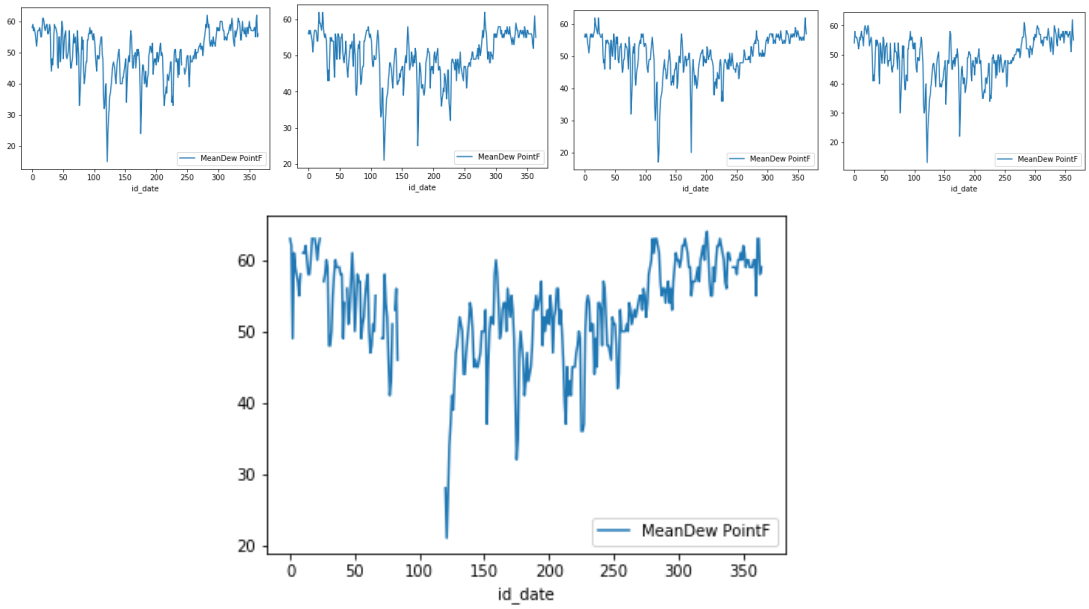
Predict the net rate of bike renting

1. Phân tích, tiền xử lý dữ liệu

(Xem notebook data_processing.ipynb)

1.1.Tiền xử lý dữ liệu

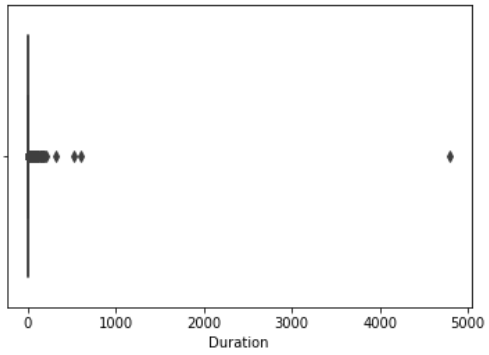
- Dữ liệu stations: loại bỏ các stations cũ với các id bằng 23, 25, 49, 69, 72 (vì đã được chuyển thành 85, 86, 87, 88, 89 tương ứng)
- Dữ liệu về weathers (thời tiết): Trong bảng weathers có chứa rất giá trị **null** (NaN). Ta cần làm sạch dữ liệu này. (Data bao gồm thời tiết từ ngày 01/09/2015 – 31/08/2015 tại 5 thành phố mà các stations được đặt)
  2 trường với giá trị NaN nhiều nhất là **Max Gust SpeedMPH** và **Events**. Tuy nhiên **Events** là trường categorical, giá trị NaN biểu thị thời tiết bình thường (không Rain, không Fog…), ta thay NaN của Events bởi "Normal". Cột **Max Gust SpeedMPH** nhận giá trị thực tuy nhiên giá trị NaN nhiều nên ta bỏ cột này đi
  Đối với các cột còn lại có ít giá trị NaN. Xét NaN tại 1 ngày và 1 thành phố nào đó, ta thực hiện việc điền giá trị này bằng cách lấy trung bình giá trị ngày tương ứng tại các thành phố còn lại. Sở dĩ làm được như vậy vì quan sát được đồ thị biến thiên theo 1 thuộc tính thời tiết tại các thành phố là có dạng giống nhau.



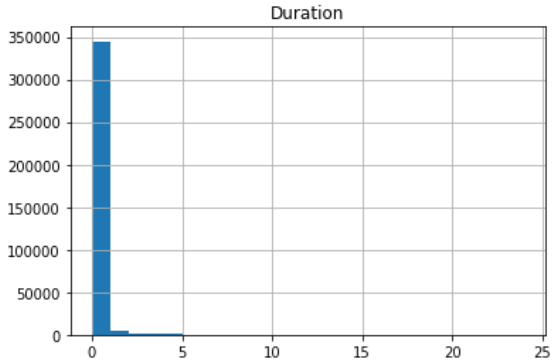
Hình 1. Giá trị theo thời gian trường MeanDew PointF tại 5 thành phố, giá trị NaN được tính bằng trung bình ngày tương ứng tại 4 thành phố còn lại.

- Dữ liệu về các trips (các lần thuê): Hình 2 chỉ ra khoảng thời gian thuê duration (theo giờ). Nhận thấy có 1 trip thời gian quá khác biệt so với các trip còn lại => outlier => loại bỏ khỏi dataset. Hình 3 chỉ ra hầu hết các trips có duration nhỏ hơn 1 giờ (tỉ lệ số lượng trips này chiếm đến 97.35%)

Từ bảng trips, ta thực hiện xử lý tạo ra bảng df bao gồm 2 keys là id_station, id_hour cùng với các trường số lượng người đến thuê xe (num_trips_start), số lượng đến trả xe (num_trips_end) và net rate = num_trips_end – num_trips_start tương ứng.



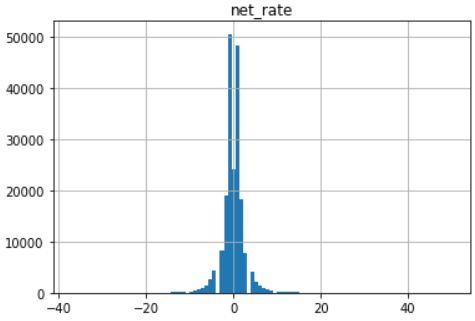
Hình 2



Hình 3

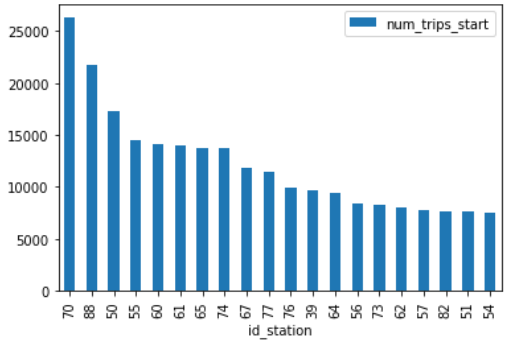
1.2. Phân tích dữ liệu

- Net rate có biến thiên nhiều hay không? Hình 4 là 1 histogram cho thấy phân bố giá trị net rate tập trung gần 0 rất nhiều. Tức là số lượng đến trả xe so với số lượng đến thuê xe tại 1 station đa số là không quá chênh lệch nhau nhiều

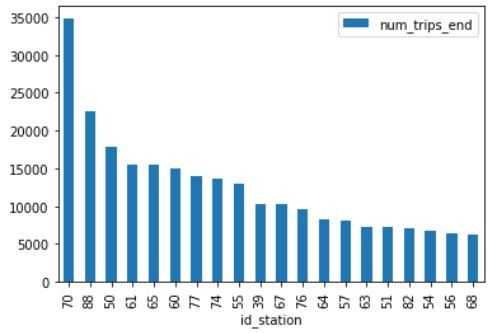


Hình 4.

- Stations nào được ưa thích nhất? Hình 5, 6 chỉ ra các stations theo thứ tự giảm dần có số lượng đến thuê/trả xe nhiều nhất. Một điểm đặc biệt có thể thấy là 1 station có số lượng thuê nhiều thì cũng có số lượng trả nhiều. Điều này có thể giải thích đơn giản là một người xuất phát từ 1 station sau 1 khoảng thời gian sẽ lại quay trở về station đó. Theo thống kê số lượng trips đến từ người dùng thường niên ‘Subsciber’ chiếm đến 87.59% tổng số lượng trips. Nghĩa là đây là những người dùng thường xuyên sử dụng dịch vụ thuê xe có thể để đi làm/đi học

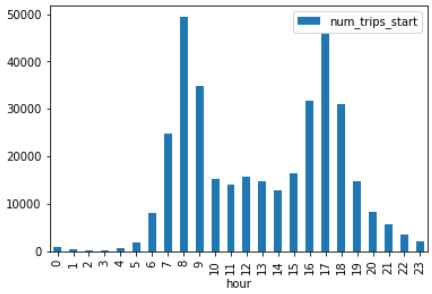


Hình 5

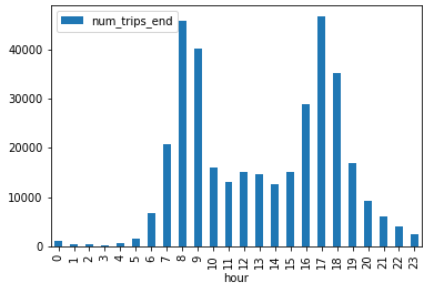


Hình 6

- Giờ nào trong ngày hay được thuê/trả xe nhiều nhất? Hình 7, 8 chỉ ra điều đó. Ta có thê thấy rằng hầu hết tập trung vào tầm sang hoặc chiều tức là lúc bắt đầu đi làm và tan làm. Càng chứng tỏ việc thuê xe hầu hết nhằm mục đích như 1 phương tiện di chuyển đi làm/đi học (lưu ý vì đa số duration trips < 1h nên giờ mà thuê xe nhiều thì cũng trả xe nhiều => 2 đồ thị hoàn toàn hợp lý)

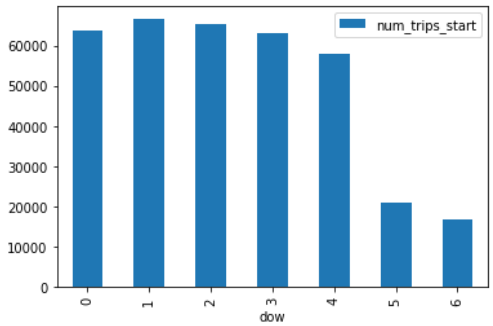


Hình 7

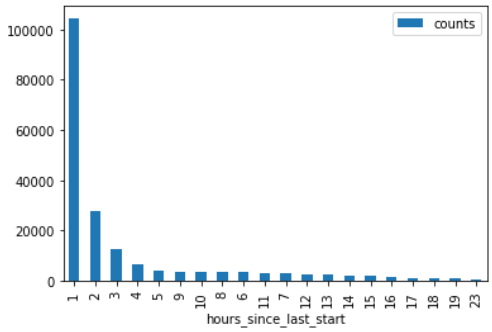


Hình 8

- Ngày nào trong tuần hay được thuê/trả xe nhiều nhất? Hình 9 càng chứng tỏ nhận định những người thuê xe thuộc đối tượng đi làm/đi học là đúng, vì số lượng tập trung thuê ngày trong tuần nhiều hơn cuối tuần. Ngày cuối tuần họ có thể đi du lịch hoặc đi chơi.



Hình 9



Hình 10

- Tại 1 station ở 1 thời điểm mà có sự kiện thuê xe, sau bao lâu thì lại có sự kiện thuê/trả xe tiếp theo? Hình 10 chỉ ra là đa số trong khoảng thời gian 1, 2, 3 giờ.

2. Xây dựng model

Ta join các bảng về stations, weather và bảng net rate vào với nhau. Bảng joined này gồm các thông tin về giờ, ngày cụ thể, các thông tin về station, các thông tin về thời tiết của ngày đó, số lượng xe được thuê (num_trips_start), số lượng xe được trả (num_end_start) và net rate.

Ta có tập training set D = $\{(x_t, y_t) \mid t=0,\ldots T\}$, trong đó y_t là net rate tại giờ thứ t, x_t là vector input features.

Feature Engineering

Tại giờ thứ t, ta chỉ quan sát được dữ liệu thuê/trả xe từ giờ thứ 0 đến t, ta muốn dự đoán net rate trong giờ thứ t+1. Như vậy cần xây dựng vector features x_{t+1} để dự đoán net rate y_{t+1} cho giờ t+1. Ta thấy rằng thông tin của station là cố định, thông tin của thời tiết tại giờ t+1

cũng hoàn toàn có thể biết trước (vì thông tin thời tiết đang xét theo ngày và thời tiết hoàn toàn có thể dự báo trước được). Nên x_{t+1} trước hết bao gồm thông tin thuộc 2 loại đó.

Riêng chỉ có features số lượng thuê/trả xe thì là không quan sát được tại thời điểm t+1. Ta lấy giá trị những features này từ trong tập đã quan sát được để biểu diễn trong x_{t+1}, gọi là "phép lấy trễ" (lag) - num_trips_start_lag, num_trips_end_lag. Dựa vào phân tích trong phần 1.2: việc thuê/trả xe có chu kì theo giờ trong ngày, ngày trong tuần, 2 lần thuê/trả liền nhau cách nhau từ 1, 2, 3 tiếng. Ta sẽ thực hiện lấy trễ 1, 2, 3 giờ và lấy giá trị cùng giờ tại ngày hôm trước, hôm trước nữa … kéo dài cho đủ 1 tuần. Như vậy ta thêm giá trị số lượng thuê/trả xe tại các thời điểm t, t-1, t-2, t+1 – 24, t+1 – 24*2, …, t+1-24*6 vào trong vector x_{t+1}.

Ngoài ra còn thêm features đếm số giờ tính từ thời điểm cuối cùng mà có xe được thuê/trả đến thời điểm t+1. Nếu giá trị này càng lớn có vẻ như mất 1 khoảng thời gian khá lâu mà station đó chưa có hoạt động thuê/trả xe nào => station đó không còn là điểm thu hút khách.

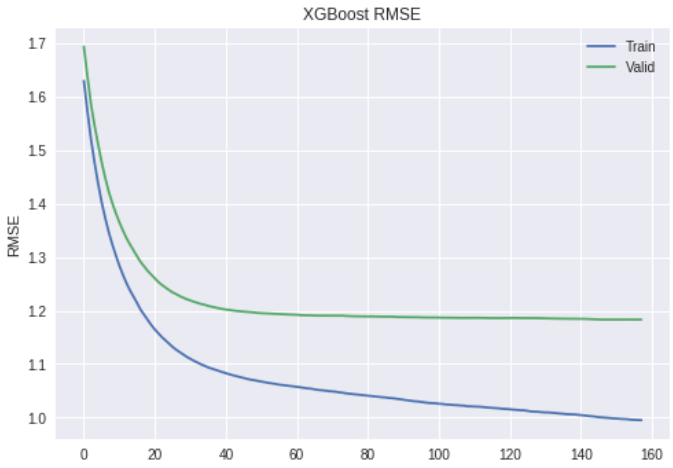
Các features giờ trong ngày (hour in day) hay ngày trong tuần (day of week) cũng được thêm vào để biểu diễn rõ ràng hơn về quy luật chu kì như đã đề cập.

Training model

Sau khi tạo được training dataset, ta học một regression model từ tập dữ liệu đó. Có rất nhiều model để giải quyết bài toán này. Ở đây, mô hình XGBoost là một 1 mô hình phân loại/hồi quy mạnh sẽ được sử dụng để build regressor.

3. Kết quả

Sau khi thực hiện tiền xử lý, ta thu được 1 dataset bao gồm 610032 examples (= 8592 hour * 71 stations). Chia 70% số giờ đầu tiên làm training set, 15% cho validation set và 15% số giờ còn lại cho test set.

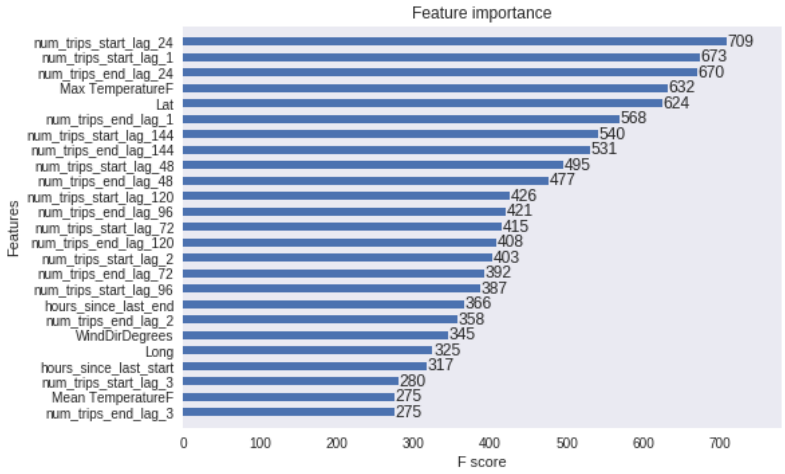


Hình 11. RMSE on training and validation set

Kết quả RMSE trên test set: 1.20435

Hình 12 minh họa 25 features quan trọng nhất có ảnh hưởng đối với kết quả dự đoán. Ta có thể thấy rằng đúng như phân tích các giá trị số lượng thuê/trả xe tại các thời điểm 1 giờ trước

(_lag_1), cùng giờ ngày hôm trước (lag_24), hay cùng giờ cùng ngày tuần tước (24*6 = 144) có ảnh hưởng đến lưu lượng thuê/trả xe tại thời điểm hiện tại. Hay các features về nhiệt độ (Temperature) trong ngày, hay vị trí của station (Lat, Long) cũng có ảnh hưởng mạnh.



Hình 12. Features Importance

4. Thảo luận

Hướng giải quyết được đề xuất trong báo cáo này sử dụng theo hướng tiếp cận feature engineering tức là tự tạo ra những feature có ý nghĩa đến đầu ra dự đoán để biểu diễn cho vector đầu vào của model. Cách tiếp cận này sau đó đi kèm với việc sử dụng model tree based để training/prediction thường cho kết quả tốt. Việc tạo ra những feature có ý nghĩa thì mô hình học càng tốt. Trong cách giải quyết của báo cáo này vẫn chưa tạo ra được những features mà encode được 1 số thông tin có ý nghĩa sau:

- Tương tác giữa các station: (start_station, end_station) là điểm xuất phát và điểm đến của cùng 1 trip. Bằng cảm nhận ta có thể thấy rằng khi số lượng xe được thuê tại 1 station cao thì khả năng số lượng xe trả tại các station liên kết mạnh với station kia cũng cao (hình dung giống như "social tie" trong social network vậy)
- Có thể thấy dữ liệu net rate như dữ liệu dạng time series. Việc encoding dữ liệu 1 giờ, 1 ngày hay 1 tuần trước cũng chưa hẳn là những pattern ẩn trong time series.

Có thể sử dụng những mô hình giải quyết với dữ liệu dạng time series để làm việc bài toán này.

5. Kết luận

Trong báo cáo này không đề cập đến việc so sánh các regression model khác nhau, vì để tìm ra được model tốt nhất thì cần thử nghiệm khá nhiều. Điều nhấn mạnh trong báo cáo là việc phân tích, quan sát dữ liệu để tạo ra những features có ý nghĩa làm tăng tính dự đoán cho model.