**TRƯỜNG ĐẠI HỌC SƯ PHẠM KĨ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**⬩⬩🙠🕮🙢⬩⬩**



**BÁO CÁO ĐỒ ÁN 2**

**ĐỀ TÀI**

**TÌM HIỂU THUẬT TOÁN**

**GRADIENT BOOSTING**

**GVHD: TS.Trần Nhật Quang**

**SVTH MSSV**

**Phạm Thị Như Hảo 16110068**

**Thành phố Hồ Chí Minh, tháng 5 năm 2019**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KĨ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**⬩⬩🙠🕮🙢⬩⬩**



**BÁO CÁO ĐỒ ÁN 2**

**ĐỀ TÀI**

**TÌM HIỂU THUẬT TOÁN**

**GRADIENT BOOSTING**

**GVHD: TS.Trần Nhật Quang**

**SVTH MSSV**

**Phạm Thị Như Hảo 16110068**

**Thành phố Hồ Chí Minh, tháng 5 năm 2019**

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Giáo viên hướng dẫn

(ký và ghi họ tên)

……………………………

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Giáo viên phản biện

(ký và ghi họ tên)

…………………………

LỜI CẢM ƠN

Lời đầu tiên, cho phép em được gửi lời cảm ơn chân thành đến thầy Trần Nhật Quang, người đã phụ trách hướng dẫn chúng em trong việc thực hiện và hoàn thành đề tài một cách tối ưu, hiệu quả nhất. Những sự chỉ dạy, nhật xét, đóng góp ý kiến từ thầy đã góp phần rất quan trọng trong việc giúp chúng em hoàn hành đề tài một cách hoàn thiện nhất.

Cảm ơn các thầy cô trong khoa công nghệ thông tin đã luôn nhiệt tình, cống hiến, tận tình giải đáp các thắc mắc của chúng em. Bên cạnh đó, chúng em xin được cảm ơn các bạn cùng khóa, các anh chị khóa trên đã không ngần ngại chia sẻ các thông tin, kinh nghiệm bổ ích giúp chúng em hoàn thiện hơn đề tài của mình.

Bài thu hoạch được thực hiện trong khoảng thời gian có hạn, cùng với những hạn chế về mặt kiến thức nên việc sai sót là điều không thể tránh khỏi trong quá trình thực hiện. Chúng em kính mong nhận được những lời nhận xét, ý kiến đóng góp quý báo từ quý Thầy Cô để chúng em có thêm kinh nghiệm và hoàn thành tốt hơn những đề tài sau này.

Chúng em xin chân thành cảm ơn.

Thành phố Hồ Chí Minh, tháng 5 năm 2019

Sinh viên thực hiện

(kí và ghi rõ họ tên)

……………………………………

TÓM TẮT

Trong những năm trở lại đây, công nghệ thông tin đã và đang không ngừng phát triển một các mạnh mẽ. Sự ra đời của công nghệ thông tin đã một phần nào giúp cho xã hội, đời sống của người dân vươn lên một tầm cao mới và một cách rõ rệt, ngoài ra công nghệ thông tin còn góp phần không nhỏ cho sự phát triển của nhân loại khắp nơi trên thế giới.

Trong những năm gần đây AI – Artifical Intelligence (Trí tuệ nhân tạo), cụ thể hơn là Machine Learning phát triển một cách mạnh mẽ. Với hàng loạt các thiết bị, công nghệ thông minh được ứng dụng vào đời sống như: xe tự hành, hệ thống nhận diện khuôn mặt,v.v… Machine Learning là một phần của AI, là một lĩnh vực nghiên cứu cho máy tính khả năng tự học thông qua kinh nghiệm mà không cần phải lập trình tường minh.

Nhận thấy sự thú vị từ lĩnh vực này nên em đã chọn tìm hiểu về thuật toán “Gradient Boosting” là đề tài cho đồ án 2. Nhằm tăng thêm vốn hiểu biết cũng như kiến thức từ lĩnh vực này

MỤC LỤC

[1. Đạo văn 8](#_Toc9537070)

[1.1. Khái niệm 8](#_Toc9537071)

[1.2. Những việc cần làm để tránh đạo văn 8](#_Toc9537072)

[1.3. Những việc không nên làm 8](#_Toc9537073)

[1.4. Lời cam kết 9](#_Toc9537074)

[2. Lý thuyết 10](#_Toc9537075)

[2.1. Khái niệm 10](#_Toc9537076)

[2.2. Thuật toán 10](#_Toc9537077)

[2.2.1. Loss Function 10](#_Toc9537078)

[2.2.2. Weak Learner (Người học yếu) 11](#_Toc9537079)

[2.2.3. Additive Model (Mô hình phụ) 12](#_Toc9537080)

[2.3. Cải tiến cho thuật toán Gradient Boosting 12](#_Toc9537081)

[3. Cài đặt thuật toán 13](#_Toc9537082)

[3.1. Datasets 13](#_Toc9537083)

[3.2. Các thư viện được sử dụng: 14](#_Toc9537084)

[3.2.1. Numpy 14](#_Toc9537085)

[3.2.2. Pandas 14](#_Toc9537086)

[3.2.3. Scikit – learn 14](#_Toc9537087)

[3.2.4. Matplotlib 15](#_Toc9537088)

[3.3. Hướng cài đặt thuật toán 15](#_Toc9537089)

[4. Đánh giá hiệu năng 16](#_Toc9537090)

[5. Kết luận 18](#_Toc9537091)

[6. Tài liệu tham khảo: 18](#_Toc9537092)

# Đạo văn

## Khái niệm

Đạo văn được hiểu là việc sử dụng các công trình nghiên cứu, các sản phẩm cùa người khác làm ra và nói là do mình làm nên, hay việc sử dụng, tham khảo mà không trích dẫn nguồn hoặc có ghi nguồn nhưng một cách thiếu chi tiết, không đầy đủ.[1]

## Những việc cần làm để tránh đạo văn

Khi sử dụng các câu văn, dữ lệu thống kê, hình ảnh, các sản phẩm nghiên cứu phải luôn nhớ trích dẫn nguồn, việc trích dẫn nguồn phải tuân theo các chuẩn như: APA reference, v.v…

Trong trường hợp không lấy nguyên văn của tác giả, mà chỉ sử dụng ý chính, người viết cần có kĩ năng tự diễn giải ý chính đấy theo cách của mình nhưng vẫn phải trích dẫn đầy đủ. Quá trình này thường diễn ra theo 3 bước:

Bước 1: Đọc kĩ câu văn hoặc đoạn văn cần tham khảo, ghi lại ý chính

Bước 2: Từ những ý chính đã có, diễn đạt lại theo cách riêng của mình, thông thường người ta sử dụng các từ đồng nghĩa, thay đổi cấu trúc câu văn,..v.v….

Bước 3: Ghi nguồn tham khảo theo các chuẩn đã quy định.[1]

## Những việc không nên làm

Không ghi nguồn, hoặc có ghi nguồn nhưng ghi một cách nửa vời, không minh bạch, không có tính chính xác làm cho người đọc không thể truy cập đến nguồn.

Sao chép gần như là hoàn toàn các bài viết, các thành quả trước đó cho sản phẩm của mình, giữa sản phẩm mới và cũ không có nhiều sự khác biệt.

Gian lận, muốn che giấu việc đạo văn của mình bằng cách sao chép từ nhiều nguồn khác nhau, biên tập lại sao cho nội dung không còn tương đồng với bản gốc nữa.

Có ghi nguồn đầy đủ, chính xác nhưng lại không có dấu trích dẫn (“ ”) khi sao chép các từ, các câu, các đoạn từ nguồn vào sản phẩm của mình.[1]

## Lời cam kết

Tôi xin cam đoan đồ án này do chính tôi thực hiện. Tôi không sao chép, sử dụng bất kỳ tài liệu, mã nguồn của người khác mà không ghi rõ nguồn gốc. Tôi xin chịu hoàn toàn trách nhiệm nếu vi phạm.

Người cam kết

(kí và ghi họ tên)

Phạm Thị Như Hảo

# Lý thuyết

## Khái niệm

Gradient Boosting là một phương thức chuyển đổi các phương pháp học yếu thành các phương pháp học mạnh hơn. Phương thức này được thực hiện bằng cách xây dựng mô hình dữ liệu từ training data sau đó tạo ra một mô hình mới mà được sửa lỗi từ mô hình đầu. Gradient Boosting là một kĩ thuật sử dụng cho các vấn đề về Regression (hồi quy) và Classification (phân loại)[2].

Ý tưởng của thuật toán gradient boosting bắt nguồn từ Leo Breiman, cho rằng đây là một thuật toán tối ưu hóa các hàm loss function (hàm tính sự sai lệch giữa các dự đoán và dữ liệu thật).

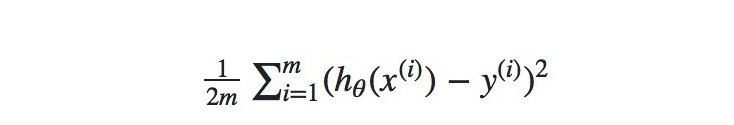
## Thuật toán

Thuật toán Gradient Boosting bao gồm 3 yếu tố:

* Tối ưu hóa hàm loss function
* Dự đoán các weak learners
* Một mô hình phụ được thêm vào mô hỉnh yếu để làm cho loss function nhỏ nhất.[3]

### Loss Function

Hàm loss function phụ thuộc vào vấn đề cần giải quyết, ví dụ như đối với thuật toán Linear Regression (hồi quy tuyến tính) thì hàm loss function có dạng:

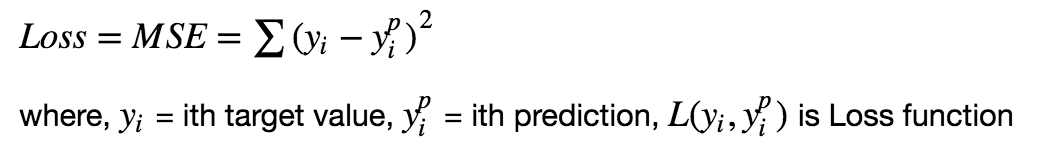


Với:

* m là tổng số dòng trong training data đầu vào
* hθ là hàm hypothesis của thuật toán
* x(i) là các feature
* y(i) là các giá trị đầu ra tương ứng với các feature x

Mặc dù các chức năng của hàm loss function tiêu chuẩn được hỗ trợ nhưng chúng ta phải làm nó khác đi bằng cách định nghĩa theo một công thức riêng.[3]

Ví dụ:



*Hình 1: Loss Function Gradient Boosting*

### Weak Learner (Người học yếu)

Chúng ta sử dụng decision tree (cây quyết định) như là một weak learner. Cụ thể, chúng ta sử dụng cây hồi quy (Regression tree) tìm ra các giá trị thực để phân tích. Các giá trị đầu ra có thể được thêm vào cùng nhau và cho phép kết quả đầu ra của các mô hình tiếp theo được thêm vào và sẽ chính xác với các dự đoán còn lại.

Cây được xây dựng bằng phương pháp tham lam nhằm mục đích chọn ra các điểm phân tích tốt nhất. Chúng ta sử dụng các cây lớn hơn từ cấp từ 4 đến 8.

Ta có thể hạn chế các weak learner bằng các cách đặc biệt như: ràng buộc số lượng tối đa của các layers (lớp), nodes (nút), split node (nút chia) hay leaf node (nút lá). Điều này dùng để chắc chắn các weak learner vẫn còn, nhưng ta vẫn có thể làm được việc này bằng phương pháp tham lam.[3]

### Additive Model (Mô hình phụ)

Cây cần thêm vào từng lần một và các cây hiện có thì không cần thay đổi. Ta sử dụng phương pháp giảm độ dốc (gradient descent) để giảm thiểu giá trị loss khi thêm cây vào mô hình.

Thông thường, chúng ta sử dụng cây gradient (gradient tree) để bỏ một tập các tham số. Chẳng hạn như các hệ số trong phương trình hồi quy hoặc trọng số trong mạng nơ ron (weight). Sau khi tính toán sai số hoặc mất mát (loss), các trọng số cần được cập nhật để giảm thiểu lỗi.

Ngoài các tham số, ta còn có các weak learner là các mô hình phụ hoặc các decision tree đặc trưng. Sau khi tính toán giá trị loss (độ lệch giữa các giá trị dự đoán và gài trị gốc), ta thực thi thủ tục gradient descent. Để giảm giá trị loss ta cần thêm một cây vào mô hình. Điều này được thực hiện bằng cách tham số hóa các cây, sau đó thay đổi các giá trị tham số đó theo đúng hướng.[3]

## Cải tiến cho thuật toán Gradient Boosting

Như chúng ta đã biết, gradient boosting là thuật toán sử dụng phương pháp tham lam để tìm ra mô hình tốt nhất mô tả dữ liệu, điều này làm cho việc mô hình bị overfitting nhanh chóng. Overfitting xảy ra khi mô hình mô tả rất tốt dữ liệu nhưng không tổng quát hóa được dữ liệu.[4]

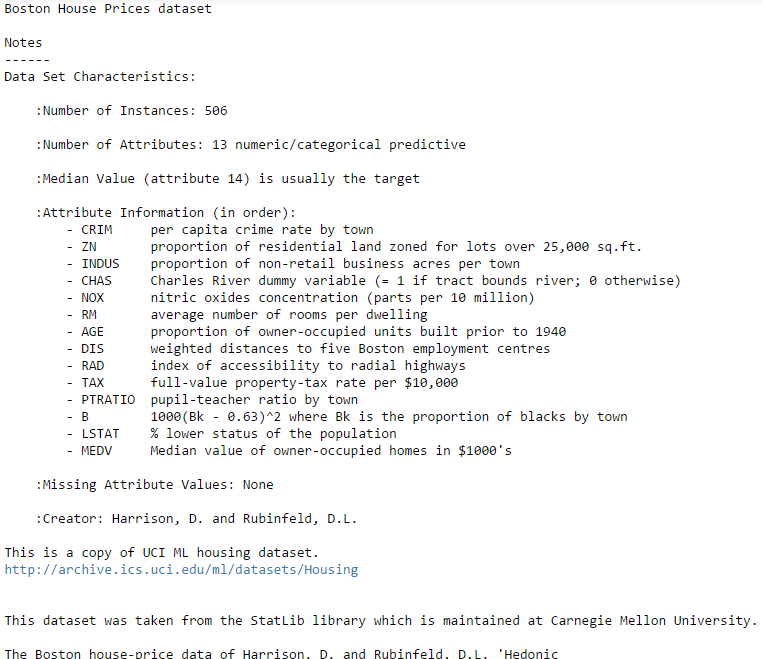
Các cách để cải tiến thuật toán gradient boosting gồm:

* Tree contraints (các ràng buộc về cây)
* Shrinkage / Weighted Updates (cập nhật các trọng số)
* Random sampling / Stochastic Gradient Boosting (Thống kê ngẫu nhiên)[4]

# Cài đặt thuật toán

## Datasets

Để cài đặt cũng như minh họa trực quan thuật toán, ta sử dụng data Boston House trong thư viện scikit – learn. Dataset này được dùng chủ yếu cho việc dự đoán giá nhà sử dụng thuật toán Regression.



*Hình 2: Chi tiết các column trong dataset Boston House in scikit - learn*

## Các thư viện được sử dụng:

### Numpy

Numpy là một gói công cụ hỗ trợ tính toán trên mảng (array), nhờ có numpy mà việc tính toán trên mảng trở nên nhanh hơn. Numpy còn hỗ trợ các phép tính trên toàn bộ mảng mà không cần duyệt qua các phần tử của mảng.

Khác với List, mảng trong numpy chỉ lưu trữ duy nhất một kiểu dữ liệu.

### Pandas

Trong python pandas là một thư viện mã nguồn mở, dùng trong việc thao tác với dữ liệu, được sử dụng rộng rãi trong lĩnh vực khoa học dữ liệu nói chung. Trong pandas dữ liệu được lưu trữ với cấu trúc riêng tên là Dataframe và cung cấp nhiều chức năng giúp cho việc thao tác và xử lí dữ liệu trở nên thuận tiện và đơn giản hơn.[5]

Trong quá trình cài đặt thuật toán Multivariate Linear Regression thư viện pandas được dùng để load dữ liệu từ dataset và xử lí.

### Scikit – learn

Scikit-learn là một bộ cung cấp các sự lựa chọn cho các thuật toán Supervised Learning và Unsupervised Learning. Đây là thư viện được xem là dễ dùng nhất. Nó được xây dựng dựa trên các thư viện và dữ liệu có sẵn như: Numpy, Scipy, Matplotlib, Pandas,v.v… nên dễ sử dụng cũng như kết hợp giữa chúng với nhau.[6]

Trong quá trình cài đặt thuật toán Gradient Boosting có sử dụng các phương thức như:

* sklearn.ensemble.GradientBoostingClassifier: hỗ trợ các phương thức thực thi thuật toán Gradient Boosting cho thuật toán Classification
* sklearn.ensemble.GradientBoostingRegressor: hỗ trợ các phương thức thực thi thuật toán Gradient Boosting cho thuật toán Regressor.
* sklearn.metrics.mean\_squared\_error: phương thức tính các loss function

### Matplotlib

Maplotlib là một thư viện trực quan trong python, giúp việc trực quan hóa dữ liệu đa nền tảng dựa vào cấu trúc mảng của numpy, nó cũng có thể hoạt động với ngăn xếp Scipy. Nó giúp chúng ta có thể hình dung một lượng lớn dữ liệu thông qua hình ảnh.[7]

Trong quá trình cài đặt thuật toán Gradient Boosting thư viện Matplotlib được dùng để mô tả trực quan dữ liệu trong thuật toán.

## Hướng cài đặt thuật toán

B1: import các thư viện cần dùng

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn import datasets

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn import ensemble

from sklearn.model\_selection import cross\_val\_predict

B2: load data

boston = datasets.load\_boston()

B3: Set các biến cần thiết cho việc train dữ liệu:

X = data.iloc[:,:-1]

y = data.iloc[:,-1]

x\_training\_set, x\_test\_set, y\_training\_set, y\_test\_set = train\_test\_split(X,y,test\_size=0.10, random\_state=42,shuffle=True)

B4: Tiến hành training dữ liệu và tính ra các loss fuction

params = {'n\_estimators': 500, 'max\_depth': 4, 'min\_samples\_split': 2,

'learning\_rate': 0.01, 'loss': 'ls'}

model = ensemble.GradientBoostingRegressor(\*\*params)

model.fit(x\_training\_set, y\_training\_set)

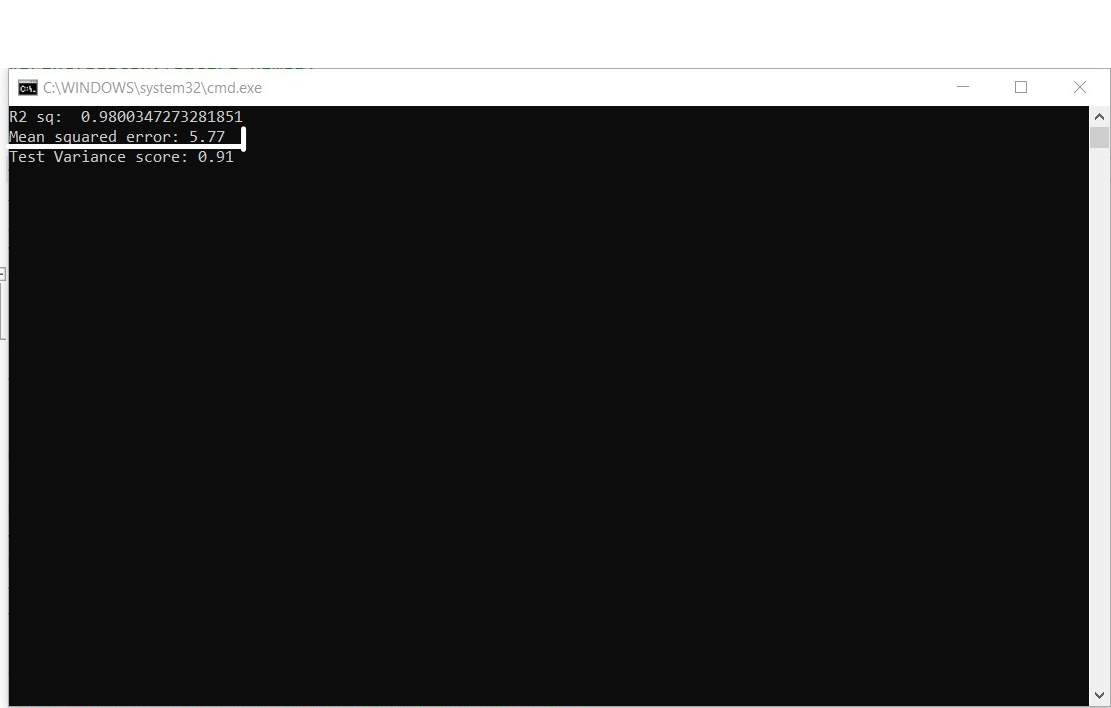
y\_predicted = model.predict(x\_test\_set)

print("Mean squared error: %.2f"% mean\_squared\_error(y\_test\_set, y\_predicted))

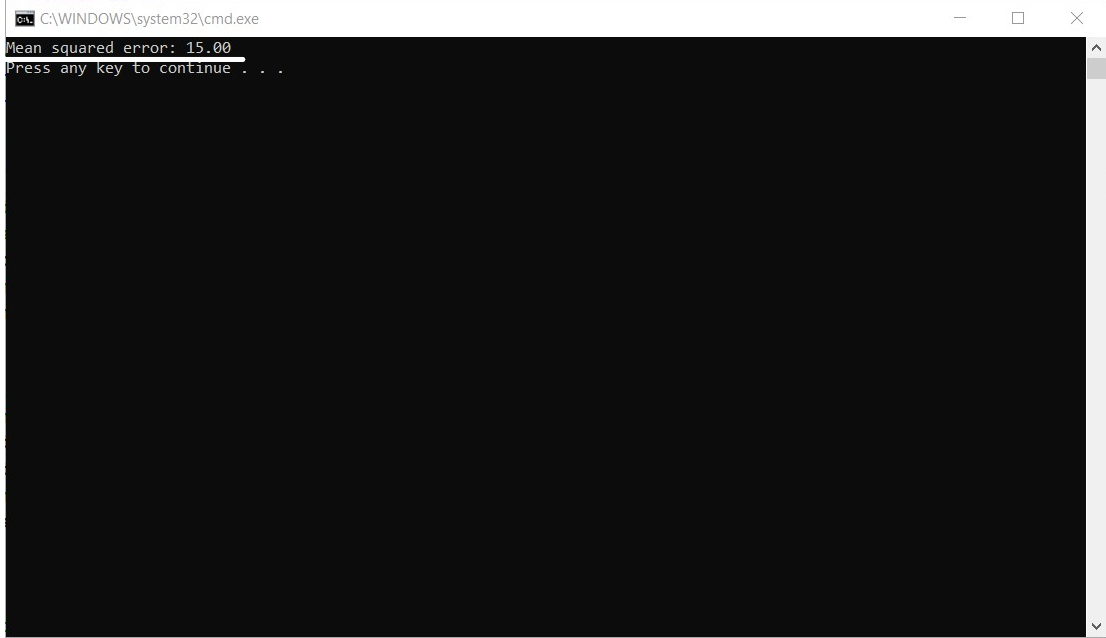
B4: Biểu diễn dữ liệu học được lên hệ tọa độ

# Đánh giá hiệu năng

Sau khi thực hiện train cùng tập dữ liệu bằng 2 phương thức là sklearn.linear\_model.LinearRegression và sklearn.ensemble.Regressor và tính loss function bằng phương thức sklearn.metrics.mean\_squared\_error thì nhận thấy loss fuction của thuật toán Gradient Boosting nhỏ hơn so với loss function khi train dữ liệu với thuật toán Linear Regression.



*Hình 3: Loss function khi train data với Gradient Boosting*



*Hỉnh 4: Loss function khi train data với Linear Regression*

Vậy nên ta có thể đánh giá hiệu năng hoạt động của thuật toán Gradient Boosting trên Regression khi train với dataset là Boston House trong scikit - learn là tốt hơn so với Linear Regression

# Kết luận

Đồ án đã được hoàn thành 85% mục tiêu đề ra.

Khó khăn: Do đề tài còn mới mẻ nên việc tìm hiểu gặp nhiều khó khăn, đòi hỏi kĩ năng tìm đọc tài liệu tiếng anh cao, khả năng coding và tư duy cao.

Ưu điểm: Hiểu được cơ chế làm việc của thuật toán Gradient Boosting.

Khuyết điểm: Chưa cài đặt tốt thuật toán, chỉ mới thực hiện trên một dataset. Chưa minh họa được các trường hợp overfitting.

# Tài liệu tham khảo:

[1]: <https://rces.info/sinh-vien-kinh-te-nckh/loi-dao-van-trong-nghien-cuu-khoa-hoc-va-cach-phong-tranh/>

[2]: Gradien Boosting (22 March, 2010)

Retrieved from: <https://en.wikipedia.org/wiki/Gradient_boosting>

[3]: Gradient Boosting Algorithm - Working and Improvements (21 Sep, 2018)

Retrieved from: <https://data-flair.training/blogs/gradient-boosting-algorithm/>

[4]: A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning (20 Nov, 2018)

Retrieved from: <https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/>

[5]: Free service for technical knowledge sharing

Retrieved from: <https://viblo.asia/p/huong-dan-su-dung-thu-vien-pandas-trong-python-XL6lAxaDZek>]

[6]: An Introduction to Scikit Learn: The Gold Standard of Python Machine Learning (Dec 26, 2018).

Retrieved from: <https://towardsdatascience.com/an-introduction-to-scikit-learn-the-gold-standard-of-python-machine-learning-e2b9238a98ab>

[7]: Python | Introduction to Matplotlib. (May 14,2018)

Retrieved from: <https://www.geeksforgeeks.org/python-introduction-matplotlib/>