Đồ án CS116

Nguyễn Anh Tuấn-Lớp CS116.O11.KHCL-MSSV: 20522114[1,2]
[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
Email:20522114@gm.uit.edu.vn

***Tóm tắt nội dung*—Trong đồ án này, em đã thực hiện các bước như là đánh giá data xem data đã phù hợp để được vào model đưa ra dự đoán hay chưa. Sau đó xem xét chọn model phù hợp để dự đoán**

***Index Terms*—CS116-UIT, phân tích dữ liệu, mô hình hồi quy.**

I. Giới thiệu

Bài toán hồi quy này với bộ dữ liệu bao gồm mười một tín hiệu thu thập từ một phương tiện, trong đó chín tín hiệu đầu tiên được sử dụng như các đặc trưng đầu vào cho việc dự đoán, còn hai tín hiệu cuối cùng biểu thị đầu ra mong muốn. Các tín hiệu này bao gồm thông số quan trọng như tốc độ trung bình của động cơ, tốc độ của phương tiện, mô-men xoắn động cơ tính toán lại, trạng thái côn, và nhiều giá trị liên quan đến mô-men xoắn. Đáng chú ý, RoadSlope_100ms cung cấp thông tin về độ dốc thực tế, và Vehicle_Mass cho biết trọng lượng của phương tiện. Lưu ý rằng dữ liệu không có thông tin thời gian rõ ràng, và thứ tự ghi âm đã được phân tán, do đó mọi thuật toán hồi quy được áp dụng phải hoạt động độc lập trên từng khung dữ liệu. Mục tiêu là phát triển một mô hình hồi quy tuyến tính có khả năng dự đoán chính xác đầu ra mong muốn dựa trên các tín hiệu đầu vào đã cho, đồng thời xem xét đến sự phức tạp được giới thiệu bởi tập hợp đa dạng các đặc trưng và thiếu thông tin về thời gian trong dữ liệu.

II. Phương pháp giải quyết bài toán

Trong bài toán dự đoán RoadSlope_100ms và Vehicle_Mass thì em đã thực hiện thông qua các bước sau:

A. Đánh giá dữ liệu

Trước khi bắt đầu xây dựng mô hình, việc đánh giá dữ liệu là quan trọng để hiểu rõ về tính chất và phân phối của các biến, đặc trưng cũng như biến mục tiêu (RoadSlope_100ms và Vehicle_Mass). Điều này có thể bao gồm việc kiểm tra giá trị ngoại lệ (outlier), phân phối, và tương quan giữa các biến.

B. Tiền xử lý dữ liệu

Trong bước này, em sẽ tiến hành tiền xử lý dữ liệu để làm sạch và chuẩn hóa dữ liệu. Các bước cụ thể có thể bao gồm loại bỏ giá trị ngoại lệ, xử lý giá trị thiếu, và chuẩn hóa các biến để đảm bảo mô hình được huấn luyện trên dữ liệu đồng nhất và đồng đều.

C. Lựa chọn đặc trưng

Dựa trên đánh giá dữ liệu và tính quan trọng của các đặc trưng trong bài toán mà em sẽ lựa chọn các đặc trưng quan trọng để sử dụng trong mô hình. Các phương pháp như phân tích độ tương quan và kiểm tra thống kê sẽ hỗ trợ quyết định này.

D. Chọn mô hình và huấn luyện mô hình

Em lựa chọn mô hình để huấn luyến trên tập dữ liệu đã được tiền xử lý để huấn luyện mô hình hồi quy tuyến tính. Quá trình này sẽ điều chỉnh các tham số của mô hình để tối ưu hóa dự đoán cho RoadSlope_100ms và Vehicle_Mass.

E. Đánh Giá Mô Hình

Cuối cùng, mô hình sẽ được đánh giá trên tập kiểm tra để đảm bảo rằng nó có khả năng dự đoán chính xác và có khả năng tổng quát hóa tốt trên dữ liệu mới. Sử dụng các metric như Mean Squared Error (MSE) sẽ giúp đo lường độ chính xác của dự đoán.

III. Thực hiện phương pháp

A. Đánh giá dữ liệu

1) Giới thiệu về dữ liệu: Mười một tín hiệu thu thập từ phương tiện là:

-**Epm_nEng_100ms**: Tốc độ trung bình của động cơ của một đoạn xi-lanh (vòng/phút)

-**VehV_v_100ms**: Tốc độ của xe (km/h)

-**ActMod_trqInr_100ms**: Mô-men xoắn hiện tại của động cơ được tính lại từ bên trong (Nm)

-**RngMod_trqCrSmin_100ms**: Mô-men xoắn tối thiểu của động cơ ở mức trục khuỷu (Nm)

-**CoVeh_trqAcs_100ms**: Yêu cầu mô-men xoắn của các phụ kiện (Nm)

-**Clth_st_100ms**: Trạng thái được chuyển động của côn sau khi làm mờ (-)

-**CoEng_st_100ms**: Trạng thái hoạt động của động cơ (enum, 0 COENG_STANDBY, 1 COENG_READY, 2 COENG_CRANKING, 3 COENG_RUNNING, 4 COENG_STOPPING, 5 COENG_FINISH)

-**Com_rTSC1VRVCURtdrTq_100ms**: Mô-men xoắn mong muốn hoặc giới hạn mô-men xoắn (

-**Com_rTSC1VRRDTrqReq_100ms**: Mô-men xoắn được yêu cầu bởi hạn chế (retarder) (

-**RoadSlope_100ms**: Độ dốc thực tế từ chân trời ADASIS (

-**Vehicle_Mass**: Trọng lượng của xe, có thể là 38 t hoặc 49 t

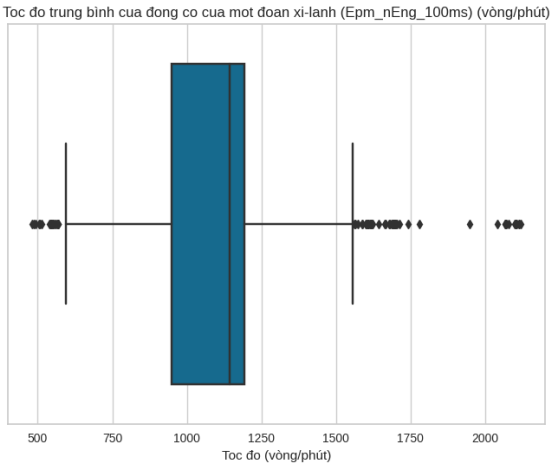
2) Xem xét từng loại dữ liệu: Khi đến với bài toán này thì ta cần xem xét giá trị max, min, mean,... nên ta có bảng sau:

toprule	count	mean	std	min	25%	50%	75%	max
midrule Epm_nEng_100ms	8496.0	1104.67	157.27	481.50	948.38	1145.00	1192.50	2119.50
VehV_v_100ms	8496.0	64.35	10.69	38.31	57.09	67.38	71.58	88.93
ActMod_trqInr_100ms	8496.0	1419.60	989.41	0.00	363.92	1627.50	2348.74	2688.00
RngMod_trqCrSmin_100ms	8496.0	-158.92	26.45	-308.00	-168.00	-168.00	-140.00	-84.00
CoVeh_trqAcs_100ms	8496.0	9.999747	0	9.999747	9.999747	9.999747	9.999747	9.999747
Clth_st_100ms	8496.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CoEng_st_100ms	8496.0	3.0	0.0	3.0	3.0	3.0	3.0	3.0
Com_rTSC1VRVCURtdrTq_100ms	8496.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Com_rTSC1VRRDTrqReq_100ms	8496.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RoadSlope_100ms	8496.0	0.88	2.20	-4.80	-0.40	1.00	2.10	5.00
Vehicle_Mass	8496.0	41.60	5.16	38.00	38.00	38.00	49.00	49.00

bottomrule

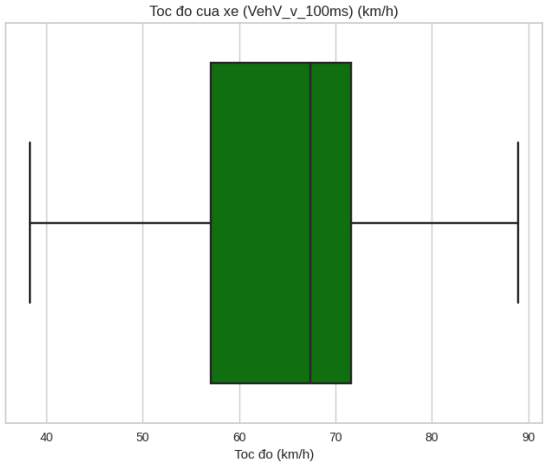
Bảng I: Bảng thống kê mô tả các loại tín hiệu của xe

-Dễ thấy được là dữ liệu của cột CoEng_st_100ms là dạng enum mà trong đó thì chỉ có một giá trị duy nhất không thay đổi là số 3: COENG_RUNNING.
-Khi nhìn vào bảng trên ta thấy được các cột CoVeh_trqAcs_100ms, Clth_st_100ms, Clth_st_100ms, CoEng_st_100ms, Com_rTSC1VRVCURtdrTq_100ms, Com_rTSC1VRRDTrqReq đều là những giá trị gần như không thay đổi trong tổng số 8496 mẫu dữ liệu.
-Kế tiếp ta sẽ xét tới từng loại đặc trưng. Cái đầu tiên là Epm_nEng_100ms:



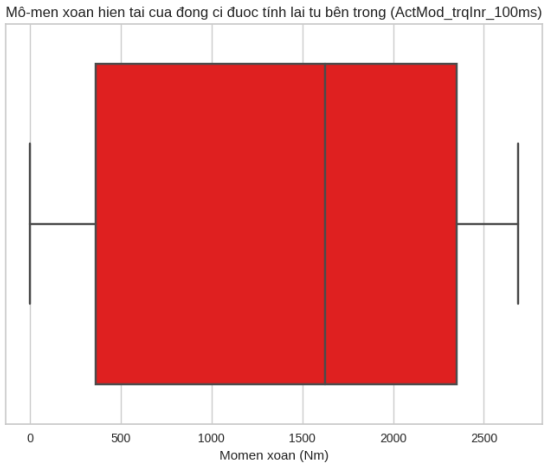
Hình 1: Biểu đồ Box plot Plot của Epm_nEng_100ms

-Thông qua biểu đồ trên em có nhận xét là sự phân bố dữ liệu của Epm_nEng_100ms khá là sát nhau không xê lệch nhau qua lớn nhưng lại tồn tại giá trị ngoại lai. Gồm có những giá trị ngoại lai. Theo tính được thì gồm có **82** giá trị ngoại lai. Các giá trị ngoại lai gồm các giá trị bé hơn 500 hơn lớn hơn 1500
-Kế đến em sẽ xem xét trong đặc trưng VehV_v_100ms:



Hình 2: Biểu đồ Box plot Plot của VehV_v_100ms

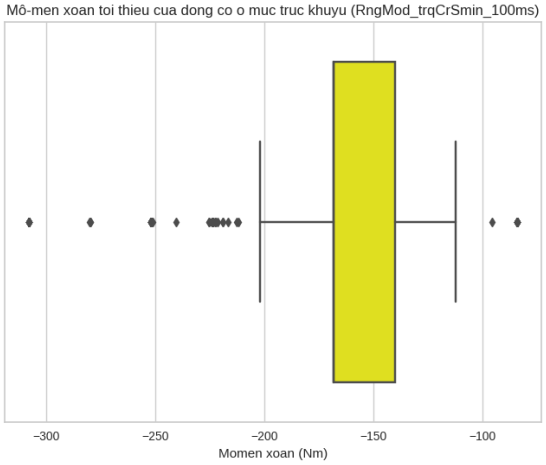
-Đặc trưng VehV có sự phân bố đồng đều trong khoảng 60 đến 70. Mặc dù phân bố trong khoảng thấp vậy mà nó không tồn tại ngoại lai. Nên dự đoán đây sẽ là đặc trưng có sức ảnh hưởng đến bài toán.



Hình 3: Biểu đồ Box plot Plot của ActMod_trqInr_100ms

-Dữ liệu của cột này có sự phân bố rất rộng từ 500 đến 2500

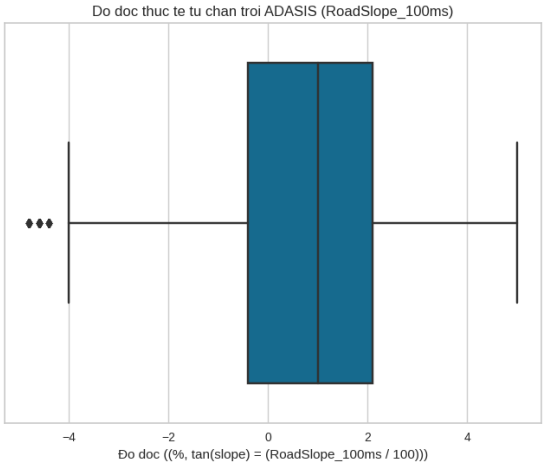
chính vì thế không tồn tại ngoại lai.



Hình 4: Biểu đồ Box plot Plot của RngMod_trqCrSmin_100ms

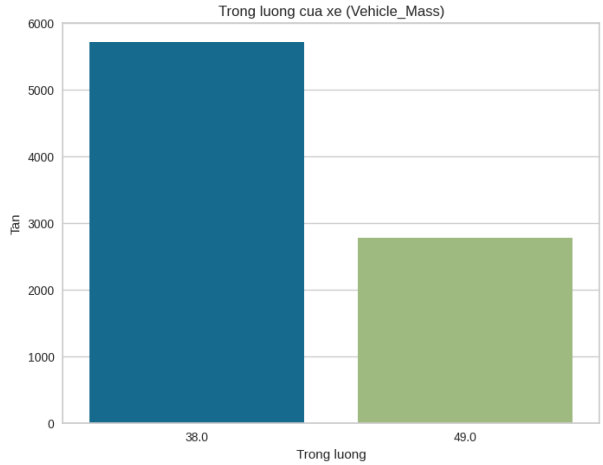
-Các giá trị của RngMod_trqCrSmin_100ms phân bố rất ít có sự biến động quá nhiều chỉ có các ngoại lai là lệch rất lớn. Các giá trị ngoại lai là các giá trị lớn hơn -100 và bé hơn -200. Có **180** giá trị ngoại lai ở trong RngMod_trqCrSmin_100ms.

-Em sẽ xem xét tiếp 2 cột dữ liệu biến mục tiêu được yêu cầu :



Hình 5: Biểu đồ Box plot Plot của RoadSlope_100ms

-Ở đây em sẽ kiểm tra xem ngoại lai của RoadSlope_100ms có phải do bị ảnh hưởng bởi các giá trị ngoại lai từ các đặc trưng khác không. Thì kết quả đưa ra là dữ liệu ngoại lai của RoadSlope_100ms không bị ảnh hưởng bởi ngoại lai của các dữ liệu của ngoại lai từ 2 đặc trưng đề cập trên mà là do các nguyên nhân khác. Nhưng khi kiểm tra ngoại lai của 2 đặc trưng đó thì sẽ thấy được 2 đặc trưng có ngoại lai đó lại cùng là ngoại lai khi đặc trưng thứ nhất bị ngoại lai thì đặc trưng thứ 2 cũng bị. Nên có một phương pháp để giải quyết là có thể xóa đi ngoại lai vì cả 3 đều là những con số rất nhỏ.



Hình 6: Biểu đồ phân bố của Vehice_mass

-Em thấy được là dữ liệu của đặc trưng này mất cân bằng cột trọng lượng 49 tấn chỉ bằng một nửa so với lượng dữ liệu của lượng dữ liệu mà cột 38 tấn mạng lại. Cho nên với bài toán hồi quy mất cân bằng dữ liệu này thì em chia làm 2 cách xử lý là có thể giải quyết bằng 2 cách là Oversampling hoặc là Undersmapling.

B. Tiền xử lý dữ liệu

1) Xử lý ngoại lai: Xử lý ngoại lệ trong dữ liệu số đồng nghĩa với việc đối mặt và giải quyết các giá trị ngoại lệ trong tập dữ liệu. Quá trình này có nhiều ảnh hưởng tích cực, bao gồm tăng tính ổn định và chính xác của mô hình học máy, giảm tác động của nhiễu, và cải thiện hiệu suất dự đoán.

Đến với phần dữ liệu được cho em đã nhận xét trên là những ngoại lai trên đều không có ảnh hưởng gì đến với kết quả của 2 dữ liệu RoadSlope_100ms với Vehice_mass. Tổng số lượng ngoại lai ở đây là 180 + 82= **262** giá trị ngoại lai (rất nhỏ nếu so với số lượng 8496).

2) Xử lý mất cân bằng dữ liệu: Với dữ liệu của Vehice_mass thì chỉ có hai giá trị là 38 và 49. Thì 2 cột dữ liệu này lại có cảm giác bị mất cân bằng. Giá trị 38 xuất hiện 5719 lần, giá trị 49 xuất hiện 2777 lần. Để giải quyết trường hợp này thì sẽ áp dụng 2 phương pháp là Oversmapling hoặc Undersmapling. Ở đây em áp dụng cả 2 phương pháp để xem là với số lượng mẫu là 8234 thì nó sẽ phù hợp nhất với loại nào.

3) Mã hóa dữ liệu : Có rất nhiều phương pháp mã hóa dữ liệu nhưng như đã đề cập trên thì dữ liệu có tồn tại ngoại lai nên các cách như Min-Max Scalling, Standardization sẽ không phù hợp bởi vì ngoại lai có thể ảnh hưởng đến việc mã hóa của 2 phương pháp trên sẽ bị lệch rất lớn. Còn Log Transformation không thể áp dụng vì sự phân bố dữ liệu có giá trị âm. Chính vì thế còn một phương pháp phù hợp đó chính là Robust Scaler.

Robust Scaler bắt đầu bằng việc tính toán giá trị trung bình trung vị (median), điều này giúp chọn ra một giá trị trung tâm chống lại sự ảnh hưởng của giá trị ngoại lệ. Sau đó, nó sử dụng phạm vi tứ phân vị (IQR - Interquartile Range), đo

lường sự biến động của dữ liệu bằng cách tính chênh lệch giữa tứ phân vị thứ nhất và tứ phân vị thứ ba.

Đối với mỗi điểm dữ liệu, Robust Scaler áp dụng một công thức chuẩn hóa, sử dụng giá trị trung bình trung vị và IQR để điều chỉnh giá trị. Cụ thể, công thức này chia sự chênh lệch giữa giá trị dữ liệu và median cho IQR. Quá trình này giúp chuẩn hóa dữ liệu sao cho giá trị trung bình xấp xỉ 0 và độ biến động của dữ liệu được kiểm soát bởi IQR. Kết quả là dữ liệu đã được chuẩn hóa, giảm ảnh hưởng của giá trị ngoại lệ và giữ nguyên tính chất của phân phối ban đầu. Robust Scaler thường được sử dụng khi dữ liệu chứa nhiều giá trị ngoại lệ và muốn giữ nguyên tính chất của phân phối dữ liệu. Đó chính là cách hoạt động của Robust Scaler.

C. Lựa chọn đặc trưng

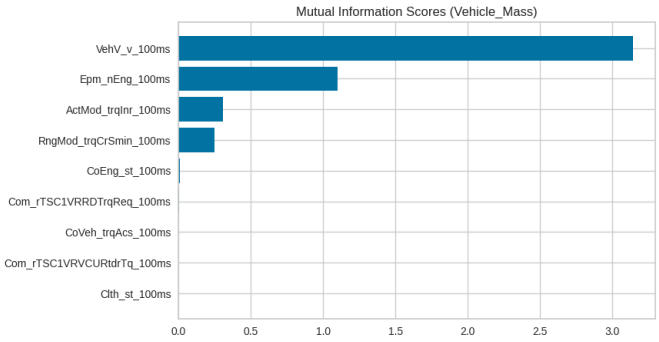
Để lựa chọn các đặc trưng phù hợp, ở đây em sử dụng phương pháp Fillter để xem xét và đánh giá xem đặc trưng nào sẽ là tốt nhất cho mô hình. Phương pháp Filter áp dụng một chỉ số đã chọn để tìm các thuộc tính không liên quan và lọc ra dữ liệu dư thừa. Phương pháp Filter gồm có chọn lọc dựa trên các yếu tố sau là: Tương quan Person, Ngưỡng phương sai, Thiếu tỉ lệ giá trị, Thông tin tương hỗ (MI_scores) Ở đây em chỉ sử dụng Ngưỡng phương sai kèm với Thông tin tương hổ bởi vì trong dữ liệu được cho thì không hề tồn tại thiếu tỉ lệ giá trị trong cột dữ liệu nào cả. Các dữ liệu trong cột đều được xếp đầy đủ

1) Ngưỡng phương sai: Phương sai là một đo lường cho mức độ biến động của một biến số. Trong ngữ cảnh của việc lựa chọn đặc trưng, ngưỡng phương sai được sử dụng để loại bỏ các đặc trưng mà phương sai của chúng quá nhỏ, tức là chúng không đưa thêm thông tin đặc biệt nhiều cho mô hình và có thể được coi là "đặc trưng không quan trọng". Nếu giá trị của phương sai càng thấp thì cho thấy biến đó không có nhiều sự biến động ở trong dữ liệu.

Khi xét đến trong dữ liệu thì thấy Clth_st_100ms, CoEng_st_100ms, Com_rTSC1VRVCURtdrTq_100ms và Com_rTSC1VRRDTrqReq_100ms gần như có phương sai là bằng 0 nên có thể loại bỏ khỏi dữ liệu để cho mô hình bớt phức tạp.

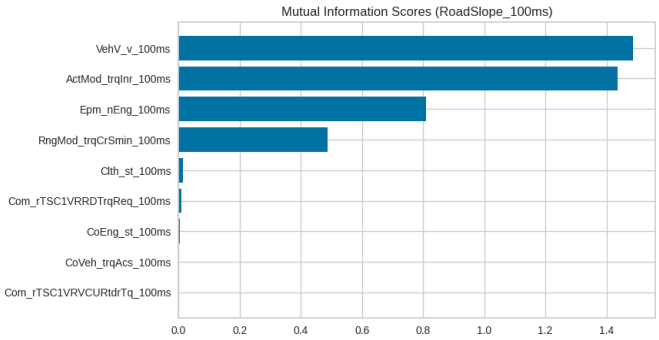
2) Thông tin tương hỗ(MI socres): Thông tin tương hỗ (Mutual Information - MI) là một độ đo thống kê trong thống kê và học máy, được sử dụng để đo lường mức độ phụ thuộc tuyến tính hoặc phi tuyến tính giữa hai biến ngẫu nhiên. MI giữ thông tin về mối quan hệ giữa các biến và có thể được sử dụng để đánh giá sự quan trọng của mỗi biến đối với một biến khác.

Đối với bài toán lựa chọn đặc trưng trong học máy, MI thường được sử dụng để đo lường độ tương hỗ giữa mỗi đặc trưng và biến mục tiêu. Nếu MI giữa một đặc trưng và biến mục tiêu cao, có thể ngụ ý rằng đặc trưng đó chứa nhiều thông tin hữu ích về biến mục tiêu và có thể là một đặc trưng quan trọng cho mô hình.



Hình 7: Thông tin tương hỗ của Vehice_mass

Thấy được mức độ phụ thuộc của 5 thuộc tính CoEng_st_100ms, Clth_st_100ms, CoEng_st_100ms, Com_rTSC1VRVCURtdrTq_100ms và Com_rTSC1VRRDTrqReq_100ms đều rất ít hoặc là gần như không có so với Vehice_mass.



Hình 8: Thông tin tương hỗ của RoadSlope_100ms

Thấy được mức độ phụ thuộc của 5 thuộc tính CoEng_st_100ms, Clth_st_100ms, CoVeh_trqAcs_100ms, Com_rTSC1VRVCURtdrTq_100ms và Com_rTSC1VRRDTrqReq_100ms đều rất ít hoặc là gần như không có so với RoadSlope_100ms

Chính vì vậy đặc trưng lựa chọn cho 2 mô hình đều là loại bỏ các đặc trưng CoEng_st_100ms, Clth_st_100ms, CoVeh_trqAcs_100ms, Com_rTSC1VRVCURtdrTq_100ms và Com_rTSC1VRRDTrqReq_100ms giữ lại các đặc trưng ActMod_trqInr_100ms, RngMod_trqCrSmin_100ms, Epm_nEng_100ms và cuối cùng VehV_v_100ms.

D. Chọn mô hình và huấn luyện mô hình

Sử dụng K-Fold để chia các tập dữ liệu sau đó đánh giá trên các Fold.

1) Mô hình cho bài toán Vehice_mass: Bài toán dự đoán kết quả trọng lượng của xe chỉ có 38 và 49 sẽ phù hợp các mô hình sau: Logistic Regression và xét thêm các mô hình sau: Decision Tree Classifier, Support Vector Machine.

2) Mô hình cho bài toán RoadSlope_100ms: Bài toán hồi quy tuyến tính đa biến sẽ phù hợp các mô hình sau: Decision Tree Classifier, Linear Regression, Support Vector Machine.

E. Đánh giá mô hình

Để có thể đánh giá được hiệu quả của các mô hình thì cần phải có các độ đo để xem xét xem là mô hình này hơn mô hình kia như thế nào. Sau đây là các độ đo được sử dụng trong hai bài toán,

1) Độ đo đánh giá trên mô hình dự đoán Vehice_mass: Trong bài toàn dự đoán Vehice_mass thì em sử dụng các độ đo là Accuracy, Recall, Precision

Accuracy dùng để đo lường tỷ lệ dự đoán đúng trên tổng số mẫu. Nó là một độ đo tổng thể về hiệu suất của mô hình. Với độ đo Accuracy (Độ Chính Xác): Càng cao, mô hình càng đúng đắn vì nó dự đoán đúng một lượng lớn các mẫu. Công thức tính:

$$\text{Accuracy} = \frac{\text{Số lượng dự đoán đúng}}{\text{Tổng số dự đoán}}$$

Recall dùng để đo lường khả năng của mô hình nhận diện tất cả các trường hợp positive. Nó quan trọng khi muốn tránh bỏ sót các trường hợp positive. Với độ đo Recall (Độ Nhớ): Càng cao, mô hình càng tốt trong việc nhận diện tất cả các trường hợp positive thực sự. Quan trọng khi không muốn bỏ sót các trường hợp positive quan trọng. Công thức tính:

$$\text{Recall} = \frac{\text{Số lượng dự đoán đúng positive}}{\text{Tổng số positive thực sự}}$$

Precision dùng để đo lường khả năng của mô hình không làm giả mạo khi dự đoán positive. Nó quan trọng khi muốn giảm số lượng dự đoán positive sai. Với độ đo Precision (Độ Chính Xác): Càng cao, mô hình càng giảm thiểu việc dự đoán sai positive. Quan trọng khi muốn giảm số lượng dự đoán positive sai. Công thức tính:

$$\text{Precision} = \frac{\text{Số lượng dự đoán đúng positive}}{\text{Tổng số positive được dự đoán}}$$

2) Độ đo đánh giá trên mô hình dự đoán RoadSlope_100ms: Trong bài toán dự đoán RoadSlope_100ms thì em sử dụng độ đo là MAE và MSE.

MAE (Mean Absolute Error) đo lường trung bình giá trị tuyệt đối của sự chênh lệch giữa giá trị thực tế (y_i) và giá trị dự đoán ($\hat{y}_i$), được tính bằng công thức:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Trong đó:
- n là số lượng điểm dữ liệu.
- y_i là giá trị thực tế của điểm dữ liệu thứ i.
- $\hat{y}_i$ là giá trị dự đoán của điểm dữ liệu thứ i.

MSE (Mean Squared Error) đo lường trung bình của bình phương của sự chênh lệch giữa giá trị thực tế (y_i) và giá trị dự đoán ($\hat{y}_i$), được tính bằng công thức:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Trong đó:
- n là số lượng điểm dữ liệu.
- y_i là giá trị thực tế của điểm dữ liệu thứ i.
- $\hat{y}_i$ là giá trị dự đoán của điểm dữ liệu thứ i.

IV. KẾT QUẢ

1) Mô hình dự đoán Vehice_mass: Mô hình dự đoán gồm 2 loại là sử dụng OverSampling và Under sampling

Sau đây là mô hình sử dụng OverSmapling trước: Sử dụng mô hình K-fold thì em đánh giá được là các con số sau:

Model	**Precision**	**Recall**	**Accuracy**
Logistic Regression	0.96	0.96	0.964
SVM	0.99	0.995	0.998
Decision Tree	0.98	0.99	0.997

Bảng II: Đánh giá Vehice_mass với OverSampling

Sau đây là sử dụng UndeSmapling:

Model	**Precision**	**Recall**	**Accuracy**
Logistic Regression	0.96	0.96	0.964
SVM	0.97	0.975	0.978
Decision Tree	0.98	0.97	0.985

Bảng III: Đánh giá Vehice_mass với UndeSmapling

Thấy được OverSmapling dễ dàng gây ra hiện tượng overfit khiến cho dự đoán quá đúng với dữ liệu đã cho nhưng sẽ sai khi gặp các dữ liệu mới . Chính vì thế trong tập data này thì sử dụng OverSampling sẽ hợp lí nhất. Trong 3 cách sử dụng thì thấy được Decision Tree là model cho được kết quả chung nhất và không bị quá overfit hay kết quả đánh giá thấp.

2) Mô hình dự đoán RoadSlope_100ms: Kết quả thể hiện thông qua bảng sau:

Số liệu cho thấy được là model Decision Tree là model tốt nhất.

Model	**MAE**	**MSE**
Linear Regression	0.3488	0.2487
SVM	0.2155	0.2155
Decision Tree	0.0890	0.0780

Bảng IV: Đánh giá RoadSlope_100ms

V. KẾT LUẬN

Số liệu cho thấy được là model Decision Tree là model tốt nhất. Mô hình Decision Tree có thể tốt hơn Linear Regression và SVM trong bài toán vì nó linh hoạt trong việc học mối quan hệ phi tuyến tính, xử lý ảnh hưởng không đồng đều, và không đặt nhiều giả định về phân phối dữ liệu. Chính vì thế mà cả hai bài toán hồi quy đều tốt khi sử dụng mô hình Decison Tree.