

**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**DỰ ĐOÁN NHU CẦU ĐI XE TAXI MÀU VÀNG
THEO TỪNG KHU VỰC TRONG THÀNH PHỐ
NEW YORK**

Sinh viên thực hiện		
STT	Họ tên	MSSV
1	Võ Đình Tứ	18521589

TP. HỒ CHÍ MINH – 12/2020

1. GIỚI THIỆU

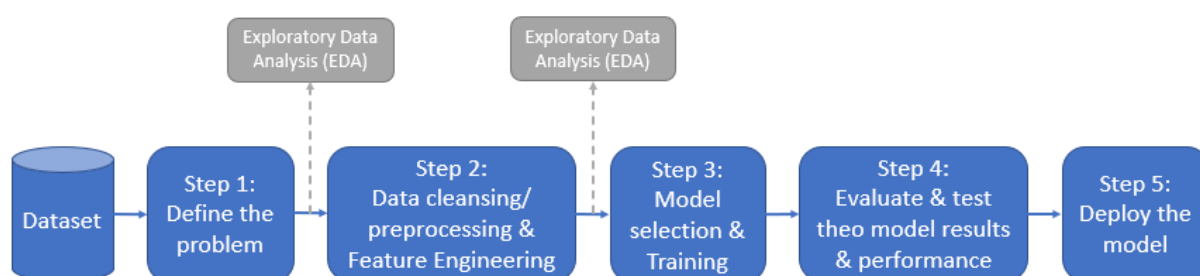
- Ngày nay trên thế giới, kể cả Việt Nam các loại hình xe TAXI để có đầy đủ trên tất cả mọi vùng miền của đất nước, nó chính công cụ đi lại rất phổ biến đối với mỗi người trên thế giới. Ở một nơi rất xa hoa của nước Mỹ - thành phố New York, các loại hình Taxi này vô cùng phổ biến và phát triển, trong đó có một xe taxi màu vàng, nhu cầu đi Taxi vàng của người dân rất cao, điều này được minh chứng qua hàng trăm bộ dữ liệu được records cho đến nay. Chính vì điều này, tôi đã nảy sinh một bài toán dự đoán về nhu cầu đi Taxi của khách trong từng khu vực trong cho những tài xế xe Taxi vàng có thể đoán và nắm bắt những nơi có nhu cầu cao hoặc thấp.

- Để hiểu và đi làm và hoàn thành tốt bài toán dự đoán này, tôi đã lấy ý tưởng từ một bài toán dự đoán cổ phiếu sử dụng những công cụ máy học, học sâu trong khoa học dữ liệu. Tôi đã dùng đến mô hình tiếp cận trong phân tích dữ liệu để thăm dò, làm sạch dữ liệu, xây dựng mô hình, đánh giá hiệu suất,... Quy trình sẽ được tôi trình bày rõ ràng theo bộ cục như ở dưới

- Kết quả tôi thu được có tiêu biểu nói đến như là: những bộ dataset của từng khu vực trong thành phố New York đã chuẩn hóa, thực nghiệm mô hình hoàn thiện cũng như đánh giá được hiệu suất của nó.

2. NỘI DUNG

Khuyến nghị tiếp cận theo quy trình hoặc các bước thực hiện phân tích dữ liệu đã được học.



Hình 1. Quy trình PTDL.

2.1 Giới thiệu dataset

- **Tên dataset:** 2020 Yellow Taxi Trip Data (January - June)
- **Kích thước dataset:** 16 847 778 samples (records) và 18 features.
- **Mô tả tổng quan:** Bộ dữ liệu ghi kết quả của những chuyến xe Taxi màu vàng từ tháng 1 đến tháng 06 năm 2020 hoạt động trong thành phố New York.

- Mô tả dataset:

Tên features	Mô tả chi tiết
VendorID	Mã chỉ ra nhà cung cấp TPEP đã cung cấp hồ sơ. 1= Creative Mobile Technologies, LLC; 2= VeriFone Inc.
tpep_pickup_datetime	Ngày và giờ khi đồng hồ bắt đầu tính khi nhận khách
tpep_dropoff_datetime	Ngày và giờ khi đồng hồ bắt đầu tính khi trả khách.
Passenger_count	Số lượng hành khách trên xe. Đây là giá trị do tài xế nhập vào.
Trip_distance	Khoảng cách chuyển đi được tính báo cáo của đồng hồ xe tắc xi (miles)
PULocationID	Khu vực taxi TLC mà đồng hồ tính tiền đã bắt đầu hoạt động (khu vực nhận khách)
DOLocationID	Khu vực taxi TLC mà đồng hồ tính tiền đã kết thúc (khu vực trả khách)
RateCodeID	Mã giá cuối cùng có hiệu lực vào cuối chuyến đi. 1= Standard rate 2=JFK 3=Newark 4=Nassau or Westchester 5=Negotiated fare 6=Group ride
Store_and_fwd_flag	Đặt flag này để biết liệu bản ghi chuyến đi có được lưu trong bộ nhớ xe hay không trước khi gửi đến nhà cung cấp, hay còn gọi là “lưu trữ và chuyển tiếp”, vì xe không có kết nối với máy chủ. Y= lưu trữ và chuyển tiếp chuyến đi N= không lưu trữ và chuyển tiếp chuyến đi
Payment_type	Mã số biểu thị hình thức hành khách thanh toán cho chuyến đi.. 1= Credit card 2= Cash 3= No charge 4= Dispute5= Unknown 6= Voided trip
Fare_amount	Giá vé được tính theo thời gian và quãng đường trong đồng hồ.
Extra	Các khoản phụ phí và phụ phí khác. Hiện tại, khoản phí này chỉ bao gồm 0,5 đô la và 1 đô la cho giờ cao điểm và phí qua đêm.
MTA_tax	Thuế MTA \$ 0,50 được tự động kích hoạt dựa trên tỷ giá đo được sử dụng.
Improvement_surcharge	0,30 phụ phí cải thiện các chuyến đi được đánh giá tại điểm thả cò. Phụ phí cải tiến bắt đầu được đánh vào năm 2015
Tip_amount	Số tiền doanh thu - Trường này tự động được điền cho các tips về thẻ tín dụng.Tiền boa không được bao gồm.
Tolls_mout	Tổng số tiền của tất cả các khoản phí phải trả trong chuyến đi.
Total_amount	Tổng số tiền phải trả lại cho hành khách. Không bao gồm tiền boa
Congestion_surcharge	Số tiền phải trả khi qua các trạm thu phí.

2.2 Vấn đề dataset, vấn đề bài toán và phương pháp giải quyết

- **Vấn đề dataset:** Dữ liệu lượng sample lớn, vì vậy những dữ liệu sai lệch, nhiễu, khuyết,... có thể có rất nhiều. Điều này dẫn đến việc thăm dò dữ liệu lúc đầu sẽ tốn rất nhiều thời gian

- **Vấn đề bài toán:** Bài toán của chúng ta ở đây là dự đoán nhu cầu của khách hàng đối với những chiếc taxi màu vàng theo khu vực trong hành phố New York, cho nên dữ liệu của chúng ta để đáp ứng cho bài toán là những dữ liệu chứa trong features liên quan đến vị trí, cũng như các features có liên quan đến chúng.

=> **Hướng giải quyết chung 2 vấn đề trên:** Chọn lọc ra những features đã biết được ý nghĩa của chúng qua phần mô tả dataset ở trên để chúng ta tiến hành đi phân tích thăm dò, làm sạch, tổng hợp và cũng như các bước phân tích, đánh giá tiếp theo. Việc làm này có thể giảm được lượng lớn sample mà vẫn có thể giải quyết được bài toán này.

=> Hướng giải quyết cho việc đọc dữ liệu lớn trong giai đoạn đầu của thăm dò và tiền xử lý dữ liệu: dùng thư viện Dask⁽¹⁾ có sẵn trong Python.


2.3 Thăm dò và tiền xử lý những dữ liệu liên quan đến bài toán đặt ra

- Trong quá trình tìm hiểu, thăm dò và xử lý các dữ liệu liên quan đến bài toán, tôi đã tóm tắt và phân chia các giai đoạn các bước trong mục này thành 3 phần:

2.3.1 Dữ liệu thời gian

- Chuẩn hóa dữ liệu thời gian sang đúng dữ liệu dạng đầy đủ các thông số:

	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime
0	1.0	01/01/2020 12:28:15 AM	01/01/2020 12:33:03 AM
1	1.0	01/01/2020 12:35:39 AM	01/01/2020 12:43:04 AM



	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime
413581	NaN	2020-06-30 23:05:00	2020-06-30 23:32:00
413582	NaN	2020-06-30 23:21:47	2020-06-30 23:25:24

- Tôi đã truy vấn những dữ liệu thời gian được records nằm ngoài năm 2020.

```
[254] 1 # Giá trị ngoại lệ trong 2 cột thời gian
      2 outlier_dates = data.query("(tpep_pickup_datetime.dt.year != 2020) | (tpep_dropoff_datetime.dt.year != 2020)").compute()
      3 len(outlier_dates)
```

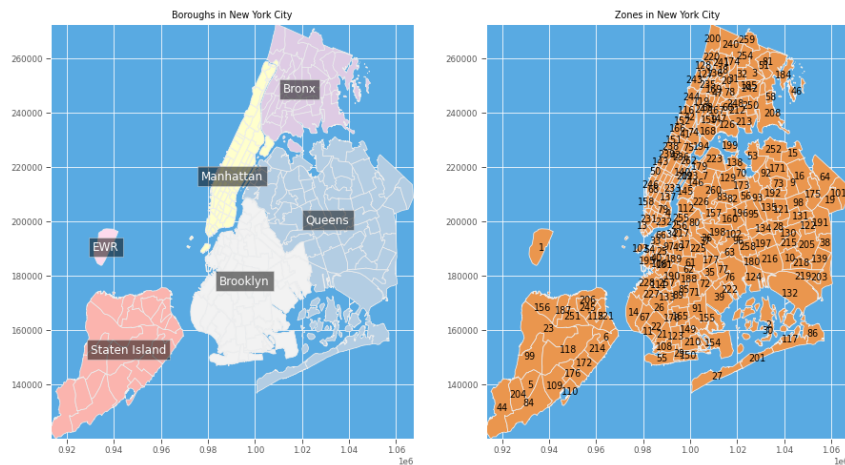
221

Hình 2. Outlier của dữ liệu thời gian

- Cụ thể là dữ liệu cao nhất có sẵn là 2021-01-02 đã được ghi sẵn, mặc dù thời gian này vẫn chưa tới. Ngược lại những dữ liệu rất lâu rồi nhưng vẫn còn loại bỏ trong bộ dữ liệu, cụ thể điểm thấp nhất là 2003-01-01. Tôi đã tiến hành loại bỏ 211 điểm ngoại lệ đó.

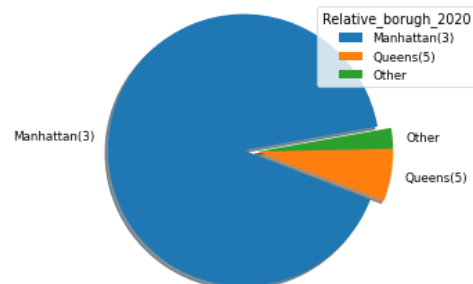
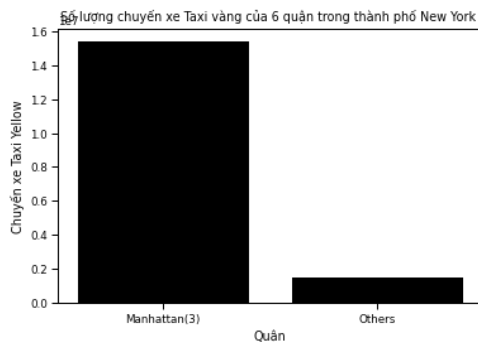
2.3.2 Dữ liệu vị trí

- Để trực quan hơn các khu vực được quy định để thực hiện những chuyến xe taxi màu vàng tôi đã cho vẽ bản đồ các khu vực trên trong toàn bộ 6 quận của thành phố.



Hình 3. Mã khu vực các vị trí đoán và trả khách trong thành phố New York

- Tiếp theo, tôi tiếp tục cho thăm dò số lượng chuyến xe (sample) của 6 quận bằng cascg gắn tên các quận và tính tổng số chuyến xe của mỗi quận, kết quả được tôi thống kê qua các biểu đồ sau:



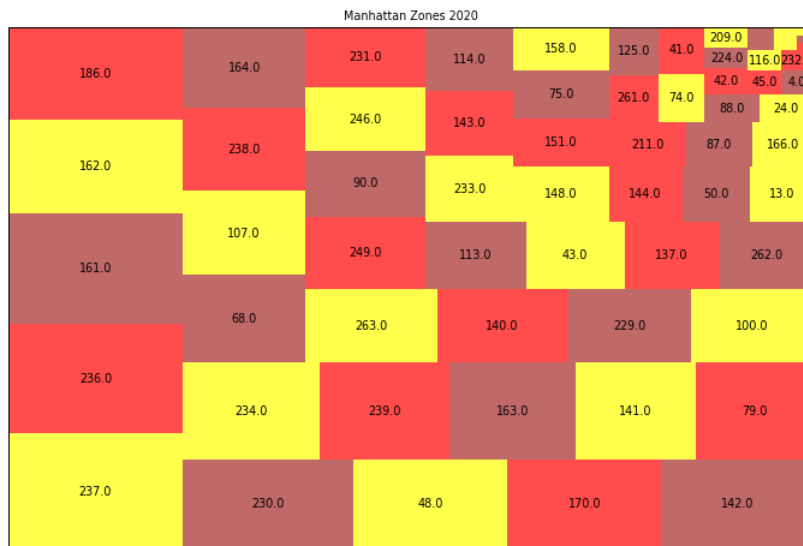
Hình 4: Biểu đồ cột thể hiện số lượng chuyến xe taxi vàng.

Hình 5: Biểu đồ tròn thể hiện tỉ lệ % các chuyến xe taxi vàng chiếm trong 6 quận.

=> Qua 2 biểu đồ trên, tôi quyết định chọn toàn bộ dữ liệu cả quận Manhattan để tiếp tục thăm dò và phân tích sâu hơn, lí do đơn giản vì đa số các xe taxi vàng nhận khách ở quận này, cụ thể số lượng chuyến xe là 15386529 chiếm tỉ lệ (91,3%) so với những quận còn lại. Ngoài ra, tôi cũng có thống kê số chuyến xe được trả khách của tất cả các quận và kết quả thu được số lượng được trả ở Manhattan cũng chiếm đa số (14310435-93%).

2.3.3 Dữ liệu quận Manhattan

- Thăm dò dữ liệu tất cả các khu vực trong quận (67 khu vực) ta được số lượng chuyến xe được thực hiện trong mỗi khu vực, cũng như tỉ lệ % của chúng chiếm trong đó. Cụ thể khu vực có số lượng cao nhất là khu vực 237 với 740022 chuyến xe được ghi lại (4,81%), và khu vực thấp nhất là khu vực 105 với 7 chuyến xe được ghi lại ($0,45.10^{-4}$).



Hình 6: Visuzalie treemap của tất cả khu vực trong quận Manhattan

2.4 Tổng hợp dữ liệu và chọn ra features cho bộ dữ cuối cùng để phục vụ bài toán

- **Phương pháp thực hiện ở phần này:** Chọc lọc thủ công dựa trên vấn đề bài toán, các chính sách được cho phép một xe taxi vàng hoàn thành một vé chuyển đi cho khách hàng. Dĩ nhiên, vì chọn lọc tự nhiên cho nên các features không phải hoàn toàn tối ưu, nhưng các features chứa dữ liệu rất dễ đi thăm dò và làm sạch, dựa trên yêu cầu bài toán đặt ra và những chính sách sau: số lượng hành khách, vé cho một chuyến xe⁽³⁾,...

- Các features được tôi chọn ra, đồng thời cũng được thăm dò và làm sạch từng features:

- Mô tả chi tiết các giá trị của dữ liệu trong các features được chọn:

Describe	Count	Mean	Std	Min	75%	Max
Features selected						
Passenger_count	1,525e	1,494e	1,138e	0,0000e	2,0000e	9,0000e
Trip_distances	1,538e	2,378e	6,96e	-2,947e	3,19000e	2,203ee
Fare_amount	1,538e	1,102e	2,033e	-1,230e	1,4000e	6,711e

- Giải thích và chi tiết cách làm sạch dữ liệu:

+ **2 features thời gian và 2 features vị trí** chắc chắn được chọn, vì dữ liệu của chúng hoàn toàn liên quan trực tiếp đến bài toán và đã được thăm dò và chuẩn hóa ở trên

%). Bên đây tôi đã vẽ tree bank⁽²⁾ để thể hiện những số liệu trên: chúng ta có thể thấy khu vực 237 chiếm phần diện tích lớn nhất. Vì vậy, tôi quyết định dữ liệu ở khu vực này để tiến hành train model cũng đánh giá độ đo và xem hiệu suất model.

+ **Passenger_count**: Số lượng hành khách trên xe trong một chuyến đi – Vì để tính được số tiền vé chuyến xe nên features không thể thiếu. Theo chính sách quy định, số lượng hành khách tối đa trên xe taxi vàng là 6 và tối thiểu là 1.

➔ **Làm sạch**: Loại bỏ các chuyến xe có số khách hàng bằng 0 là 304802 chuyến xe, loại bỏ các chuyến xe lớn 6 là 81 chuyến, như ở mô tả trên có chuyến tận 9 khách.

+ **Fare_amount**: Giá vé chuyến xe được tính sau mỗi chuyến xe – Features này là features dự đoán để giải quyết bài toán nên bắt buộc phải chọn. Theo như chính sách giá vé chuyến xe trên ta sẽ chọn được những features phụ thuộc dưới đây.

➔ **Làm sạch**: Chỉ lấy những chuyến xe thanh toán lớn hơn 2,5\$ vì đây là số tiền vé tối thiểu mà khách hàng sẽ trả.

+ **Payment_type**: Hình thức thanh toán vé chuyến xe của khách hàng – Giá vé được tính như thế nào cũng hoàn thành phụ thuộc vào hình thức thanh toán.

➔ **Làm sạch**: Chỉ lấy những chuyến xe có 2 hình thức thanh toán chủ yếu là Credit card (khoảng 1m14 chuyến) và Cash (khoảng 372k chuyến), còn lại đều loại bỏ.

+ **Trip_distances**: Rõ hơn trên mô tả là quãng đường xe đi được trong một chuyến xe – Features này cũng phải chọn vì nó phụ thuộc hoàn toàn features fare_amount.

➔ **Làm sạch**: Loại bỏ những chuyến xe có khoảng cách nhỏ hơn 0, những khoảng cách này chủ yếu do những chuyến bị hủy chưa xóa được đồ hồ ghi lại nên kết quả được đồng hồ cho là giá trị âm. Ngoài ra, chuyển giá trị khoảng cách về đúng giá trị đơn vị mét của Việt Nam quy định.

+ **RatecodeID**: Mã giá vé cuối cùng sau khi hoàn thành chuyến xe sẽ là loại nào – Features cũng được chọn chỉ để giúp sạch các features trên.

➔ **Làm sạch**: Chọn loại mã được thống kê chủ yếu là Standard rate có khoảng 1m5 vé chuyến là thuộc loại này, những mã còn lại đều loại bỏ hết.

➔ *Chiều dài dataset lúc này là : từ 16m8 sample còn 13.806.846 sample .*

2.5 Chuẩn hóa các features (Features Engineering) phù hợp với format input bài toán, EDA lần cuối.

- **Input bài toán**: Là điểm dữ liệu có giá trị biến thiên liên tục tuyến tính theo thời gian, bản chất của đầu vào này là giống như đầu vào của các mô hình hồi quy trong máy học chỉ có sự khác biệt là giá trị nó biến thiên theo thời gian.

- **Chuẩn hóa các features:** + Loại bỏ một cột vị trí trả khách là DOLocationID và đưa cột thời gian nhận khách để phù hợp với đầu vào bài toán.

+ Dùng kĩ thuật resample trong thư viện pandas để chuẩn hóa cột thời gian nhận khách về những khoảng biến thiên với độ lệch là 1 giờ, 10 phút,.. để phù hợp với input của bài toán dự đoán này. Bước này rất quan trọng đánh dấu bước ngoặt của bài toán.

+ Đồng thời dùng các kĩ thuật gom nhóm (Groupby) và sum() tính tổng các features sau: passenger_count, trip distance, fare_amount sau mỗi những steps của resample("10min").

+ Cũng thực hiện như trên vào những features phụ thuộc khác như là: Duration (Thời lượng chuyến đi (giờ)), và Speed (vận tốc(km/h)) để có thêm biến phụ thuộc để thực hiện training model đạt độ tin cậy cao hơn.

- **EDA lần cuối:** Thăm dò dữ liệu hiện tại sau khi bước chuẩn hóa trên có kích thước là 1.858.404 sample và 6 features, có 896.114 điểm dữ liệu khuyết của features Speed. Làm sạch lần nữa bằng cách điền những giá đó.

- **Bộ dữ liệu cuối cùng:** tôi chọn bộ của khu vực có số lượng chuyến xe lớn nhất là 237.

- **Kết quả:** dữ liệu của khu vực 237 trong quận Manhattan

	PULocationID	passenger_count	trip_distance_VN	Duration	fare_amount	Speed
tpep_pickup_datetime						
2020-01-01 00:00:00	237.0	44.0	108.904309	5.208611	294.0	20.908512
2020-01-01 00:10:00	237.0	118.0	199.639123	11.191389	621.0	17.838637
2020-01-01 00:20:00	237.0	180.0	263.546174	38.904722	862.5	6.774144

Hình 7: Kết quả bộ dữ liệu cuối cùng

→ **Chiều dài dữ liệu được chọn để train model hiện tại:** từ 16m8 sample còn 37877 sample.

2.6 Các model để tiến hành training thực hiện bài toán

- Chia tập **dữ liệu của khu vực 237** trên thành 2 tập train/test tỉ lệ 2/1: tập train là lấy dữ liệu từ tháng 1 đến tháng 2 của bộ(8640), và tập test là dữ liệu tháng 3 của bộ(4464). Lí do có 6 tháng chỉ lấy 3 tháng tôi đã trình bày rõ phần vấn đề dataset.

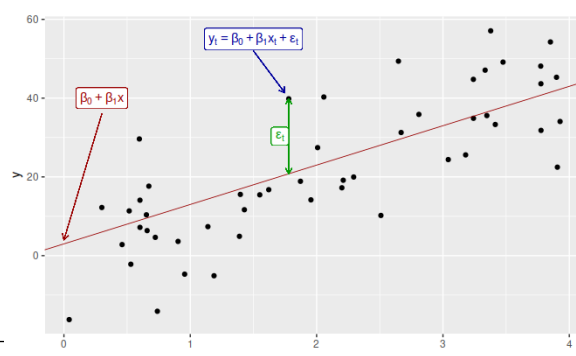
-Tiến hành thực hiện trên 3 mô hình: 2 mô hình máy học có giám sát và một mô hình học sâu:

2.6.1 Multiple regression model

- **Loại hình model:** là mô hình hồi quy tuyến tính thuộc phương pháp máy học có giám sát.

-**Ý tưởng dùng model trong bài toán:** dùng features

Hình 8: Hình multiple regression model (4)



Fare_amount để dự đoán giá xe từ các features còn lại.

-Mục đích dùng model: để pretraining (tiền thực nghiệm)

để đánh giá hiệu suất của model hồi quy này

với việc dùng input đầu vào có sự biến thiên theo thời gian

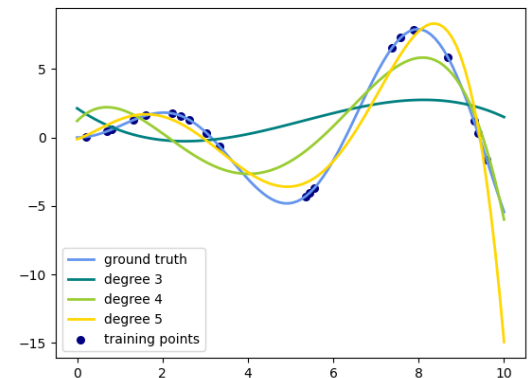
2.6.2 Poly regression model

-Loại hình model: là mô hình hồi quy tuyến tính thuộc phương pháp máy học có giám sát.

-Ý tưởng dùng model trong bài toán: dùng features

Fare_amount để dự đoán giá xe từ các features còn

lại với bậc của mô hình là 3.



-Mục đích dùng model: để pretraining (tiền thực nghiệm)

Hình 9: Hình poly regression model⁽⁵⁾

để đánh giá hiệu suất của model hồi quy này với việc dùng input đầu vào có sự biến thiên theo thời gian.

2.6.3 LSTM model

-Loại hình model: là mô hình mạng nơ ron

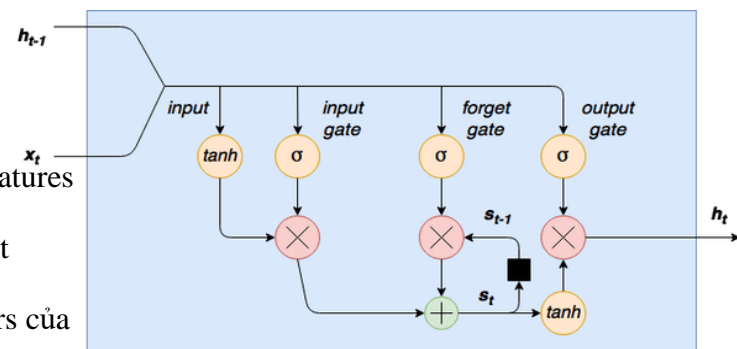
Network thuộc phương học sâu.

-Ý tưởng dùng model trong bài toán: dùng features

Fare_amount để dự đoán giá trị giá xe trong một

khoảng thời gian biến thiên là 10 phút, với layers của

model là một lớp.



Hình 10: Hình LSTM model⁽⁶⁾

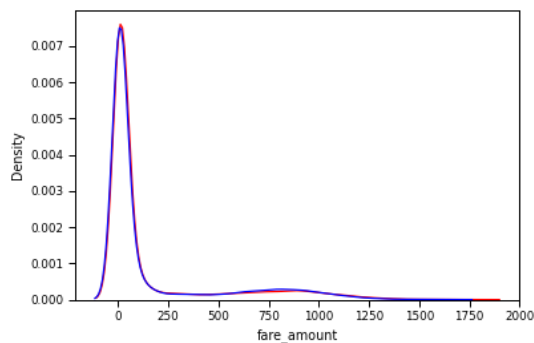
-Mục đích dùng model: làm được điều mà 2 mô hình ở trên không làm được, dự đoán được giá trị của giá xe rất chính xác trong tương lai, với những khoảng thời gian tiếp theo được dữ đoán là 10 phút. Điều đặc biệt của nó nữa là LSTM một loại mạng nơ ron thường xuyên có khả năng ghi nhớ thông tin quá khứ và trong khi dự đoán các giá trị tương lai, nó cần thông tin trong quá khứ. Hiệu suất và độ đo nó mang lại sẽ rất phù hợp với bài toán của chúng ta.

2.7 Hiệu suất model và độ đo đánh giá model

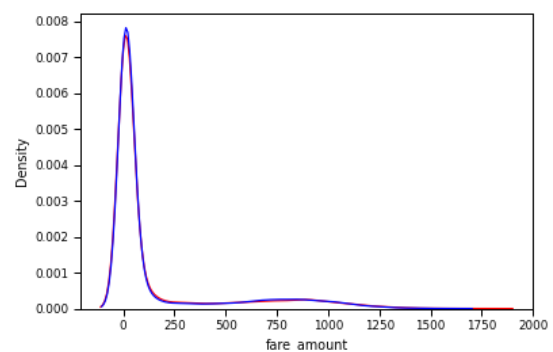
- 2 model machine: Visulze kết quả giá vé chuyển xe thực tế so với dự đoán.

- Model deep: Visuzle kết quả sai số (RMSE) thực tế so với dự đoán.

2.7.1 Model machine learning



Hình11 : Visualize hiệu suất mô hình MR



Hình12 : Visualize hiệu suất mô hình PR

- Kết quả độ đo đánh giá:

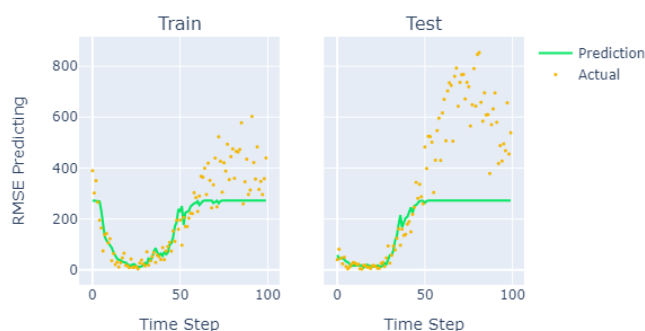
ĐỘ ĐO ĐÁNH GIÁ	MUTIPLE REGRESION MODEL	POLY REGRESION MODEL
RMSE	31,43	41,75
R2 Scores	0,989	0,982

- **Nhận xét kết quả:** Kết quả của 2 độ đo đạt kết quả tốt, vì vậy dữ liệu của features của chúng ta tối ưu, đây cũng một nhận xét đáng ghi nhận.

+ Kết quả hiệu suất của 2 model chưa thực sự đạt yêu cầu bài toán, khi những sai số của giá vé thực tế và dự đoán đường như không chênh lệch với nhau. Mô hình chưa phù hợp với bài toán.

2.7.1 Model deep learning

Model performance of time step trained



Hình13 : Visualize hiệu suất mô hình LSTM không scaler

Model performance of time step trained



Hình14 : Visualize hiệu suất mô hình LSTM có scaler

- Kết quả độ đo đánh giá:

ĐỘ ĐO ĐÁNH GIÁ	MODEL VANILA NOT SCALER		MODEL VANILLA SCALER	
Train/Test	Train	Test	Train	Test
RMSE	330,3	138,6	71,7	51,7

- **Nhận xét kết quả:** +Kết quả của độ đo đánh giá tương đối tốt mặc dù vẫn train lần 1 vẫn chưa có dùng đến chuẩn hóa dữ liệu theo Minmaxscaler, tương kết quả test cũng tương đối tốt
+Kết quả của hiệu suất model đã đạt được yêu cầu của bài toán, đường biểu diễn giá trị dự đoán của độ đạt biến thiên rất phù hợp để dự đoán với những điểm biểu diễn giá trị thực tế.

2.8 Deploy model

- Tôi hiện tại vẫn chưa có nhiều ý tưởng việc này, nhưng trong tương lai có đủ thời gian, tôi nghĩ có thể làm tốt được phần này để cho giúp model của tôi sau khi train có thể phát triển hơn ở nhiều bài toán tương tự, cũng như ở nhiều bộ dữ liệu khác.
- Tôi thật xin lỗi vì sự chưa hoàn thiện này trong đồ án.

3.KẾT LUẬN

-Qua đồ án về phân tích dữ liệu, tôi nắm rõ được tốt hơn quy trình phân tích dữ liệu. Không những thế, qua việc phân tích bộ dữ liệu rất lớn này giúp tôi có có nhiều kinh nghiệm hơn khi làm việc với bộ dữ liệu lớn.

-Tôi giải quyết và thực hiện thành công của bài toán mình đã đặt ra là đã dự đoán được nhu cầu đi Taxi màu vàng của khách hàng của từng khu vực trong thành phố đạt hiệu suất rất cao.

- Ý tưởng ứng dụng bài toán này là: Chúng ta có thể cho bộ dữ liệu đầu vào của 2 hay nhiều hơn 2 khu vực, cho tất cả chúng training model, sau đó ta thu được kết quả hiệu suất dự đoán biến thiên của giá vé của mỗi khu vực với nhau trong cùng một khoảng thời gian cố định, cứ thế sau 10 phút/1 giờ người tài xế có thể đoán biết khu vực nào sẽ đông khách hay ít khách trong thời gian tiếp theo.

- Vẫn còn chưa đủ thời gian nghiên cứu làm bước deploy, tương lai tôi sẽ làm và mở rộng ứng dụng của bài toán.

TÀI LIỆU THAM KHẢO

- [1] <https://docs.dask.org/en/latest/dataframe.html>
- [2] <https://fcpython.com/visualisation/python-treemaps-squarify-matplotlib>
- [3] <https://www1.nyc.gov/site/tlc/passengers/taxi-fare.page>
- [4] <https://otexts.com/fpp2/regression-intro.html>
- [5] https://scikitlearn.org/stable/auto_examples/linear_model/plot_polynomial_interpolation.html

LSTM Model

- [6] <https://adventuresinmachinelearning.com/keras-lstm-tutorial/>
- [7] https://nttuan8.com/bai-14-long-short-term-memory-lstm/#Gioi_thieu_ve_LSTM
- [8] <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>
- [9] Jason Brownlee LSTM

Dr. N.D Lewis Deep Learning for Time-series

<https://livebook.manning.com/#!/book/data-science-at-scale-with-python-and-dask>

CÁC NGUỒN TÀI LIỆU KHÁC TRONG KHI CODE

- [10] <https://machinelearningmastery.com/make-predictions-time-series-forecasting-python/>
- [11] <https://machinelearningmastery.com/time-series-data-stationary-python/>
- [12] <https://machinelearningmastery.com/time-series-forecasting-methods-in-python-cheat-sheet/>
- [13] <https://tomaugspurger.github.io/modern-7-timeseries.html>
- [14] <https://github.com/aryafarkhondeh/NYC-Taxi-Data-Analysis>

PHỤ LỤC PHÂN CÔNG NHIỆM VỤ

STT	Thành viên	Nhiệm vụ
1	Võ Đình Tứ	<div><div>-Nghiên cứu dataset và đưa đề tài để thực hiện đồ án</div><div>-Phân tích bộ dataset sơ bộ, thăm dò dữ liệu ở những dữ liệu sơ bộ phân tích và báo cáo tiến độ.</div><div>-Hoàn thành source code của toàn bộ bài toán và báo cáo tiến độ.</div><div>-Hoàn thành được bài báo cáo cho đồ án.</div><div>-Hoàn thành được silde báo cáo.</div><div>-Thuyết trình kết thúc đồ án.</div></div>