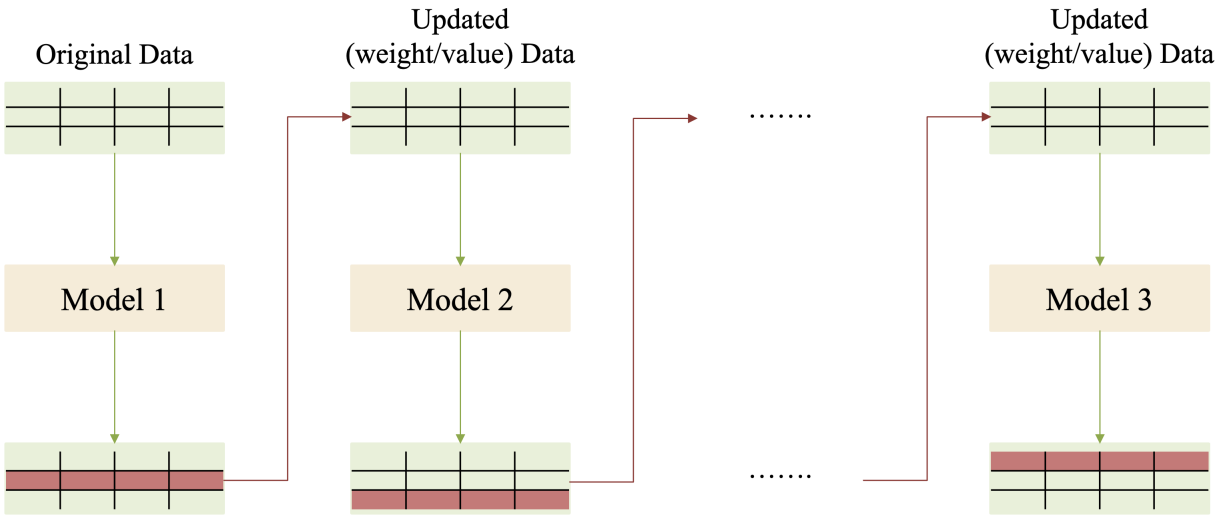
Hình 1: Ví dụ minh họa về một Decision Tree.

Random Forest (RF) là một thuật toán Machine Learning theo kiểu Ensemble Learning có thể dùng để giải quyết cho hai dạng bài toán là Regression và Classification. Ý tưởng của RF liên quan đến việc sử dụng nhiều DT, mỗi cây sẽ học trên một tập con của một bộ training dataset (Boostrap). Khi thực hiện dự đoán, mỗi cây sẽ đưa ra kết quả dự đoán của mình, sau đó tổng hợp toàn bộ kết quả lại theo một cách thức nào đó (ví dụ: lấy trung bình với bài Regression hay thực hiện biểu quyết với bài Classification) để trả về kết quả cuối cùng.



Hình 2: Ví dụ minh họa về Random Forest.

AdaBoost và Gradient Boosting là hai mô hình điển hình cho kiến trúc tiếp cận Ensemble Learning, thay vì các mô hình được học song song nhau trên các tập con dataset, thì với 2 phương pháp này sẽ được học theo trình tự. Mô hình sau sẽ học kế thừa từ tri thức của mô hình trước.



Hình 3: Ví dụ minh họa về Boosting cho AdaBoost và Gradient Boosting.

Trong phần này chúng ta sẽ đánh giá tính hiệu quả của các mô hình này trên bộ dữ liệu cho bài toán Regression.

Phần II: Bài tập

A. Phần lập trình

Trong phần này, chúng ta sẽ cài đặt và huấn luyện hai mô hình Decision Tree và Random Forest để giải quyết bài toán Regression về dự đoán giá nhà. Các bước thực hiện như sau:

1. **Import thư viện các thư viện cần thiết:**

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

from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import OrdinalEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

2. **Tải bộ dữ liệu:** Các bạn tải bộ dữ liệu Housing.csv tại đây.

3. **Đọc bộ dữ liệu:** Sử dụng thư viện pandas, chúng ta sẽ đọc file .csv lên như sau:

```python
dataset_path = './Housing.csv'
df = pd.read_csv(dataset_path)
```

Khi đó, ta có thể thấy nội dung của bảng dữ liệu có dạng như sau:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished

Hình 4: Một vài mẫu dữ liệu của bộ dữ liệu Housing.

Vì bài toán của chúng ta là dự đoán giá nhà dựa trên các thuộc tính cho trước, từ đó có thể xác định được cột **price** sẽ là cột label y và tất cả các cột còn lại là cột features X.

4. **Xử lý dữ liệu categorical:** Để có thể thực hiện bất kì tính toán nào, ta cần phải số hóa toàn bộ giá trị trong bảng dữ liệu. Nhận thấy có một số cột trong bảng dữ liệu có dạng string, ta sẽ phải đổi toàn bộ các giá trị này về dạng số. Đầu tiên, ta kiểm tra các cột có kiểu dữ liệu là Object như sau:

```python
categorical_cols = df.select_dtypes(include=['object']).columns.to_list()
print(categorical_cols)

# ['mainroad', 'guestroom', 'basement', 'hotwaterheating', 'airconditioning', '
    prefarea', 'furnishingstatus']
```

Sau khi đã xác định được đối tượng cần xử lý, ta sẽ sử dụng OrdinalEncoder() để chuyển đổi chúng thành dạng số như sau:

```
ordinal_encoder = OrdinalEncoder()
encoded_categorical_cols = ordinal_encoder.fit_transform(
    df[categorical_cols]
)
encoded_categorical_df = pd.DataFrame(
    encoded_categorical_cols,
    columns=categorical_cols
)
numerical_df = df.drop(categorical_cols, axis=1)
encoded_df = pd.concat(
    [numerical_df, encoded_categorical_df], axis=1
)
```

Khi đã hoàn tất, ta được một DataFrame mới có dạng như sau:

	price	area	bedrooms	bathrooms	stories	parking	mainroad	guestroom	basement	hotwaterheating	airconditioning	prefarea	furnishingstatus
0	13300000	7420	4	2	3	2	1.0	0.0	0.0	0.0	1.0	1.0	0.0
1	12250000	8960	4	4	4	3	1.0	0.0	0.0	0.0	1.0	0.0	0.0
2	12250000	9960	3	2	2	2	1.0	0.0	1.0	0.0	0.0	1.0	1.0
3	12215000	7500	4	2	2	3	1.0	0.0	1.0	0.0	1.0	1.0	0.0
4	11410000	7420	4	1	2	2	1.0	1.0	1.0	0.0	1.0	0.0	0.0

Hình 5: Bộ dữ liệu Housing sau khi đã chuyển đổi toàn bộ các đặc trưng Categorical về dạng số.

5. **Chuẩn hóa bộ dữ liệu:** Để việc tính toán thuận lợi, ta tiến hành chuẩn hóa toàn bộ giá trị trong bộ dữ liệu sử dụng StandardScaler() như sau:

```
normalizer = StandardScaler()
dataset_arr = normalizer.fit_transform(encoded_df)
```

```
array([[ 4.56636513,  1.04672629,  1.40341936, ...,  1.4726183 ,
         1.80494113, -1.40628573],
       [ 4.00448405,  1.75700953,  1.40341936, ...,  1.4726183 ,
        -0.55403469, -1.40628573],
       [ 4.00448405,  2.21823241,  0.04727831, ..., -0.67906259,
         1.80494113, -0.09166185],
       ...,
       [-1.61432675, -0.70592066, -1.30886273, ..., -0.67906259,
        -0.55403469,  1.22296203],
       [-1.61432675, -1.03338891,  0.04727831, ..., -0.67906259,
        -0.55403469, -1.40628573],
       [-1.61432675, -0.5998394 ,  0.04727831, ..., -0.67906259,
        -0.55403469,  1.22296203]])
```

Hình 6: Bộ dữ liệu Housing (ở dạng ndarray) sau khi được chuẩn hóa.

6. **Tách dữ liệu X, y:** Khi đã hoàn tất các bước tiền xử lý, lúc này ta sẽ tách dữ liệu ban đầu thành hai biến X, y đại diện cho các đặc trưng và nhãn. Nhận thấy cột đầu tiên, **price**, là nhãn của bộ dữ liệu, ta sẽ tiến hành tách X, y ra như sau:

```
X, y = dataset_arr[:, 1:], dataset_arr[:, 0]
```

7. **Chia tập dữ liệu train, val:** Dựa vào bộ dữ liệu gốc, ta cần chia thành hai tập dữ liệu con, một dùng cho việc huấn luyện mô hình và một cho việc đánh giá mô hình. Ở đây, ta sẽ chia theo tỷ lệ 7:3 và tham số ngẫu nhiên $random\_state = 1$:

```
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
)
```

8. **Huấn luyện mô hình:** Ta thực hiện huấn luyện mô hình với bộ dữ liệu train. Để huấn luyện mô hình Random Forest, các bạn sẽ sử dụng RandomForestRegressor():

```
regressor = RandomForestRegressor(
    random_state=random_state
)
regressor.fit(X_train, y_train)
```

   Để huấn luyện mô hình AdaBoost, các bạn sẽ sử dụng AdaBoostRegressor():

```
regressor = AdaBoostRegressor(
    random_state=random_state
)
regressor.fit(X_train, y_train)
```

   Để huấn luyện mô hình Gradient Boosting, các bạn sẽ sử dụng GradientBoostingRegressor():

```
regressor = GradientBoostingRegressor(
    random_state=random_state
)
regressor.fit(X_train, y_train)
```

9. **Đánh giá mô hình:** Để biết được mô hình đã huấn luyện có hoạt động tốt trên các mẫu dữ liệu mới hay không, ta sẽ đánh giá thông qua tập val. Đầu tiên, ta cho mô hình đã huấn luyện thực hiện dự đoán trên toàn bộ tập val:

```
y_pred = regressor.predict(X_val)
```

   Lúc này, ta hoàn toàn có thể áp dụng những độ đo đánh giá dành cho bài Regression để thực hiện đánh giá hiệu suất mô hình. Ở đây, ta sẽ sử dụng Mean Absolute Error (MAE) và Mean Squared Error (MSE):

```
mae = mean_absolute_error(y_val, y_pred)
mse = mean_squared_error(y_val, y_pred)

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

B. Phần trắc nghiệm

1. Giá trị khởi tạo cho giá trị dự đoán trong thuật toán Gradient Boosting được xác định dựa vào?

   (a) Giá trị lớn nhất
   (b) Giá trị nhỏ nhất
   (c) Giá trị trung bình
   (d) Tổng các giá trị

2. Thuật ngữ nào sau đây không thuộc Regression Tree?

   (a) Logistic Regression
   (b) Gini Index
   (c) Pruning
   (d) Splitting Criterion

3. Công thức nào sau đây là đúng về Mean Squared Error (MSE)?

   (a) $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$
   (b) $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
   (c) $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
   (d) $\frac{1}{n}\sum_{i=1}^{n}\frac{y_i}{\hat{y}_i}$

4. Trong regression tree, thành phần nào đóng vai trò là điểm khởi đầu trong việc đưa ra quyết định?

   (a) Internal Node
   (b) Leaf Node
   (c) Branch
   (d) Root Node

5. Trong regression tree, thành phần nào đóng vai trò chứa kết quả dự đoán và không phân nhánh?

   (a) Internal Node
   (b) Leaf Node
   (c) Branch
   (d) Root Node

6. Cho một bộ dataset có nội dung như sau:

X	Y
3	12
5	20
8	28
10	32
12	36

   Dựa theo lý thuyết về Decision Tree, các bạn hãy trả lời một số câu hỏi sau:

   (a) Điều kiện nào sau đây có thể là điều kiện phân nhánh đầu tiên cho cây, sử dụng độ đo MSE làm splitting criterion?

   (a) $X \leq 3$
   (b) $X \leq 8$
   (c) $X \leq 5$
   (d) $X \leq 10$

   (b) Dựa vào kết quả đạt được ở câu 6.a, giả sử với giá trị đầu vào X là 2, kết quả dự đoán của cây là?

(a) 14
(b) 16
(c) 18
(d) 20

(c) Dựa vào kết quả đạt được ở câu 6.a, giả sử với giá trị đầu vào X là 15, kết quả dự đoán của cây là?

(a) 20
(b) 24
(c) 30
(d) 32

7. Dựa vào phần lập trình mục II.A, kết quả đánh giá MSE của mô hình Random Forest đã huấn luyện được trên tập val là (lấy giá trị xấp xỉ)?

(a) 0.36
(b) 0.38
(c) 0.40
(d) 0.42

8. Lý do chính của việc sử dụng bootstrap dataset trong Random Forest là?

(a) Tăng tốc độ huấn luyện
(b) Ưu tiên các đặc trưng quan trọng
(c) Khắc phục missing values
(d) Tránh overfitting

9. Từ nào sau đây được dùng để miêu tả số lượng cây trong Random Forest?

(a) max_depth
(b) max_features
(c) n_estimators
(d) criterion

10. Trong Random Forest, khái niệm Bagging được hiểu là?

(a) Bootstrap Aggregating
(b) Binary Aggregating
(c) Balanced Algorithm Grouping
(d) Best Algorithm for Generalization

11. Cho một bộ dataset có nội dung như sau:

X	**Y**
2	4
1	3
3	5
2	6

Dựa trên bộ dataset này, bạn sẽ xây dựng một mô hình random forest với số cây là 2. Độ sâu cho mỗi cây $max\_depth = 1$ và giá trị dự đoán cuối cùng là trung bình các giá trị dự đoán của mỗi cây. Từ đây, các bạn hãy trả lời một số câu hỏi sau:

(a) Giả sử cả hai cây đều tính toán trên toàn bộ dataset trên. Cây 1 chia nhánh với điều kiện $X \geq 2$, cây 2 chia nhánh với điều kiện $X \geq 3$. Khi đó, kết quả dự đoán của mô hình với $X = 2$ là (lấy giá trị xấp xỉ)?

(a) 4.0
(c) 5.0
(b) 4.5
(d) 5.5

(b) Giả sử bộ dữ liệu bootstrap của hai cây lần lượt là (hiển thị theo chỉ mục của các mẫu trong bộ dữ liệu gốc):

- **Cây 1:** Gồm các mẫu 0, 1, 2.
- **Cây 2:** Gồm các mẫu 1, 2, 3.

Cây 1 chia nhánh với điều kiện $X \geq 2$, cây 2 chia nhánh với điều kiện $X \geq 3$. Khi đó, kết quả dự đoán của mô hình với $X = 1$ là?

(a) 3.25
(c) 3.75
(b) 3.50
(d) 4.0

12. Dựa vào phần lập trình mục II.A, kết quả đánh giá MAE của mô hình Random Forest đã huấn luyện được trên tập val là (lấy giá trị xấp xỉ)?

(a) 0.45
(c) 0.47
(b) 0.46
(d) 0.48

- Hết -