TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**

A blue and white logo

Description automatically generated with low confidence

**BÀI TẬP LỚN**

HỌC PHẦN: HỌC MÁY

**ĐỀ TÀI: DỰ ĐOÁN GIÁ TIỀN CHUYẾN ĐI TAXI**

Giáo viên hướng dẫn: Nguyễn Thị Kim Ngân

Nhóm sinh viên thực hiện:

1. Lương Anh Tú Lớp 63CNTT2

2. Nguyễn Quốc Huy Lớp 63CNTT2

3. Dương Ngô Quyền Lớp 63CNTT2

**Hà Nội, năm 2023**

**Phần 1: Tổng quan**

***1.Mô tả bài toán***

* **Tên bài toán**: Dự đoán giá tiền chuyến đi Taxi
* Bài toán sử dụng giải thuật của học máy để dự đoán số tiền phải thanh toán của chuyến đi dựa trên những yếu tố ảnh hưởng đến nó. Các dữ liệu thu thập được tập hợp thành 1 tập huấn luyện
* **Input**: Bao gồm những thuộc tính có ảnh hưởng tới chi phí của chuyến đi:
* trip\_duration: Thời gian đi của hành trình
* distance\_traveled: quãng đường chuyến đi
* num\_of\_passengers: Số người trên chuyến đi
* fare: giá vé cơ bản
* tip: Số tiền cho thêm
* miscellaneous\_fees: phí tính thêm (phí cầu đường, tiện ích …)
* surge\_applied: có áp dụng mức giá đột biến không
* **Ouput:** Total\_fare là giá dự đoán tương đương với giá trị thuộc tính Input
* **Tóm tắt công việc thực hiện của bài toán:** Dự đoán số tiền cần thanh toán của chuyến đi thông qua các dữ liệu đã biết

2***. Phương pháp học máy***

* Trình bày lý thuyết của 2 phương pháp Hồi quy tuyến tính, Lasso, Ridge và k-fold cross validation.

*\*\*Phương pháp Hồi quy tuyến tính:*

“Hồi quy tuyến tính” là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Nói cách khác “Hồi quy tuyến tính” là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục. Ví dụ, dự đoán thời tiết tại một địa điểm nào đó, dự đoán giá tiền nhà ,v.v

**\*Phân tích toán học:**

1.1 Giới thiệu:

Giả sử chúng ta đã có số liệu thống kê từ 1000 căn nhà trong thành phố đó, liệu rằng khi có một căn nhà mới với các thông số về diện tích, số phòng ngủ và khoảng cách tới trung tâm, chúng ta có thể dự đoán được giá của căn nhà đó không? Nếu có thì hàm dựđoán y=f(x) sẽ có dạng như thế nào.

Ở đây x=[x1,x2,x3] là một vector hàng chứa thông tin input,

y là một số vô hướng (scalar) biểu diễn output

Một cách đơn giản nhất, chúng ta có thể thấy rằng: i) diện tích nhà càng lớn thì giá nhà càng cao; ii) số lượng phòng ngủ càng lớn thì giá nhà càng cao; iii) càng xa trung tâm thì giá nhà càng giảm. Một hàm số đơn giản nhất có thể mô tả mối quan hệ giữa giá nhà và 3 đại lượng đầu vào là:

A math symbols with black text

Description automatically generated with medium confidence

trong đó, w1,w2,w3,w0là các hằng số, w0 còn được gọi là bias. Mối quan hệ y≈f(x) bên trên là một mối quan hệ tuyến tính (linear). Bài toán chúng ta đang làm là một bài toán thuộc loại regression. Bài toán đi tìm các hệ số tối ưu {w1,w2,w3,w0}chính vì vậy được gọi là bài toán Linear Regression.

1.2.Dạng của Linear Regression

Trong phương trình (1) phía trên, nếu chúng ta đặt w=[w0,w1,w2,w3].T = là vector (cột) hệ số cần phải tối ưu và =[1,x1,x2,x3] (đọc là xbar trong tiếng Anh) là vector (hàng) dữ liệu đầu vào mở rộng. Số 1 ở đầu được thêm vào để phép tính đơn giản hơn và thuận tiện cho việc tính toán. Khi đó, phương trình (1) có thể được viết lại dưới dạng:

A black text with a white background

Description automatically generated

1.3.Sai số dự đoán

Chúng ta mong muốn rằng sự sai khác e giữa giá trị thực y và giá trị dự đoán   (đọc là y hat trong tiếng Anh) là nhỏ nhất. Nói cách khác, chúng ta muốn giá trị sau đây càng nhỏ càng tốt:

A black and white math equation

Description automatically generated

trong đó hệ số 1/2 là để thuận tiện cho việc tính toán (khi tính đạo hàm thì số 1/2 sẽ bị triệt tiêu). Chúng ta cần e2 vì e=y−  có thể là một số âm, việc nói e nhỏ nhất sẽ không đúng vì khi e=−∞ là rất nhỏ nhưng sự sai lệch là rất lớn.Do đó chúng ta cần 

1.4.Hàm mất mát

Điều tương tự xảy ra với tất cả các cặp *(input,outcome)* (xi,yi),i=1,2,…,Nvới N là số lượng dữ liệu quan sát được. Điều chúng ta muốn, tổng sai số là nhỏ nhất, tương đương với việc tìm w để hàm số sau đạt giá trị nhỏ nhất:

A mathematical equation with numbers and symbols

Description automatically generated

Hàm số L(w) được gọi là **hàm mất mát** (loss function) của bài toán Linear Regression. Chúng ta luôn mong muốn rằng sự mất mát (sai số) là nhỏ nhất, điều đó đồng nghĩa với việc tìm vector hệ số w sao cho giá trị của hàm mất mát này càng nhỏ càng tốt. Giá trị của w làm cho hàm mất mát đạt giá trị nhỏ nhất được gọi là điểm tối ưu (optimal point), ký hiệu:

A black text on a white background

Description automatically generated

Trước khi đi tìm lời giải, chúng ta đơn giản hóa phép toán trong phương trình hàm mất mát (2). Đặt y=[y1;y2;…;yN] là một vector cột chứa tất cả các output của training data; A black and white symbol

Description automatically generatedlà ma trận dữ liệu đầu vào (mở rộng) mà mỗi hàng của nó là một điểm dữ liệu. Khi đó hàm số mất mát L(w) được viết dưới dạng ma trận đơn giản hơn:

A math equations and symbols

Description automatically generated with medium confidence

với  là Euclidean norm (chuẩn Euclid, hay khoảng cách Euclid), nói cách khác A black and white text

Description automatically generated with medium confidence là tổng của bình phương mỗi phần tử của vector z. Tới đây, ta đã có một dạng đơn giản của hàm mất mát được viết như phương trình (3).

2.4.Nghiệm cho bài toán linear regression

Nhận thấy rằng hàm mất mát A black text on a white background

Description automatically generated có đạo hàm tại mọi w

=>Tìm giá trị tối ưu của w có thể được thực hiện thông qua việc giải phương trình đào hàm của A black text on a white background

Description automatically generated theo w bằng 0

\*Đạo hàm theo w của hàm mất mát là :

A black text with black letters

Description automatically generated with medium confidence

Nếu ma trận XX.T khả nghịch thì phương trình trên có nghiệm duy nhất là:

A black text on a white background

Description automatically generated

Nếu ma trận XX.T không khả nghịch thì nghiệm của phương trình có thể xác định dựa vào giả nghịch đảo

A black letter on a white background

Description automatically generated

\*\*Phương pháp Lasso

**Hồi quy Lasso** là một phương pháp phân tích hồi quy thực hiện cả lựa chọn biến và chính quy hóa. Hồi quy Lasso sử dụng ngưỡng mềm. Hồi quy Lasso chỉ chọn một tập hợp con của các hiệp biến được cung cấp để sử dụng trong mô hình cuối cùng.

\*Phân tích toán học

Giả sử chúng ta có một tập dữ liệu huấn luyện gồm mẫu dữ liệu (instances) và nhãn tương ứng (labels). Chúng ta muốn tìm một mô hình hồi quy tuyến tính có dạng:

y = Xw + b

Trong đó:

* y là vector nhãn (labels).
* X là ma trận mẫu dữ liệu (instances), với mỗi hàng là một mẫu và mỗi cột là một đặc trưng.
* w là vector trọng số (weights) tương ứng với các đặc trưng.
* b là sai số chệch (bias).

Phương pháp LASSO thực hiện tối ưu hóa hàm mất mát (loss function) theo công thức:

L(w) = 1/2 \* ||y - Xw - b||^2 + alpha \* ||w||\_1

Trong đó:

* ||.|| là norm L2 của một vector.
* ||w||\_1 là norm L1 của vector trọng số w.
* alpha là tham số điều chỉnh sự đóng góp của regularization.

Để tìm giá trị của w mà làm giả thiết hàm mất mát đạt giá trị nhỏ nhất, chúng ta cần tìm giá trị w mà làm giảm giá trị của hàm mất mát. Điều này có thể được thực hiện bằng cách lấy đạo hàm riêng của hàm mất mát theo w và đặt nó bằng 0:

∂L(w)/∂w = 0

Tuy nhiên, vì norm L1 không khả vi tại 0, không thể tìm được giá trị chính xác của w bằng cách lấy đạo hàm. Thay vào đó, phương pháp LASSO sử dụng một kỹ thuật gọi là subgradient để tìm giá trị gần đúng của w.

Subgradient của hàm mất mát L(w) tại w là một vector g được xác định như sau:

g = -X^T(y - Xw - b) + alpha \* sign(w)

Trong đó:

* X^T là ma trận chuyển vị của X.
* sign(w) là hàm sign trả về dấu của từng phần tử trong w.

Quá trình ước lượng trọng số w trong phương pháp LASSO thường được thực hiện bằng các thuật toán tối ưu đặc biệt như coordinate descent hoặc gradient descent, sử dụng subgradient để tìm giá trị gần đúng của w.

Khi giá trị của alpha tăng lên, phương pháp LASSO sẽ có xu hướng đưa các trọng số không quan trọng về 0, từ đó thực hiện lựa chọn đặc trưng tự động.

\*\*Phương pháp hồi quy Ridge

**Hồi quy Ridge** là một kỹ thuật để phân tích dữ liệu hồi quy nhiều lần. Khi xảy ra đa cộng tuyến, các ước lượng bình phương nhỏ nhất là không chệch. Một mức độ chệch được thêm vào các ước tính hồi quy và kết quả là hồi quy sườn núi làm giảm các sai số tiêu chuẩn.

\*Phân tích toán học

Giả sử chúng ta có một tập dữ liệu huấn luyện gồm mẫu dữ liệu (instances) và nhãn tương ứng (labels). Chúng ta muốn tìm một mô hình hồi quy tuyến tính có dạng:

y = Xw + b

Trong đó:

* y là vector nhãn (labels).
* X là ma trận mẫu dữ liệu (instances), với mỗi hàng là một mẫu và mỗi cột là một đặc trưng.
* w là vector trọng số (weights) tương ứng với các đặc trưng.
* b là sai số chệch (bias).

Phương pháp Ridge thực hiện tối ưu hóa hàm mất mát (loss function) theo công thức:

L(w) = 1/2 \* ||y - Xw - b||^2 + alpha \* ||w||\_2^2

Trong đó:

* ||.|| là norm L2 của một vector.
* ||w||\_2^2 là bình phương của norm L2 của vector trọng số w.
* alpha là tham số điều chỉnh sự đóng góp của regularization.

Để tìm giá trị của w mà làm giảm giá trị của hàm mất mát, chúng ta cần tìm giá trị w mà làm cho đạo hàm riêng của hàm mất mát theo w bằng 0:

∂L(w)/∂w = 0

Đạo hàm riêng này có thể tính được dễ dàng bằng phép tính đạo hàm thông thường. Khi giải phương trình ∂L(w)/∂w = 0, ta thu được:

(X^T*X + alpha*I)w = X^T\*y

Trong đó:

* X^T là ma trận chuyển vị của X.
* I là ma trận đơn vị.

Giải phương trình trên, chúng ta có:

w = (X^T*X + alpha*I)^(-1) \* X^T\*y

Quá trình ước lượng các trọng số w trong phương pháp Ridge được thực hiện bằng cách tính toán ma trận (X^T*X + alpha*I)^(-1) và áp dụng nó vào phép nhân ma trận để tính toán w.

Khi giá trị của alpha tăng lên, phương pháp Ridge sẽ có xu hướng giảm kích thước của các trọng số, từ đó giảm khả năng overfitting và cải thiện tính tổng quát hóa của mô hình.

*\*\*Phương pháp k-fold cross validation*

Cross validation là một phương pháp thống kê được sử dụng để ước lượng hiệu quả của các mô hình học máy. Nó thường được sử dụng để so sánh và chọn ra mô hình tốt nhất cho một bài toán. Kỹ thuật này dễ hiểu, dễ thực hiện và cho ra các ước lượng tin cậy hơn so với các phương pháp khác

Cross validation là một cải tiến của validation với lượng dữ liệu trong tập validation là nhỏ nhưng chất lượng mô hình được đánh giá trên nhiều tập validation khác nhau. Một cách thường đường sử dụng là chia tập training ra k tập con không có phần tử chung, có kích thước gần bằng nhau. Tại mỗi lần kiểm thử , được gọi là run, một trong số k tập con được lấy ra làm validate set. Mô hình sẽ được xây dựng dựa vào hợp của k−1tập con còn lại. Mô hình cuối được xác định dựa trên trung bình của các train error và validation error.

=>Cách làm này còn có tên gọi là **k-fold cross validation**.

**\*Phân tích toán học**

K-fold cross validation: Chia training set thành K tập con không giao nhau, có kích thước gần bằng nhau. X={X1 , X2 , …., XK}

Tại mỗi lần kiểm thử:

▪ 1 tập con được lấy ra làm validation set

▪ K-1 tập còn lại được dùng để xây dựng mô hình

▪ Mô hình cuối được xác định dựa trên trung bình của các train error và validation error (CV(w) là nhỏ nhất)

Tìm k mô hình

Lần 1: Validation set: X1 , Training set: X\X1={X2 , X3 , …, Xk}= > M1

Lần 2: Validation set: X2 , Training set: X\X2={X1 , X3 , …, Xk}=>M2

Lần k: Validation set: Xk , Training set: X\Xk={X1 , X2 , …, Xk-1}=>Mk

Kỹ thuật này thường bao gồm các bước như sau:

1. Xáo trộn dataset một cách ngẫu nhiên

2. Chia dataset thành k nhóm

3. Với mỗi nhóm:

-Sử dụng nhóm hiện tại để đánh giá hiệu quả mô hình

-Các nhóm còn lại được sử dụng để huấn luyện mô hình

-Huấn luyện mô hình

-Đánh giá và sau đó hủy mô hình

1. Sử dụng nhóm hiện tại để đánh giá hiệu quả mô hình

2. Các nhóm còn lại được sử dụng để huấn luyện môhình

3. Huấn luyện mô hình

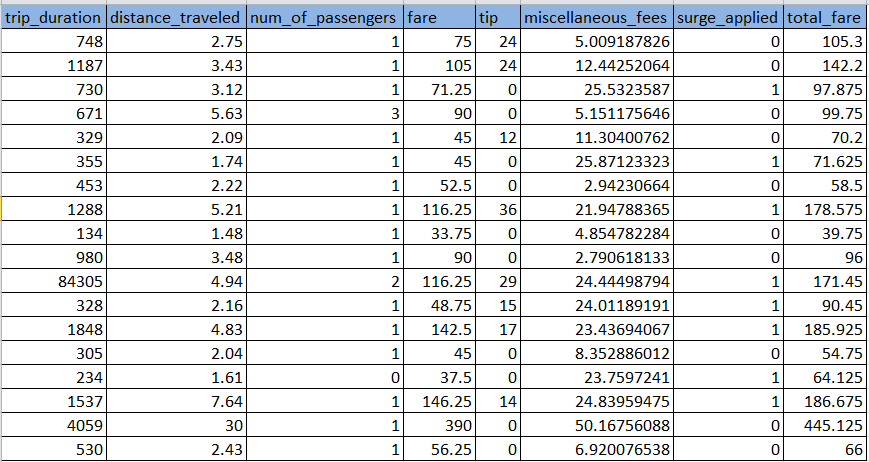
4. Đánh giá và sau đó hủy mô hình

4. Tổng hợp hiệu quả của mô hình dựa từ các số liệu đánh giá

**Phần 2: Thực nghiệm**

***1.Mô tả tập dữ liệu của bài toán***

-Tập dữ liệu sau khi thu thập:



* Thu thập được tổng cộng 209673 mẫu
* Ma trận dữ liệu X bao gồm các cột:

1. trip\_duration: Thời gian đi của hành trình
2. distance\_traveled: quãng đường chuyến đi
3. num\_of\_passengers: Số người trên chuyến đi
4. fare: giá vé cơ bản
5. tip: Số tiền cho thêm
6. miscellaneous\_fees: phí tính thêm (phí cầu đường, tiện ích …)
7. surge\_applied: có áp dụng mức giá đột biến không

* Nhãn lớp Y bao gồm cột: Total\_fare
* Chia tập dữ liệu thành 2 phần: 70% dùng để huấn luyện mô hình, 30% dùng để kiểm tra sự phù hợp của mô hình.

***2. Phân tích kết quả của chương trình***

* Công thức của 4 độ đo R2,MAE,NSE,RMSE

A table of mathematical equations

Description automatically generated

* Kết quả dự đoán của mô hình với kết quả thực tế:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | LinearRegression | Lasso | Ridge | K-fold |
| R2\_score | 0.9990966851 | 0.9997911071 | 0.9990967126 | 0.9992313123 |
| MAE | 1.2361096641849112 | 1.237939468308751 | 1.236109900310615 | 1.23515977738795 |
| NSE | 0.9990966851083478 | 0.9997911070728116 | 0.9990967126178264 | 0.9992313122560251 |
| RMSE | 2.9836852023880835 | 1.4348129351950853 | 2.9836397695883896 | 2.7523800025738727 |

* Mô hình tốt nhất là mô hình Lasso do có độ đo R2 cao nhất (0.9997911071)
* 1 số kết quả của phương pháp hồi quy tuyến tính:

Thực tế - Dự đoán = chênh lệch:

158.925 - 161.011452831453 = 2.0864528314529878

189.75 - 190.19649006328237 = 0.4464900632823685

51.0 - 49.51173552746946 = 1.4882644725305383

77.25 - 75.38705484190787 = 1.8629451580921312

97.2 - 97.38211238493884 = 0.18211238493883286

51.0 - 53.40087480466616 = 2.4008748046661594

202.8 - 203.39407705548783 = 0.5940770554878156

95.325 - 96.99409544822389 = 1.6690954482238851

69.75 - 70.08230246520502 = 0.33230246520501794

69.75 - 68.14793842628045 = 1.6020615737195527

339.75 - 341.98710479773206 = 2.2371047977320586

* 1 số kết quả của phương pháp Lasso:

158.925 - 161.0924354791185 = 2.1674354791184953

189.75 - 190.1702419529318 = 0.42024195293180355

51.0 - 49.58725748332078 = 1.4127425166792165

77.25 - 75.45683915408495 = 1.7931608459150539

97.2 - 97.46681993992938 = 0.2668199399293769

51.0 - 53.52111898895642 = 2.5211189889564167

202.8 - 203.4142822021436 = 0.6142822021435848

95.325 - 96.80511162092785 = 1.4801116209278433

69.75 - 70.17536125316435 = 0.425361253164354

69.75 - 68.29568898596982 = 1.454311014030182

339.75 - 341.8523598176379 = 2.1023598176378755

* 1 số kết quả của phương pháp Ridge:

158.925 - 161.0114597566606 = 2.086459756660588

189.75 - 190.196485451824 = 0.4464854518240031

51.0 - 49.51173645456415 = 1.4882635454358493

77.25 - 75.38705498810047 = 1.8629450118995265

97.2 - 97.38211515593241 = 0.1821151559324079

51.0 - 53.400879614323856 = 2.4008796143238555

202.8 - 203.39407871198785 = 0.5940787119878337

95.325 - 96.99408978452675 = 1.6690897845267472

69.75 - 70.08230486977855 = 0.3323048697785538

69.75 - 68.14794630251336 = 1.6020536974866388

339.75 - 341.98709716696055 = 2.237097166960552

- 1 số kết quả của phương pháp k-fold cross validation:

158.925 - 161.01853594858716 = 2.093535948587146

189.75 - 190.1848326217355 = 0.4348326217354952

51.0 - 49.50975608117771 = 1.4902439188222871

77.25 - 75.38481665879101 = 1.86518334120899

97.2 - 97.38584136688957 = 0.1858413668895622

51.0 - 53.40041391288844 = 2.400413912888439

202.8 - 203.4031757274324 = 0.6031757274323866

95.325 - 96.99641515338372 = 1.6714151533837196

69.75 - 70.08067890727159 = 0.3306789072715901

69.75 - 68.14745009402134 = 1.6025499059786625

339.75 - 341.96300243498484 = 2.213002434984844

**Kết luận**

* Tóm lược các nội dung chính mà bài tập lớn làm được

Qua các thuật toán Linear Regression, Lasso, Ridge, Linear Regression kết hợp với k-fold kross validation chúng em đã có thể tìm hiểu và tiếp thu thêm nhiều kiến thức của Hồi quy tuyến tính

Chúng em đã biết thêm về các độ đo R2, NSE, MAE , RMSE và đã áp dụng thành công vào dự án của chúng em

**Tài liệu tham khảo**

* <https://machinelearningcoban.com/2016/12/28/linearregression/>
* <https://www.kaggle.com/datasets/raviiloveyou/predict-taxi-fare-with-a-bigquery-ml-forecasting>
* <https://trituenhantao.io/kien-thuc/gioi-thieu-ve-k-fold-cross-validation/>
* Các file bài giảng cô đã gửi : Chapter 2.1,Chapter 2.2,Chapter 8