**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**Báo cáo cuối kỳ**

HỌC PHẦN: Khai phá dữ liệu (Data Mining)

Tên đề tài:Dự đoán thời gian kẹt xe ở giao lộ

GVHD: Th.S Quách Đình Hoàng

**NHÓM : 06**

**Sinh viên thực hiện:**

Trần Quốc Tuấn 19133064

Trần Công Tuấn Mạnh 19133035

Trần Phát Đạt 19133018

Nguyễn Lâm Sơn 19133050

**TP.HCM, ngày 25 Tháng 05 năm 2022**

Mục Lục

[1. Tóm tắt 2](#_Toc106109327)

[2. Giới thiệu 2](#_Toc106109328)

[3. Dữ liệu 3](#_Toc106109329)

[4. Phương pháp 6](#_Toc106109330)

[5. Thực nghiệm, kết quả, và thảo luận 6](#_Toc106109331)

[5.1 Linear Regression 6](#_Toc106109332)

[5.2 K-nn model 7](#_Toc106109333)

[5.3 Random forest 9](#_Toc106109334)

[5.4 Gradient boosting 10](#_Toc106109335)

[6. Kết luận 12](#_Toc106109336)

[7. Đóng góp 13](#_Toc106109337)

[8. Tham khảo 13](#_Toc106109338)

# 

# 

# 1. Tóm tắt

Hàng ngày, mỗi khi đi làm, đi học về, chúng ta thường bị tắc đường, thậm chí là kẹt tới vài chục phút mới ra được quãng đường ngắn ngủi. Mà đèn đỏ thì quá ít thời gian, có những chỗ còn dưới 20 giây, trong khi đó trên đường có tới đoàn người dày đặc khiến chúng ta chưa kịp đi thì đèn xanh đã hết, lại phải chờ đèn đỏ tiếp theo. Vì vậy, mục đích của bài toán này, nhóm muốn dự đoán được thời gian tác đường trong bao lâu để có thể góp phần chỉnh sửa thời gian của đèn giao thông một cách hợp lý, nếu được, có thể điều hướng lại những con đường bị tắc sang những hướng đi khác ít tắc hơn.

# 2. Giới thiệu

Tất cả chúng ta đều đã bị kẹt trước đèn giao thông, chỉ được dành vài giây để đi qua ngã tư, phía sau đoàn người đông đúc với những người đi làm khác. Nếu có thể giúp các nhà quy hoạch thành phố và chính phủ dự đoán trước các điểm nóng về giao thông và giảm bớt căng thẳng khi dừng và đi của hàng triệu người đi làm như chúng ta.

Input của bài toán là tập các giá trị về thời gian dừng chờ đèn đỏ tại các ngã tư, ngã bảy, khoảng cách dừng xe của số phần trăm lượng xe….

Nhóm sử dụng các thuật toán Linear Regression, K-nn, Random Forest, Gradient boosting, … để dự đoán thời gian kẹt xe và khoảng cách bao xa từ số lượng xe đó tại các giao lộ.

# 3. Dữ liệu

Tập dữ liệu đến từ cơ sở quan sát các giao lộ giao thông ở mỹ, nhưng dataset đã được nhóm lại nên ở đây chỉ có 4 thành phố ở mỹ, nó ghi lại lượng xe đi qua các giao lộ mỗi giờ.

Tập dữ liệu có 2 phần: train và test

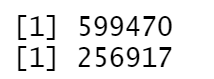
* Tập train có 856387 dòng dữ liệu.
* Tập test có 1921357 dòng dữ liệu.

Bài toán ghi kết quả lượng xe đi qua các giao lộ mỗi giờ của nhóm quan sát thực tế. Tập dataset chứa dữ liệu trên 4 thành phố là Atlanta, Boston, Chicago & Philadelphia.

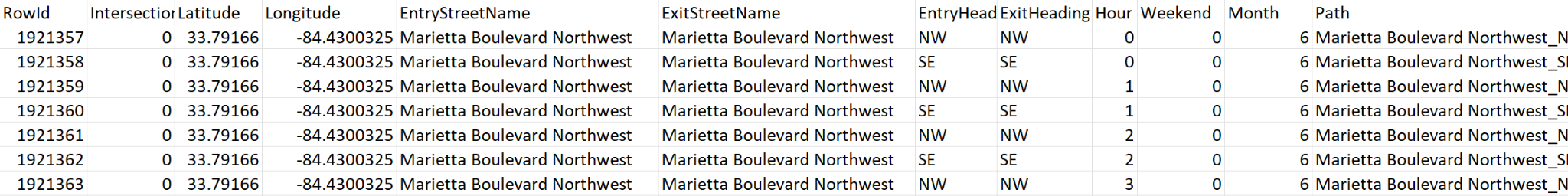
Mục tiêu của bài toán này là dự đoán:

* **TotalTimeStopped\_p20:** Tổng thời gian dừng ở giây thứ 20
* **DistanceToFirstStop\_p20:** Khoảng cách đến điểm dừng đầu tiên ở giây thứ 20
* **TotalTimeStopped\_p50:** Tổng thời gian dừng ở giây thứ 50
* **DistanceToFirstStop\_p50:** Khoảng cách đến điểm dừng đầu tiên ở giây thứ 50
* **TotalTimeStopped\_p80:** Tổng thời gian dừng ở giây thứ 80
* **DistanceToFirstStop\_p80:** Khoảng cách đến điểm dừng đầu tiên ở giây thứ 80

Ta cần một tập validation để đánh giá mô hình trước khi sử dụng tập test. Vậy nên nhóm sẽ chia dữ liệu thành 7 phần train và 3 phần validation. Sau khi chia, ta có được 599470 dòng tập train và 256917 dòng cho tập valida.



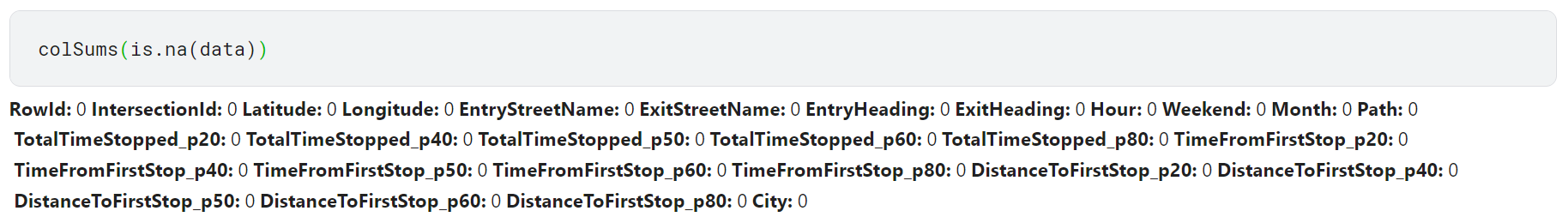
Để làm hiểu rõ hơn về tập dữ liệu, nhóm sẽ hiển thị một số dòng đầu của tạp train như sau:



Một số dòng tập test:

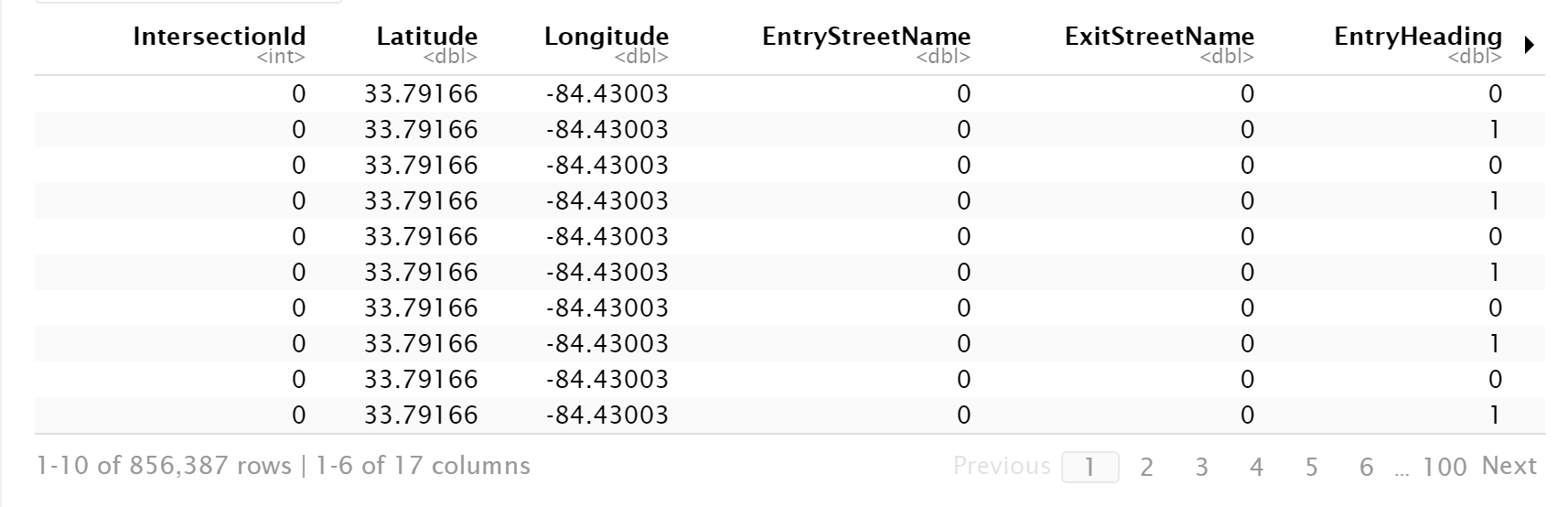


Điều mà trước khi dự đoán bất kì mô hình nào chúng ta càn phải làm đó là kiểm tra giá trị NA. Dưới đây, nhóm sử dụng hàm isna để kiểm tra giá trị NA trong tạp dữ liệu:



Có thể thấy không có cột nào có giá tị NA. Tiếp theo là xử lý các giá trị category, bởi hình mô hình cần các giá trị số để có thể dự đoán, ta sẽ chuyển chúng thành kiểu number bằng hàm labelencoder.

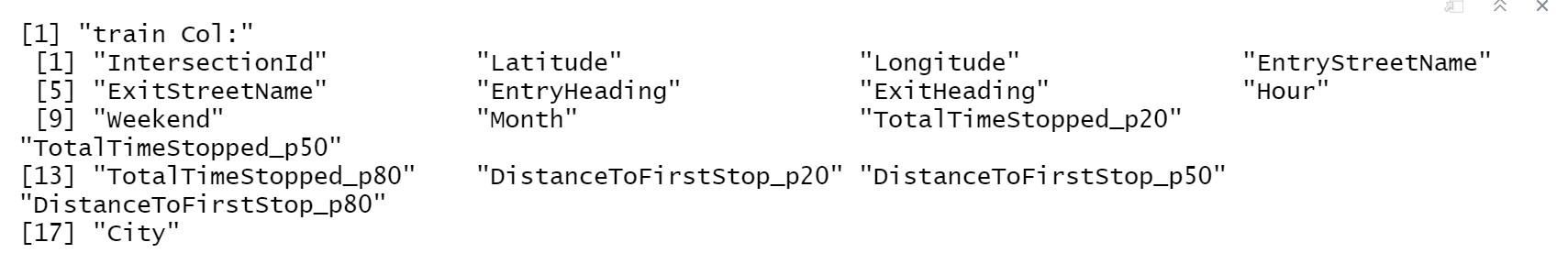
Dưới đây là dữ liệu sau khi đã tiến hành chuyển đổi thành number.



Có thể thấy, tất cả các giá trị category như `EntryStreetName, ExitStreetName, EntryHeading…` đã chuyển đổi thành kiểu số.

Cùng với hai việc đã làm trên kia. Trong dữ liệu cũng có một cột ảnh hưởng lớn tới kết quả dự đoán nhưng lại không có ý nghĩa thiết thực đó là cột `Rowid`, vì vậy chúng ta sẽ bỏ cột này trước khi quá trình training mô hình được bắt đầu.

Sau khi loại bỏ những biến không cần thiết cho việc đào tạo mô hình, ta còn lại những cột sau đây trong tập test:

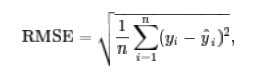


Vậy là việc tiến hành tiền xử lý dữ liệu đã coi như hoàn thành, tiếp theo đây, nhóm sẽ đi vào chọn lọc các phương pháp đánh giá.

# 4. Phương pháp

* **Các phương pháp đánh giá**

Trong competition này yêu cầu người tham gia đánh giá độ tốt mô hình dựa vào root mean square error (rmse) nên nhóm sẽ dùng độ đo này đánh giá các mô hình của nhóm.

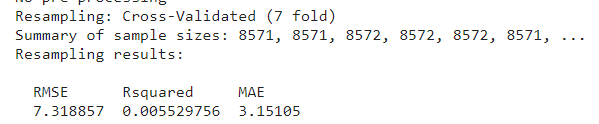


# 5. Thực nghiệm, kết quả, và thảo luận

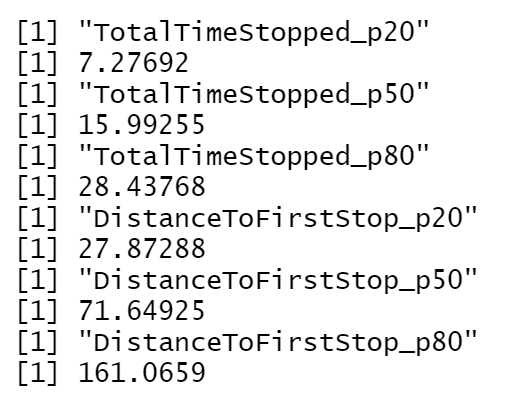
* Các thuật toán được áp dụng: linear, knn, random forest, gradient boosting.
* Dựa vào độ phù hợp của model mà chọn ra model tốt nhất.

## 5.1 Linear Regression

Ở đây, với mô hình này nhóm sử dụng cross validation với 7 fold. Sau khi thử với việc sử dụng method = ‘cv’ có 5 với 7 fold, nhóm nhận ra rằng: dù có chia fold thì linear không phụ thuộc vào mức dao động của dữ liệu.

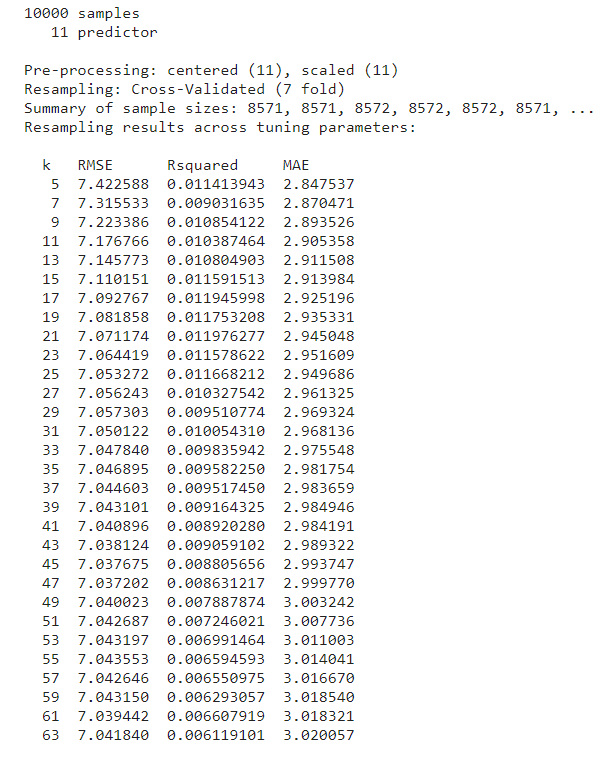


Kết quả đào tạo của mô hình này như bên dưới:

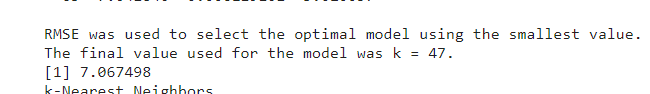


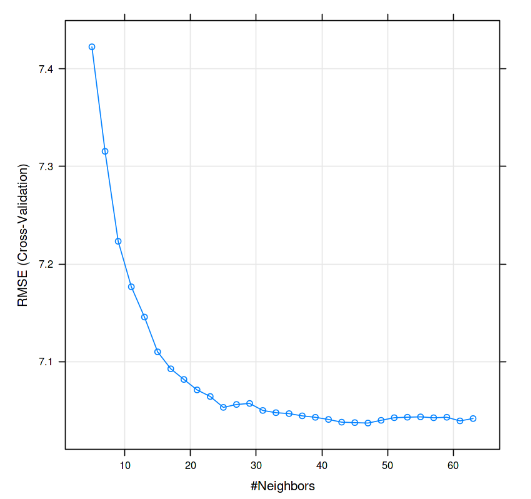
## 5.2 K-nn model

Một đoạn run code trong knn, cụ thể là total\_p20:



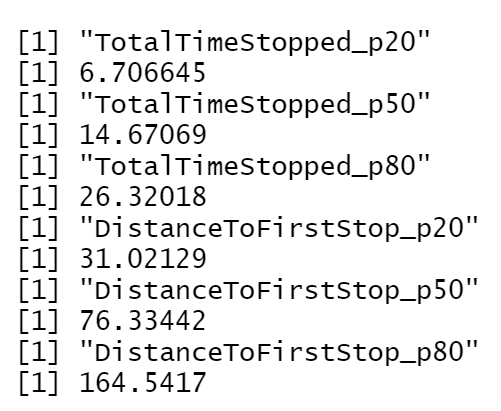
Mẫu ban đầu lấy là 10000, chia làm 7, Độ rộng để chạy k là 30, sau đó nó lấy kết quả tốt nhất:





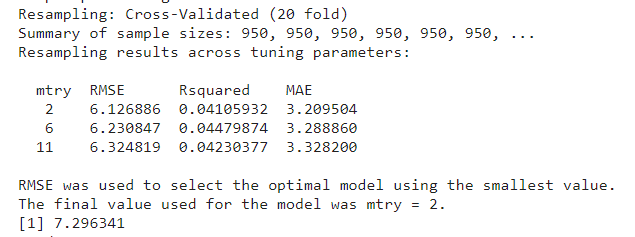
Biểu diễn bằng đồ thị:

Kết quả sau khi đào tạo mô hình này:

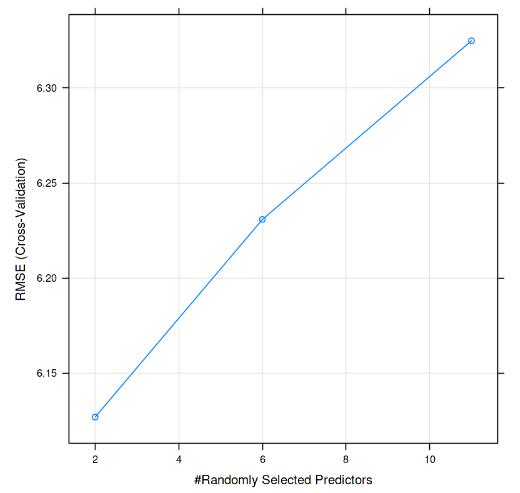


## 5.3 Random forest

Một đoạn biểu diễn luồng:



Ở đây, vì mức độ lâu của tập mà mình chỉ chia làm 20 fold, ở đây ta thấy mtry=2 cho kết quả tốt nhất, nên nó lấy là 2.

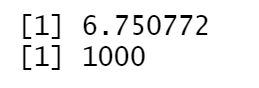
Biểu diễn bằng đồ thị:

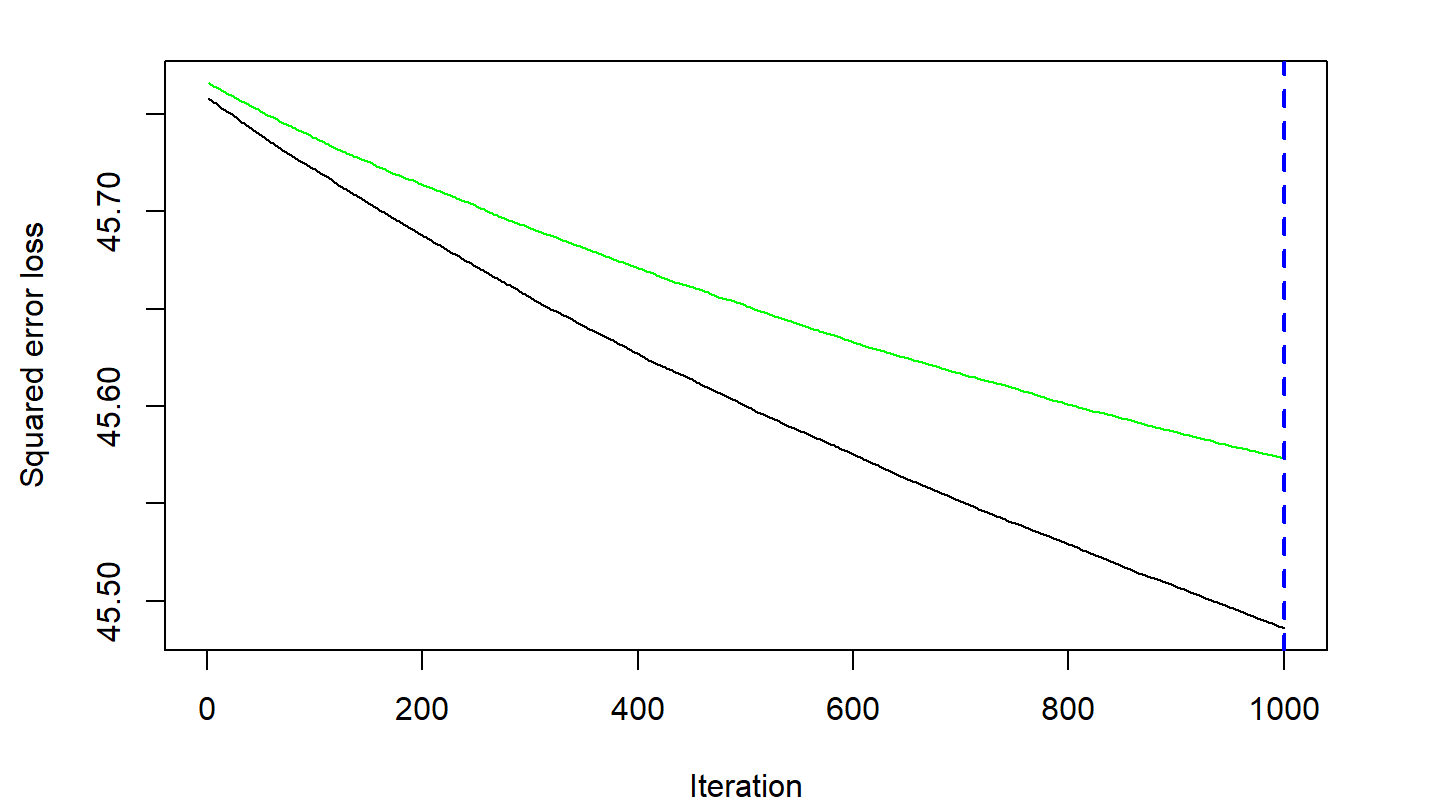
## 5.4 Gradient boosting

Cài đặt mặc định trong gbm có tỷ lệ học tập (learning rate) là 0,001. Đây là một tỷ lệ học tập rất nhỏ và thường yêu cầu một số lượng lớn cây để tìm ra MSE tối thiểu. Tuy nhiên, gbm sử dụng số lượng cây mặc định là 100, điều này là không đủ trong trường hợp này. Do đó, ta sẽ sử dụng lên đến 1000 cây. Độ sâu mặc định của mỗi cây (depth) là 1, có nghĩa là chúng ta đang tập hợp một loạt các gốc cây. Cuối cùng, ta cũng sẽ sử dụng cv.folds để thực hiện xác thực chéo 5 lần.

Ở đây, chúng ta sẽ sử dụng 1 biến phản hồi duy nhất `TotalTimeStopped\_p20` để kiểm tra mô hình gbm này:

Kết quả ta có RMSE 6,75 với sai số đang ở cây số 1000:

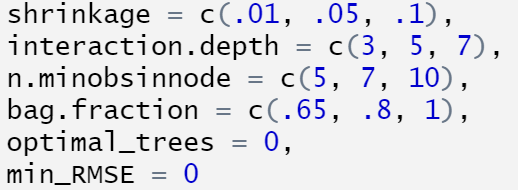




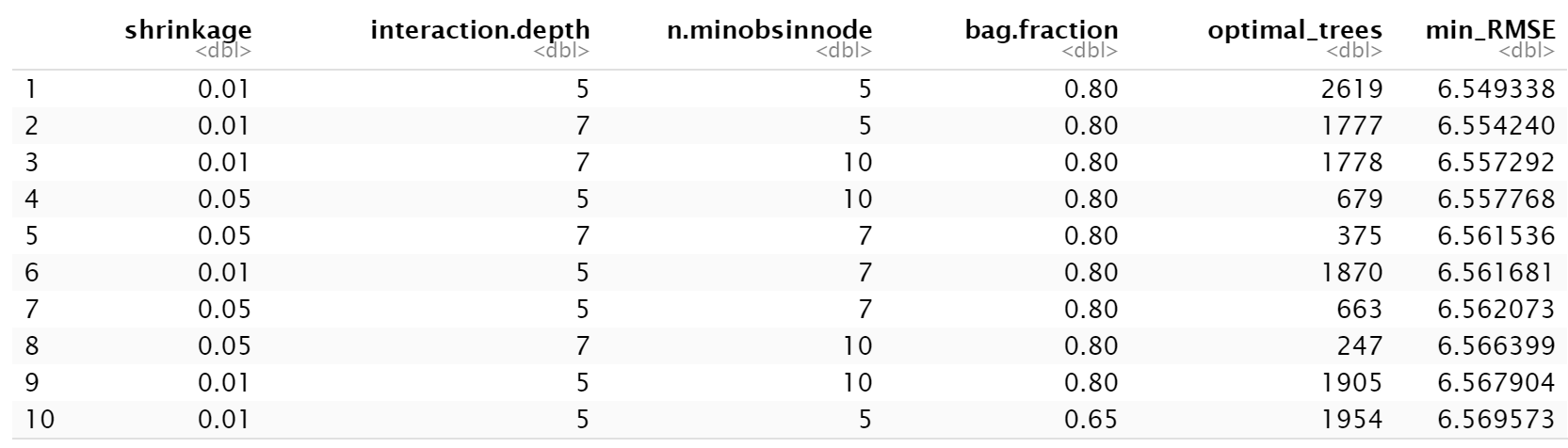
Mô hình mất khoảng 2 phút để chạy và kết quả cho thấy rằng hàm mất mát MSE được giảm thiểu với 10000 cây.

Bởi vì RMSE của mô hình này hơn 6,7 giây cho biến phản hồi y `TotalTimeStopped\_p20` nên ta cần phải thử các tham số khác để tìm ra mô hình dự đoán tốt nhất. Ta cần tạo một hyperparameter grid để áp dụng một lần nhiều giá trị khác nhau.

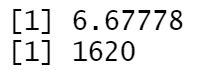
Đầu tiên, cần tạo một lưới các hyperparameter với các giá trị bên dưới:



Sau đó, áp dụng vào đào tạo mô hình, ta được 81 mô hình với các tham số và kết quả RMSE khác nhau, sắp xếp từ min RMSE cho tới max, ta sẽ chọn các tham số của mô hình có RMSE nhỏ nhất, và áp dụng vào mô hình cuối cùng để dự đoán.

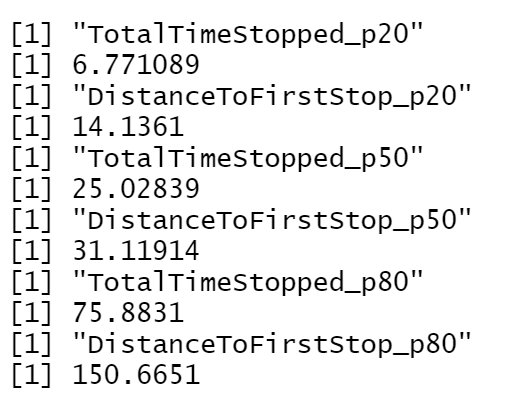


Với các tham số vừa tìm được, nhóm sẽ áp dụng vào để đào tạo các mô hình cho các biến phản hồi như đầu bài.

Sau khi đào tạo mô hình cuối cùng, ta được mô hình với kết quả:

RMSE của mô hình này là 6.677, thấp hơn so với lúc chưa dùng tham số tìm được.

Vậy ta sẽ áp dụng các tham số này để đào tạo mô hình với các biến phản hồi khác.



Kết quả sau khi đào tạo cho các biến phản hồi khác nhau

# 6. Kết luận

Sau quá trình thử nghiệm và đào tạo 4 mô hình trên, nhóm rút ra được với bài toán này,thuật toán Gradient Boosting có hiệu suất cao nhất và cho kết quả sát thực tế hơn những ô hình còn lại.

Nếu nhóm có nhiều thời gian hơn hoặc nhiều tài nguyên tính toán hơn, nhóm sẽ sử dụng nhiều dữ liệu hơn trên các mô hình, để có cái nhìn tổng quan nhất trên các thuật toán.

# 7. Đóng góp

|  |  |
| --- | --- |
| Trần Công Tuấn Mạnh | Gradient Boosting model,  Label Encoder |
| Trần Phát Đạt | Linear model, Data preprocessing |
| Nguyễn Lâm Sơn | Overview, Data preprocessing |
| Trần Quốc Tuấn | KNN, Random forest model |

# 8. Tham khảo

<http://uc-r.github.io/gbm_regression>

<https://machinelearningcoban.com/tabml_book/ch_model/random_forest.html>

<https://bookdown.org/tpinto_home/Regression-and-Classification/k-nearest-neighbours-regression.html>

<https://www.scribbr.com/statistics/linear-regression-in-r/#:~:text=Linear%20regression%20is%20a%20regression,total%20error%20of%20the%20model>