**Mô hình học máy và Kỹ thuật Stacked Regression**

1.Support Vector Regression

1.1 Xây dựng mô hình

SVR tạo ra một mô hình hồi quy bằng cách tìm ra một siêu phẳng trong không gian đặc trưng sao cho tổng khoảng cách của các điểm dự đoán cho các mẫu dữ liệu gần với siêu phẳng này không vượt quá một ngưỡng cho trước (được gọi là epsilon, thường ký hiệu là ε). Mô hình SVR có dạng:

Trong đó:

* là giá trị dự đoán
* là vector trọng số
* là vector đặc trưng đầu vào
* là sai số (bias)

1.2 Hàm mất mát

SVR sử dụng hàm mất mát để đo sự sai lệch giữa giá trị dự đoán và giá trị thực tế y. Hàm mất mát này được điều chỉnh bằng cách thêm vào thành phần kiểm soát mềm (soft-margin) nhằm đảm bảo tính chất hồi quy và kiểm soát các điểm ngoại lai (outliers). Hàm mất mát được viết dưới dạng:

Trong đó:

* là tham số kiểm soát độ chặt của biên (regularization parameter), giá trị càng lớn, biên càng chặt
* là biến nhỏ dương (slack variables) cho phép điểm dữ liệu nằm ngoài biên chấp nhận được
  1. Tối ưu hóa

Mục tiêu của SVR là tối ưu hóa hàm mất mát theo các tham số và , đồng thời tìm ra các biến sao cho hàm mất mát là nhỏ nhất. Điều này được thực hiện bằng cách sử dụng thuật toán tối ưu hóa Gradient Descent

* 1. Kernel Trick

Một điểm mạnh của SVR là khả năng sử dụng kernel trick để ánh xạ dữ liệu từ không gian đặc trưng ban đầu sang một không gian đặc trưng cao hơn, giúp mô hình SVR có khả năng học được các mô hình phi tuyến tính. Các kernel được sử dụng bao gồm: Linear, Polynomial, Radial Basis Function

2. LASSO Regression

2.1 Xây dựng mô hình

LASSO Regression là một mô hình hồi quy tuyến tính có dạng

Trong đó:

* là các giá trị dự đoán
* là các hệ số bias (intercept)
* là các hệ số của các đặc trưng tương ứng

2.2 Hàm mất mát

Trong LASSO Regression, giá trị mất mát được tính toán bằng hàm Mean Squared Error (MSE) sử dụng thêm L1 regularization:

Trong đó :

* là hàm mất mát bình phương trung bình giữa giá trị dự đoán và giá trị thực tế
* là tham số regularization, quyết định mức độ kiểm soát của hình phạt L1
* là tổng giá trị tuyệt đối của các hệ số

2.3 Tối ưu hóa

Mục tiêu của LASSO Regression là tối thiểu hàm mất mát bằng cách điều chỉnh các hệ số . Quá trình này được thực hiện bằng thuật toán tối ưu Gradient Descent

3. RIDGE Regression

3.1 Xây dựng mô hình

RIDGE Regression là một mô hình hồi quy tuyến tính có dạng

Trong đó:

* là các giá trị dự đoán
* là các hệ số bias (intercept)
* là các hệ số của các đặc trưng tương ứng

3.2 Hàm mất mát

Trong LASSO Regression, giá trị mất mát được tính toán bằng hàm Mean Squared Error (MSE) sử dụng thêm L2 regularization:

Trong đó :

* là hàm mất mát bình phương trung bình giữa giá trị dự đoán và giá trị thực tế
* là tham số regularization, quyết định mức độ kiểm soát của hình phạt L2
* là tổng bình phương của các hệ số

3.3 Tối ưu hóa

Mục tiêu của LASSO Regression là tối thiểu hàm mất mát bằng cách điều chỉnh các hệ số . Quá trình này được thực hiện bằng thuật toán tối ưu Gradient Descent

4. Elastic Net (ENET) Regression

ENET Regression là một sự kết hợp giữa LASSO Regression và RIDGE Regression, sử dụng cả hai thành phần kiểm soát L1 và L2 vào hàm mất mát. Hàm mất mát trong ENET Regression được biểu diễn như sau:

Trong đó:

* là hàm mất mát bình phương trung bình giữa giá trị dự đoán và giá trị thực tế
* là tham số regularization, quyết định mức độ kiểm soát của hình phạt L1, L2 tương ứng
* là hệ số kết hợp giữa L1 và L2 regularization. Giá trị của được chọn từ khoảng

5.Partial Least Squares (PLS) Regression

PLS Regression là một mô hình hồi quy bằng cách tạo ra các thành phần mới từ các biến đầu vào ban đầu và sử dụng chúng để dự đoán biến mục tiêu qua các bước sau:

* PLS thực hiện quy trình tạo ra các thành phần mới (latent variables) thông qua việc tổ hợp tuyến tính giữa biến đầu vào và biến mục tiêu
* PLS tìm một thành phần latent variable đầu tiên bằng cách tối ưu hóa sự tương quan giữa biến đầu vào và biến mục tiêu
* Sau đó, PLS loại bỏ phần tương quan khỏi biến ban đầu và tiếp tục tìm thành phần tiếp theo, quy trình này được thực hiện cho đến khi có đủ số lượng thành phần hoặc khi sự tương quan giữa các thành phần và biến mục tiêu không còn đủ lớn

6. Random Forest Regression

Random Forest Regression là một kĩ thuật hồi quy sử dụng thuật toán ensemble dựa trên cây quyết định. Thuật toán hoạt động bằng cách xây dựng nhiều cây quyết định ngẫu nhiên và kết hợp dự đoán từ tất cả các cây để tạo ra một dự đoán cuối cùng.

6.1 Lựa chọn mẫu và đặc trưng ngẫu nhiên

Random Forest Regression bắt đầu bằng việc chọn ngẫu nhiên một mẫu con (bootstrap sample) từ tập dữ liệu huấn luyện. Bootstrap sample là một tập con của dữ liệu được tạo ra bằng việc lấy ngẫu nhiên nhiều mẫu từ dữ liệu gốc có thay thế

Sau đó, Random Forest chọn một số đặc trưng ngẫu nhiên từ tập dữ liệu để xây dựng cây quyết định. Việc này giúp giảm tính đa dạng của các cây con và ngăn ngừa overfitting

6.2 Xây dựng các cây quyết định

Đối với mỗi cây quyết định trong Random Forest, việc xây dựng cây quyết định được thực hiện dựa trên tập mẫu con và số đặc trưng được chọn ngẫu nhiên. Quá trình này bao gồm tìm ra các ngưỡng (thresholds) tốt nhất cho việc chia tập dữ liệu thành các nhánh (nodes) sao cho hàm mất mát được giảm tối thiểu.

6.3 Kết hợp các cây

Khi tất cả các cây con được xây dựng , Random Forest sử dụng chúng để dự đoán giá trị bằng các lấy trung bình. Điều này tạo ra một dự đoán tổng hợp từ các cây con

7. XGBoost Regression

7.1 Xây dựng cây quyết định

XGBoost bắt đầu bằng việc xây dựng cây quyết định, đó là cây quyết định có chiều sâu bằng 1. Cây quyết định này sẽ là cây cơ sở (base learner)

7.2 Tạo dự đoán ban đầu

Ban đầu, mô hình XGBoost sử dụng cây cơ sở để tạo dự đoán ban đầu cho mỗi điểm dữ liệu trong tập huấn luyện. Dự đoán ban đầu này là giá trị trung bình của biến mục tiêu trên toàn bộ tập huấn luyện.

7.3 Tính toán hàm mất mát

XGBoost sử dụng Mean Squared Error làm hàm mất mát để đo lường sự sai lệch giữa giá trị dự đoán ban đầu và giá trị thực tế của biến mục tiêu.

7.4 Xây dựng cây mới để tối thiểu hàm mất mát

XGBoost xây dựng một cây mới (cây tầng) để làm giảm mất mát so với giá trị dự đoán ban đầu. Cây tầng này được xây dựng bằng cách tối ưu hàm mất mát với thuật toán Gradient Descent/

7.5 Tạo trọng số cho cây

Mỗi cây được xây dựng trong XGBoost được gán một trọng số dựa trên sự giảm mất mát mà nó đóng góp vào trong mô hình. Các cây có trọng số lớn hơn sẽ được coi trọng hơn khi tạo dự đoán cuối cùng.

7.6 Cập nhật dự đoán

Dự đoán cuối cùng cho một điểm dự đoán sẽ là tổng hợp của các dự đoán của tất cả các cây với trọng số của nó.

7.7 Tiến trình tạo thêm cây và cập nhật trọng số

Quá trình xây dựng các cây tầng và cập nhật trọng số tiếp tục cho đến khi một số lượng cây tầng tối ưu được xây dựng hoặc cho đến khi tiêu chí dừng được đáp ứng. Tiêu chí dừng bao gồm: giới hạn về độ sâu của cây, giới hạn về số lượng cây tầng, …

8. Light Gradient Boosting Machine (LightGBM) Regression

LightGBM là một thuật toán hồi quy biến thể của Gradient Boosting Machine và là một loại thuật toán ensemble learning. Nó được thiết kế để cải tiến hiệu suất và tốc độ của các thuật toán Gradient Boosting truyền thống như XGBoost.

8.1 Xây dựng cây quyết định phát triển theo cách tối ưu

Một điểm đặc biệt của LightGBM là cách cây quyết định được xây dựng. LightGBM sử dụng một phương pháp tối ưu “Leaf-wise” thay vì phương pháp “Level-wise” như các thuật toán Gradient Boosting khác.

Phương pháp “Leaf-wise” xây dựng cây bằng cách tìm lá có profit lớn nhất để mở rộng. Điều này giúp giảm độ sâu của cây và tạo ra các cây cân bằng hơn.

8.2 Xử lý dữ liệu thiếu

LightGBM có khả năng xử lý dữ liệu thiếu (missing data) một cách tự động.

8.3 Tích hợp tăng cường (Regularization)

LightGBM hỗ trợ tích hợp tăng cường bằng cách sử dụng L1 và L2 regularization như trong thuật toán LASSO và RIDGE giúp kiểm soát overfitting và tạo ra mô hình ổn định hơn.

8.4 Tăng tốc bằng GPU và Parallel Computing

LightGBM có khả năng sử dụng GPU để tăng tốc quá trình huấn luyện, giúp giảm thiểu thời gian huấn luyện đối với tập dữ liệu lớn. Không những thế, nó cũng hỗ trợ tính toán song song trên nhiều lõi CPU.

9. Stacked Regression

Phương pháp này kết hợp 10 kĩ thuật hồi quy học máy bao gồm: Support Vector Regression (SVR) với 3 kernel: Linear, Polynomial, Radial; LASSO Regression; RIDGE Regression; Elastic Net Regression; Partial Least Squares Regression; Random Forest Regression; XGBoost Regression; Light Gradient Boosting Machine Regression

9.1 Quá trình huấn luyện

Trong phương pháp này, chúng tôi sử dụng Light Gradient Boosting Machine Regression làm meta-model để thêm vào các mô hình cơ sở (base-learner) được lấy trung bình và sử dụng các dự đoán out-of-folds của các mô hình cơ sở này để huấn luyện meta-model

Quy trình huấn luyện được mô tả như sau:

* Chia tập huấn luyện thành hai tập con không giao nhau (train và holdout)
* Huấn luyện các mô hình cơ sở trên tập train
* Kiểm tra các mô hình cơ sở này trên tập holdout
* Sử dụng dự đoán từ việc kiểm tra (dự đoán out-of-folds) làm đầu vào và các giá trị biến mục tiêu làm đầu ra để huấn luyện cho mô hình bậc cao hơn (meta-model)

Ba bước đầu tiên được thực hiện một cách lặp lại. Trong phương pháp này, chúng tôi sử dụng quá trình stacking 5-fold, trước hết chia dữ liệu thành 5 phần. Sau đó, thực hiện lặp lại 5 lần. Trong mỗi lần lặp lại, mọi mô hình cơ sở được huấn luyện trên 4 phần và dự đoán trên phần holdout còn lại. Vì vậy, sau 5 lần lặp lại, đảm bảo rằng toàn bộ dữ liệu được sử dụng để có các dự đoán out-of-folds, sau đó chúng sẽ được sử dụng như các đặc trưng mới để huấn luyện meta-model ở bước 4.

Quy trình dự đoán được thực hiện như sau:

* Lấy trung bình của các dự đoán từ tất cả các mô hình cơ sở trên dữ liệu kiểm thử và sử dụng chúng như meta-feature.
* Sử dụng các meta-feature này đối với meta-model để đưa ra dự đoán cuối cùng.

A diagram of a diagram

Description automatically generated