TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

----- □ & □ -----

Đề tài: Dự đoán tiền bảo hiểm y tế dựa trên một số giải thuật học máy cơ bản

Học phần: Nhập môn trí tuệ nhân tạo

Giảng viên: **TS Nguyễn Nhật Quang**

Nhóm sinh viên thực hiện:

STT	Họ và tên	MSSV
1	Nguyễn Phúc Tân	20194163
2	Hữu Tường Tú	20194395

Hà Nội, tháng 7 năm 2022

Tổng quan

Các công ty về bảo hiểm thường rất chú trọng tới việc dự báo tiền chi tiêu cho bảo hiểm của công dân. Từ các số liệu dự đoán các công ty có thể phân tích số liệu, từ đó có kế hoạch phân bố hiệu quả các nguồn lực để quản lý dòng tiền và tăng trưởng trong tương lai. Việc dự báo cũng giúp họ ước tính chính xác hơn về chi phí và doanh thu, dựa vào đó có thể dự đoán hiệu quả hoạt động của công ty trong ngắn hạn và dài hạn. Do đó, các công ty cần nhiều giải thuật để có thể dự đoán càng chính xác càng tốt tiền chi tiêu này. Vậy nên, đề tài "dự đoán tiền bảo hiểm y tế dựa trên một số giải thuật học máy cơ bản" được đề xuất.

Source code và hướng dẫn chạy được lưu tại: https://github.com/huutuongtu/AI_project

Giới thiệu

Bài viết được chia làm ba phần:

- Phần một: Giới thiệu về giải thuật học máy cơ bản. Phần đầu tiên sẽ trình bày về các giải thuật các loại bài toán trong Machine Learning được sử dụng để áp dụng vào bài toán.
- Phần hai: Ứng dụng các giải thuật Machine learning giải quyết bài toán dự đoán tiền chi tiêu cho bảo hiểm. Ở phần này sẽ trình bày về tập dữ liệu sử dụng, các bước xử lý dữ liệu, huấn luyện dữ liệu và kết quả thu được.
- Phần ba: Tổng kết. Phần cuối sẽ nêu ra những gì tổng hợp được trong quá trình nghiên cứu.

MỤC LỤC

CHƯƠNG 1

GIỚI THIỆU CÁC GIẢI THUẬT HỌC MÁY CƠ BẢN TRONG BÀI TOÁN

1.1 Linear Regression

1.1.1 Phân tích

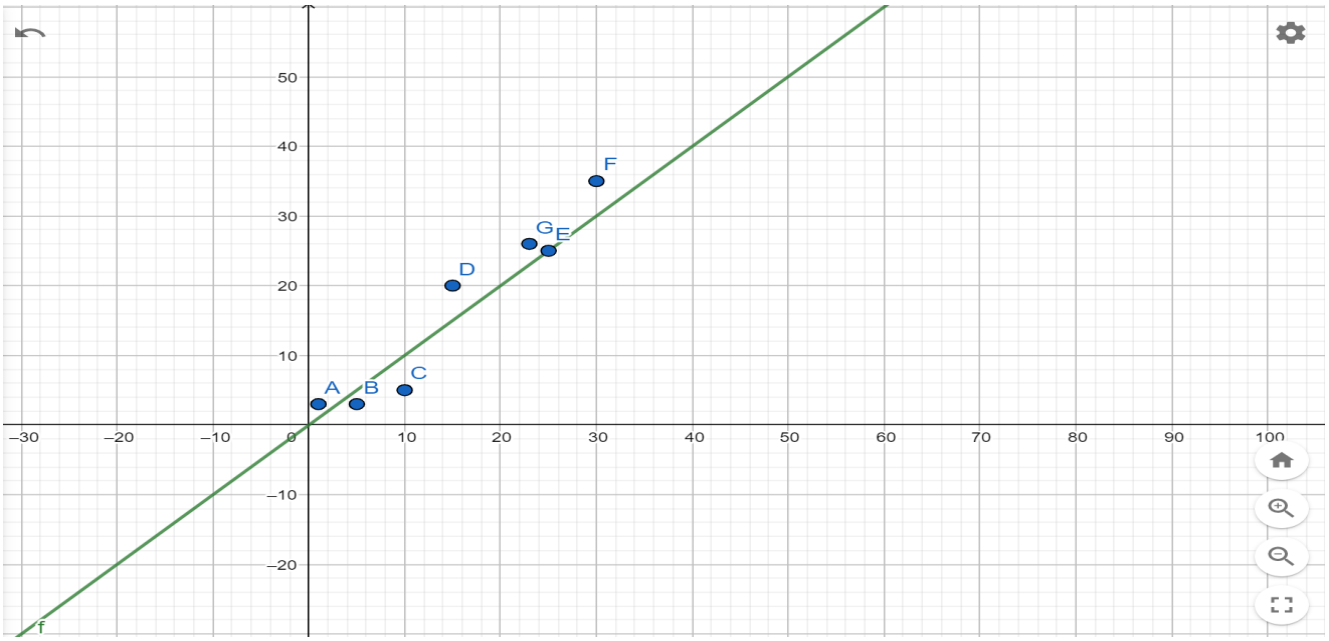
Linear Regression hay còn gọi là Hồi Quy Tuyến Tính là một trong những giải thuật cơ bản và đơn giản nhất của Machine Learning. Đây là một một thuật toán Supervised learning.

Ta sẽ nêu một ví dụ một bài toán đơn giản cho bài toán như giá thuê nhà. Khi ta so sánh một căn nhà rộng và một căn nhà nhỏ hơn ở cùng một vị trí. Tất nhiên căn nhà rộng hơn sẽ được thuê với giá cao hơn. Đây chính là cách nghĩ đơn giản nhất. Một hàm số đơn giản nhất có thể mô phỏng giữa đầu ra giá nhà và đầu vào diện tích là:

$$y \sim f(x) = w.x + b$$

Với x là diện tích căn nhà và y là giá nhà, w là một hằng số và b là bias. Bài toán đặt ra là đi tìm giá trị tối ưu w,b sao cho y gần với f(x) nhất. Do đó đây được gọi là bài toán Linear Regression.

Bài toán có thể được mô phỏng dưới dạng hình vẽ khi ta coi đường thẳng sẽ mô tả sự phụ thuộc của hàm mục tiêu với các thuộc tính



Để tối ưu hàm mục tiêu dự đoán, ta sẽ dùng một cách đo đó là hàm loss tức hàm chênh lệch giữa đầu ra cần dự đoán và đầu ra của mô hình đi tối ưu. Chúng ta mong muốn giá trị dự đoán càng gần giá trị thật tức là hàm loss này sẽ càng nhỏ càng tốt. Sai số của một đơn vị dữ liệu có thể tính bởi:

$$\frac{1}{2}(y - f(x))^2$$

Điều tương tự xảy ra với tất cả các cặp dữ liệu, nghĩa là ta sẽ muốn tổng sai số của tất cả các dữ liệu là nhỏ nhất. Nghĩa là điều này tương đương với việc đi tìm giá trị nhỏ nhất hàm sau:

$$L(w) = \frac{1}{2}\sum_{i=1}^{N}(y_i - w.x_i)^2$$

Cách phổ biến nhất để tìm nghiệm cho một bài toán tối ưu (chúng ta đã biết từ khi học cấp 3) là giải phương trình đạo hàm (gradient) bằng 0. Với một mô hình tuyến tính, việc giải đạo hàm bằng 0 này sẽ không quá phức tạp.

Ta sẽ đạo hàm theo w của hàm mất mát. Xong giải phương trình bằng 0.

Gọi X là tập dữ liệu huấn luyện và Y là tập nhãn, W là tập tham số ta có:

$$\frac{\partial L(w)}{\partial w} = X^T(XW - Y) = 0$$

Giải phương trình này ta được:

$$X^T XW = X^T Y = b; A = X^T X \Rightarrow w = A^{-1}b$$

Nếu A khả nghịch.

Nếu A không khả nghịch, ta có thể dùng giả nghịch đảo hoặc dùng một phương pháp khác có tên là Gradient Descent.

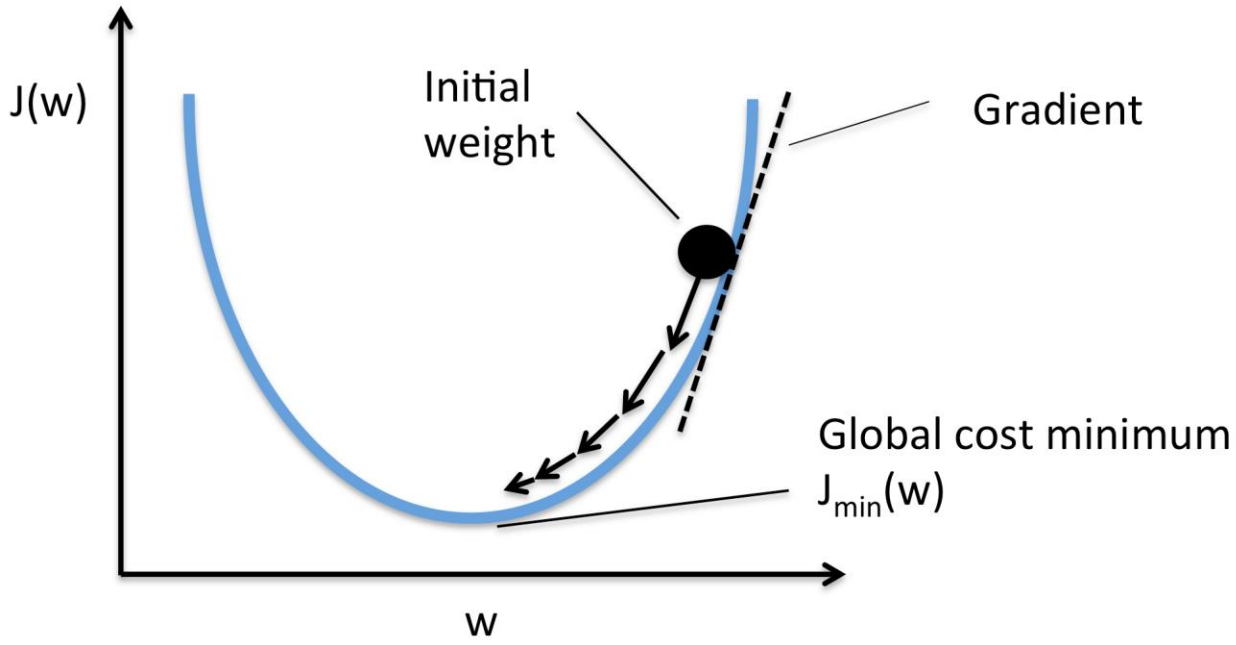
1.1.2 Gradient Descent

Hướng tiếp cận phổ biến nhất là xuất phát từ một điểm mà chúng ta coi là *gần* với nghiệm của bài toán, sau đó dùng một phép toán lặp để *tiến dần* đến điểm cần tìm, tức đến khi đạo hàm gần với 0. Gradient Descent và các biến thể của nó là một trong những phương pháp được dùng nhiều nhất.

Ý tưởng của GD là xuất phát từ một điểm ta bất kỳ ta coi là gần với nghiệm bài toán sau đó ta sẽ dùng phép lặp để tiền dần đến điểm có đạo hàm gần với 0.

Ta sẽ tính đạo hàm và cập nhật đạo hàm liên tục ngược hướng tích với một biến được gọi là learning rate. Đạo hàm sẽ nhỏ dần qua mỗi lần cập nhật và hội tụ dần đến 0.

Mô phỏng GD có thể được biểu diễn ở hình sau



1.1.3 Cook’s Distance

Các phần tử bất thường (Outliers) có thể là nguyên nhân gây ra sự vi phạm các giả thiết của mô hình hồi quy tuyến tính và làm méo mó kết quả dự báo của mô hình. Ở trong bài toán này, cook’s distance được sử dụng để tránh các phần tử bất thường đó

Trong thống kê cook’s distance hay là khoảng cách của Cook là một ước lượng tính thường được sử dụng để tính toán ảnh hưởng của một điểm dữ liệu khi thực hiện một bài toán hồi quy.

Cook’s distance được tính toán dựa trên khoảng cách giữa các điểm dữ liệu đến siêu phẳng dự đoán.

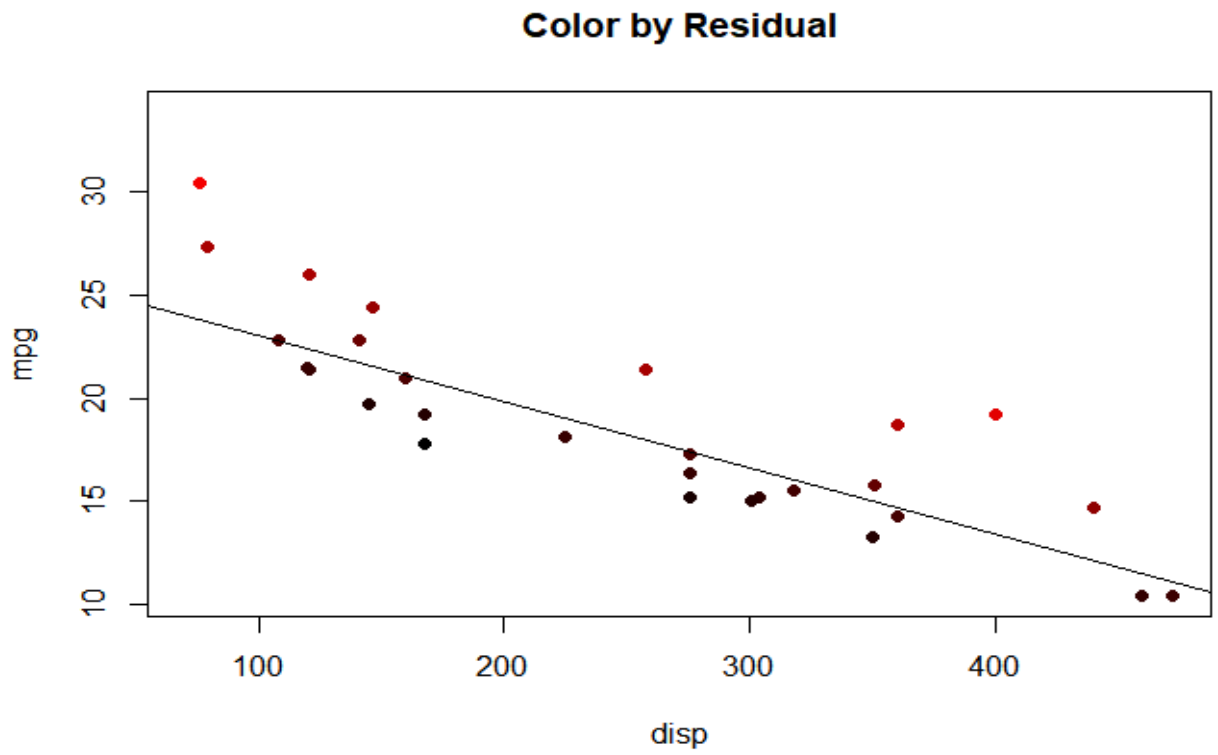
Cook’s distance sẽ được tính toán như sau:

$$D_i = \frac{\sum_{j=1}^{N} (y_j - y_{j(i)})}{p.MSE}$$

Với Di là cook’s distance của đơn vị dữ liệu i. yj là giá trị dự đoán của điểm dữ liệu thứ j. Và yj(I) là giá trị dự đoán khi ta bỏ điểm dữ liệu thứ i ra ngoài, p là số hệ số trong mô hình hồi quy (có thể hiểu là số feature). Khi Di càng lớn tức là điểm dữ liệu này càng nhiễu (ngoại lai). Việc loại bỏ các điểm ngoại lai có thể làm mô hình trở nên tốt hơn.

Cook’s distance có thể được mô phỏng dưới dạng hình vẽ như sau:

Với các điểm càng gần màu đỏ nghĩa là cook’s distance càng lớn và các điểm đen là nhỏ dần.



1.2 K-nearest neighbor

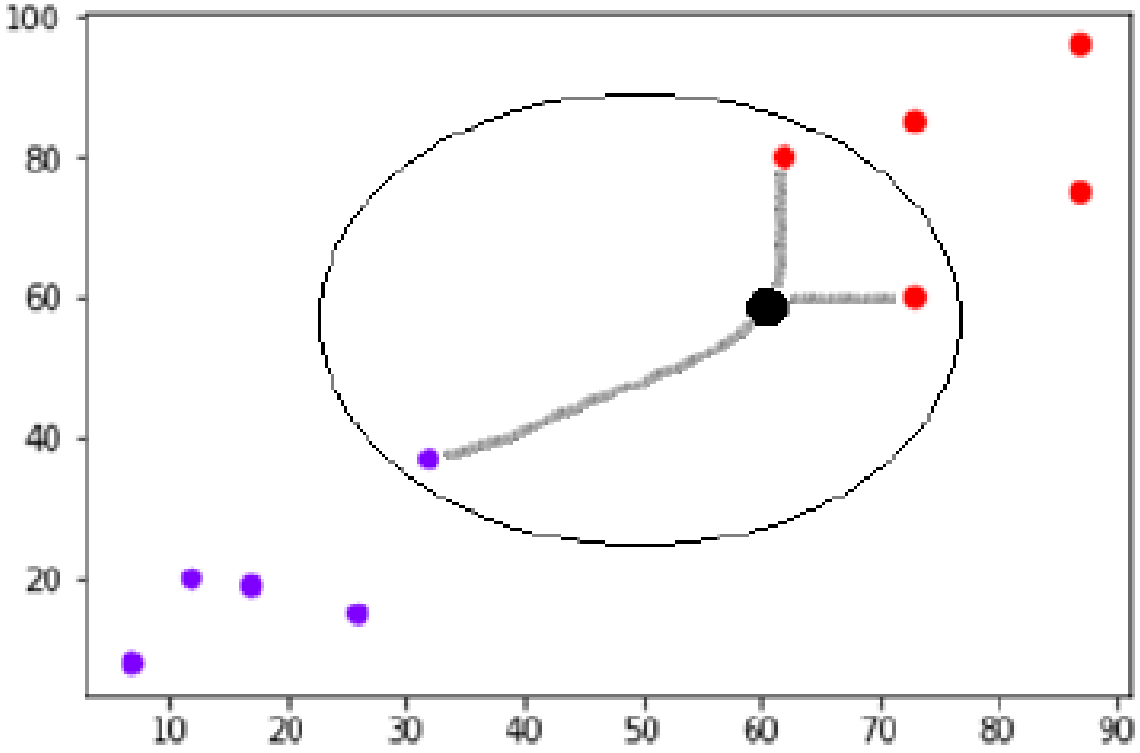
1.2.1 Tổng quát

KNN là một trong những thuật toán supervised cơ bản nhất của học máy.

KNN còn có một tên khác là học lười (lazy learning), gọi nó là học lười vì KNN gần như không cần phải train từ tập dữ liệu. Nó chỉ dựa trên những dữ liệu đã có trước đây để dự đoán. Learning phase sẽ chính là phase lưu trữ tất cả các dữ liệu. Việc dự đoán đầu ra / nhãn chỉ dựa vào những đơn vị dữ liệu gần nó nhất. Do đó KNN cũng không cần bất kỳ tham số nào. K ở đây để chỉ số lượng *hàng xóm* xử dụng để dự đoán đầu ra. Cách xác định K chỉ có thể thử nghiệm chứ không cố định.

Hai điểm chính của thuật toán KNN:

- Phép đo giống nhau được xác định bởi *khoảng cách* giữa các điểm dữ liệu. Khoảng cách này tùy vào định nghĩa theo ý muốn
- Các *hàng xóm* được sử dụng để dự đoán đầu ra



Mô phỏng KNN đơn giản ta dễ dàng thấy điểm đen *gần* với những điểm đỏ hơn nên ta có thể đánh điểm đen thuộc đỏ.

1.2.2 Khoảng cách

Khoảng cách này có thể tính bằng một số cách đơn giản như sau:

- Có thể sử dụng khoảng cách giữa 2 vector (thường là dùng chuẩn 2).
- Sử dụng cosine similarity giữa 2 vector ...

Distance measure	Formula
Euclidean	$D = \sqrt{\left(x_1 - y_1\right)^2 + \left(x_2 - y_2\right)^2 + \ldots + \left(x_n - y_n\right)^2}$
City block	$D = \left\|x_1 - y_1\right\| + \left\|x_2 - y_2\right\| + \ldots\ldots + \left\|x_n - y_n\right\|$
Cosine	$D = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}$
Correlation	$D = \frac{N\sum xy - \left(\sum x\right)\left(\sum y\right)}{\sqrt{\left[N\sum x^2 - \left(\sum x\right)^2\right]\left[N\sum y^2 - \left(\sum y\right)^2\right]}}$

1.2.3 KNN regression

Khi ta sử dụng KNN để phân lớp, ta đơn giản chỉ cần tính xem đa số hàng xóm của điểm dữ liệu thuộc lớp nào thì ta sẽ chọn nhãn theo điểm dữ liệu đó. Nhưng với KNN regression ta sẽ có cách khác.

Cách thứ nhất là tính trung bình các hàng xóm xung quanh nó (uniform)

Cách thứ hai là dựa vào khoảng cách với các hàng xóm xung quanh có thể tính theo tỉ lệ nghịch với khoảng cách

CHƯƠNG 2: ỨNG DỤNG CÁC GIẢI THUẬT VÀO GIẢI QUYẾT BÀI TOÁN

2.1 Dataset

Dataset sử dụng trong bài toán: Insurance.csv - Nguồn Kaggle:

Medical Cost Personal Datasets | Kaggle

Data gồm 1338 đơn vị dữ liệu với 7 thuộc tính (6 thuộc tính đầu làm input, expenses output):

Age: Nói về số tuổi của người đại diện: 18 -> 64 tuổi

Sex: Giới tính người đại diện: Male/Female

Bmi: Chỉ số cơ thể người đại diện: 16 -> 53.1

Children: Số con của người đại diện: Từ 0->5

Smoker: Người đại diện có hút thuốc hay không: Yes/No

Region: Địa chỉ người đại diện: Southeast, Southwest, Northeast, Northwest

Expenses: Số tiền chi: Từ 1100 -> 64000

2.2 Tiền xử lý dữ liệu

Chuyển những thuộc tính không phải numeric về numeric:

```
{"sex":     {"female": 0, "male": 1},
 "smoker":  {"no": 0, "yes": 1},
 "region":  {'northeast': 0, 'southeast': 1,'southwest': 2, 'northwest': 3}}
```

Xóa bỏ những dữ liệu không có thuộc tính (nan)

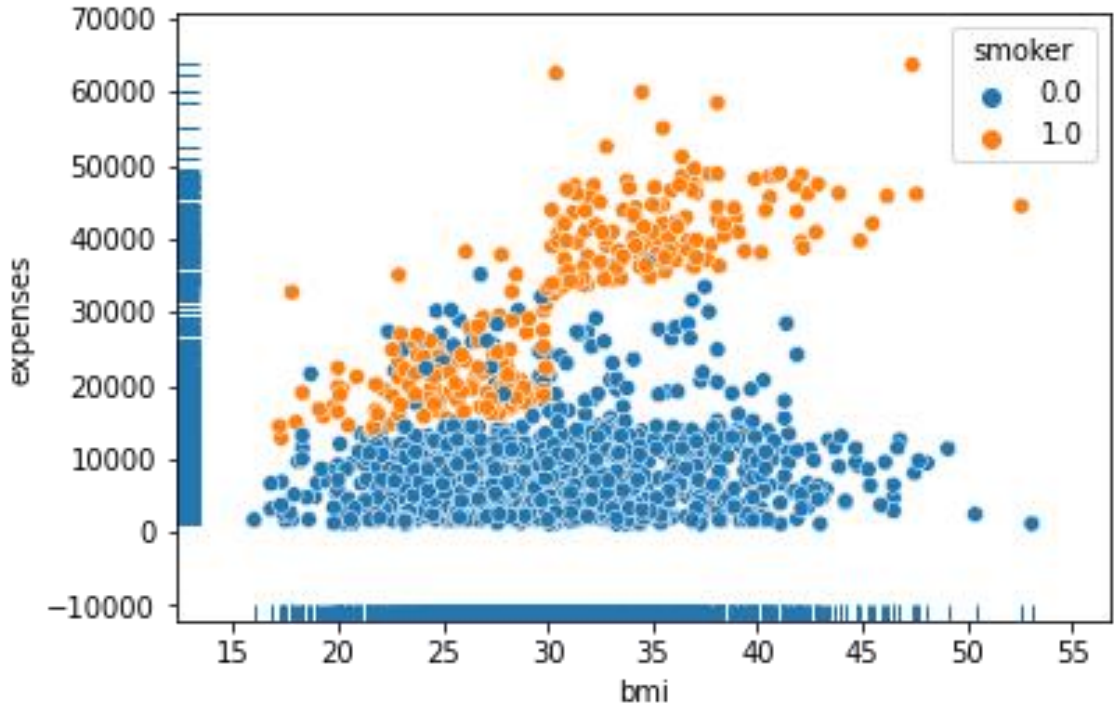
Chuẩn hóa dữ liệu:

Đưa hết dữ liệu (không có expenses) về khoảng 0->1 bằng cách sử dụng standard scaler

```
(insurance-insurance.min())/(insurance.max()-insurance.min())
```

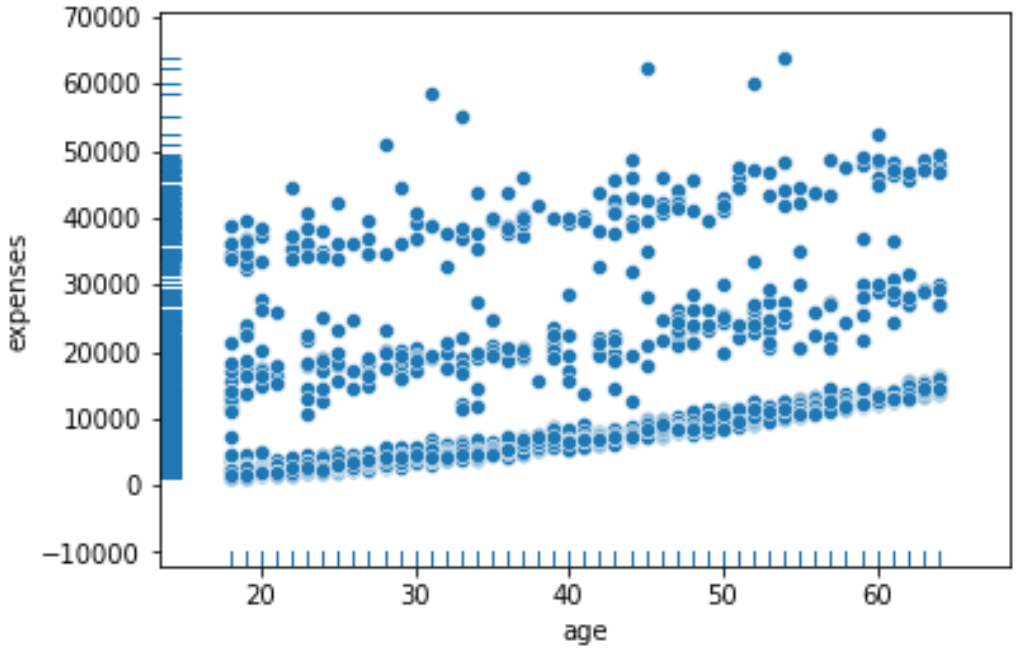
2.3 Mô phỏng dữ liệu

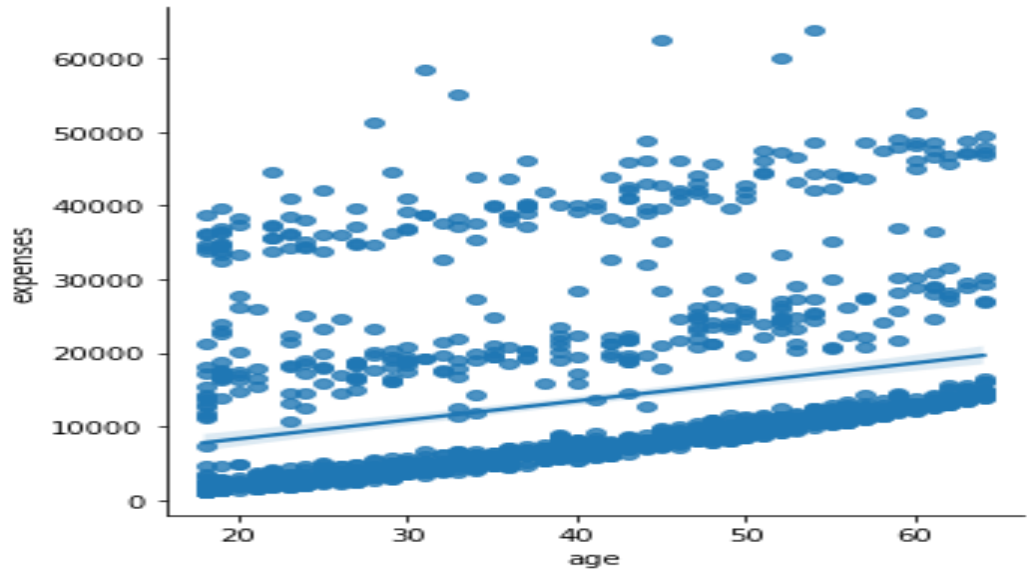
Vẽ mối quan hệ giữa bmi, insurance, smoker:



Nhận xét: Những người có bmi cao sẽ có xu hướng expenses cao hơn, những người có hút thuốc sẽ có xu hướng expenses cao hơn

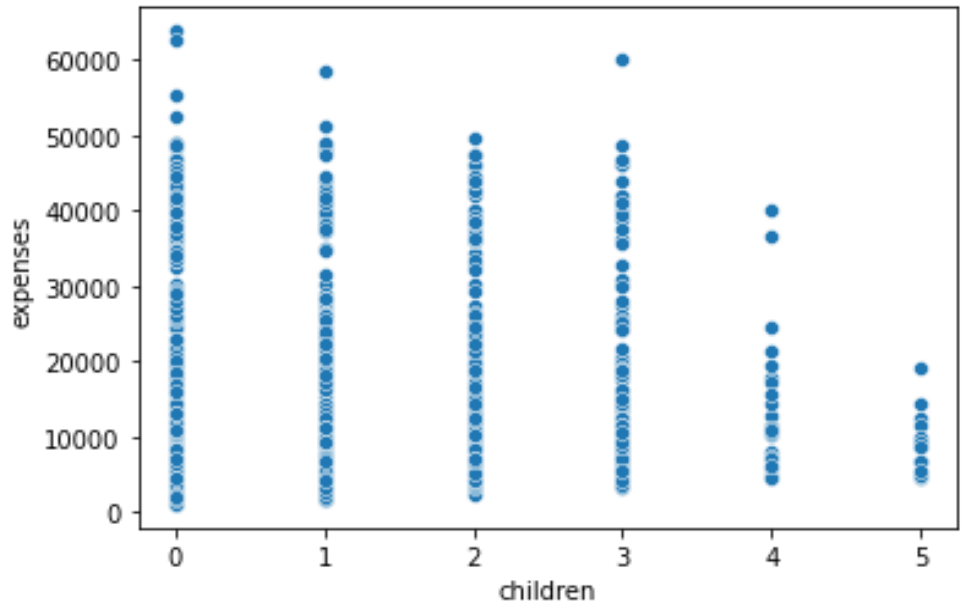
Vẽ mối quan hệ giữa age, insurance:





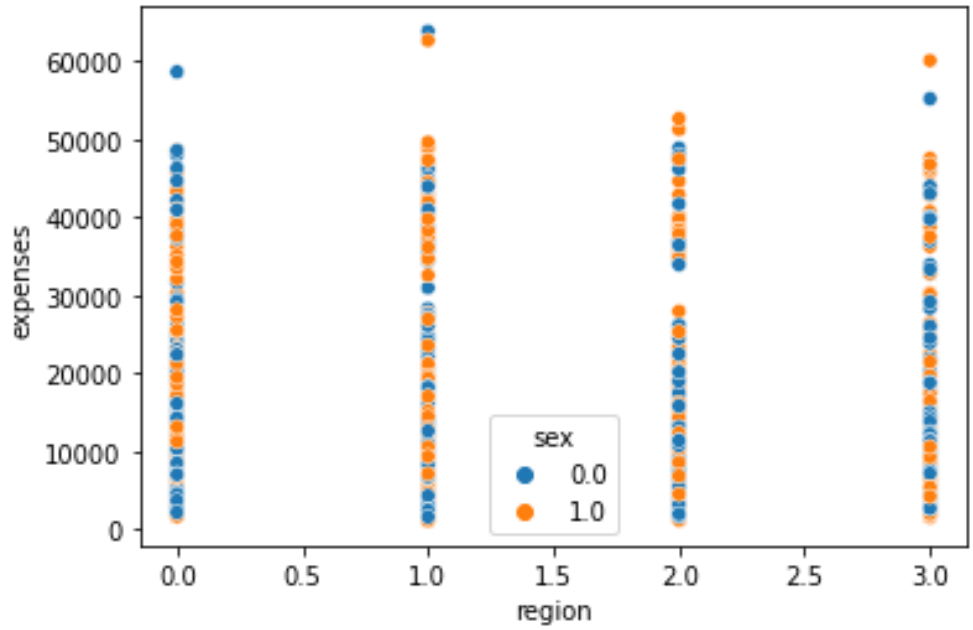
Nhận xét: Những người có tuổi cao hơn sẽ có xu hướng expenses cao hơn. Dữ liệu phân thành 3 cụm. Tập trung nhiều nhất ở cụm thấp nhất.

Vẽ mối quan hệ giữa children, expenses:



Nhận xét: Càng nhiều con sẽ càng có expenses thấp hơn

Vẽ mối quan hệ giữa region, sex, expenses:



Nhận xét: Dữ liệu phân bố theo region và sex so với expenses khá đồng đều => Expenses gần như không phụ thuộc vào 2 thuộc tính này ??

Ta sẽ chia tập dữ liệu với 2/3 dữ liệu thuộc tập train và 1/3 dữ liệu với tập test

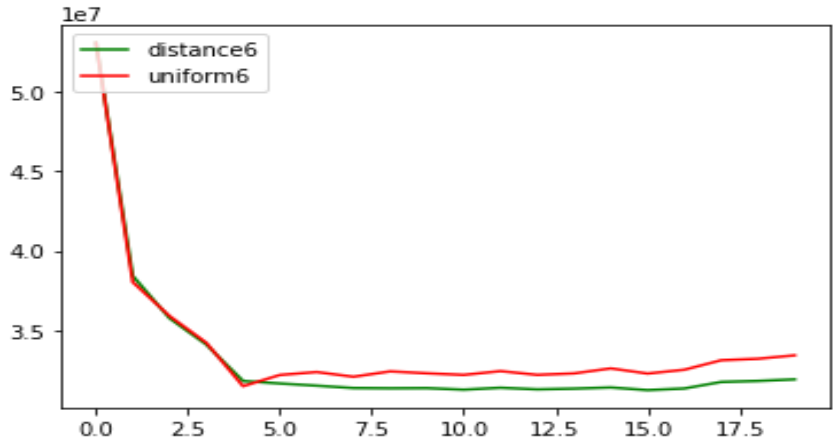
2.4 Huấn luyện dữ liệu và kết quả

Ở đây ta sẽ sử dụng 2 model là Linear Regression và KNN để train thử kết quả nguyên bản. Sau đó ta sẽ thử bỏ 2 thuộc tính Sex và Region đi và tiếp tục train với 2 model. Cuối cùng ta thử remove outlier với cook's distance và train thử rồi so sánh kết quả thu được.

Ta sử dụng MSE làm hàm loss so sánh các mô hình với nhau

Ta sẽ thử KNN với K chạy từ 1 đến 20 với tập dữ liệu để so sánh và sử dụng uniform/ distance làm trọng số để tính toán đầu ra

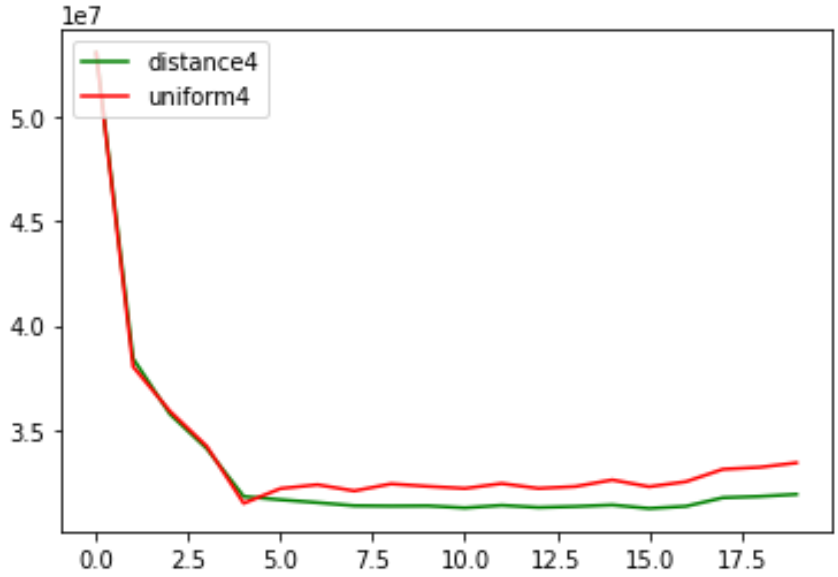
Với dữ liệu nguyên bản (đủ 6 thuộc tính) K=11 sẽ tốt cho distance và K=5 sẽ tốt cho uniform với MSE lần lượt tại 2 K này là 31282387.833133336 và 31498814.653729822. Với Linear Regression MSE là 35346221.74168072



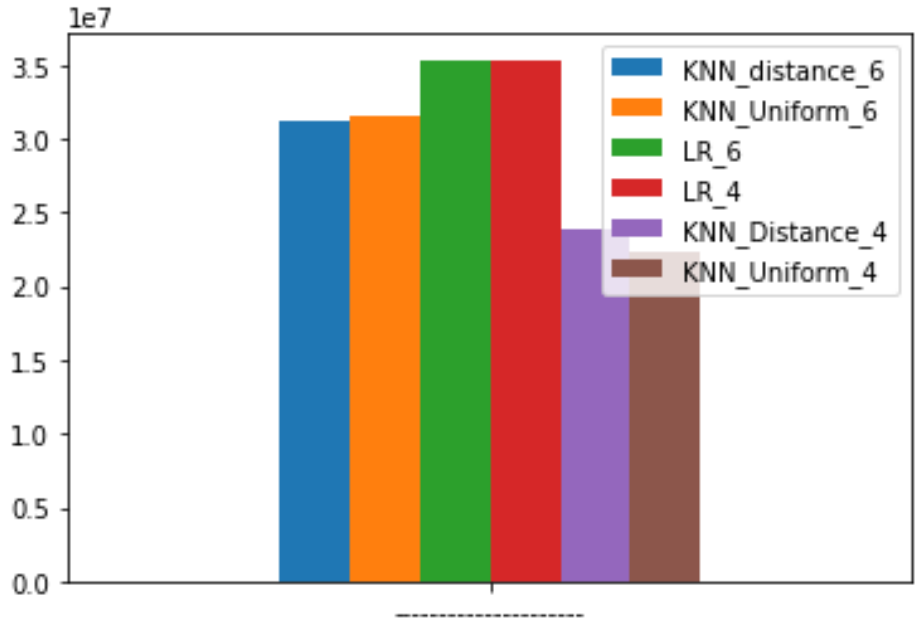
Với trung bình Expenses là 13200 RMSE lên đến gần 6000. Đây là con số lệch khá cao lên đến gần 50%.

Do MSE với data không xử lý khá lớn, vậy nên ta sẽ thử sử dụng 2 model với 4 thuộc tính (Bỏ Sex và Region)

Với dữ liệu 4 thuộc tính K=15 sẽ tốt cho cả distance và uniform với MSE lần lượt là 23898941.25042891 và 22300507.74324033 còn Linear Regression vẫn không tốt lên: 35297252.52263133

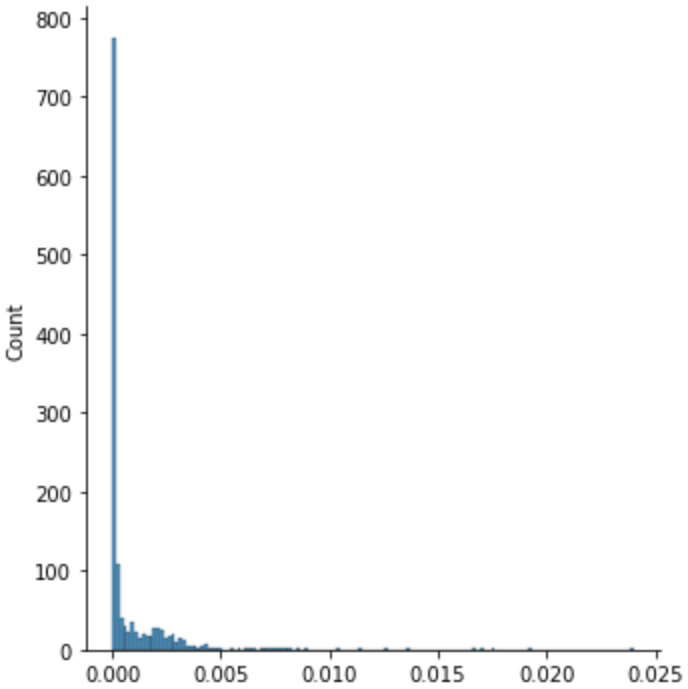


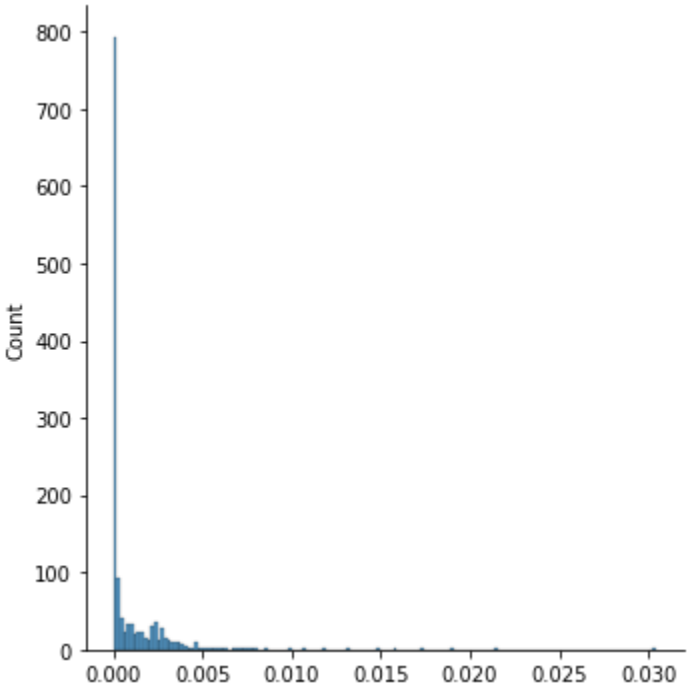
Đây là bảng so sánh 6 mô hình đầu với min MSE của mỗi mô hình:



So sánh thấy rằng KNN_uniform_4 hiện đang là tốt nhất với MSE 22300507 RMSE 4700 so với MEAN expenses là 13200 chênh lệch khoảng 35-36%. Con số này đã tốt hơn khá nhiều so với 50% ở trên, tuy nhiên ta vẫn kỳ vọng có thể làm tốt hơn do vậy ta sẽ thử với việc remove outlier qua cook's distance.

Đây là biểu đồ phân bố cook's distance của data 6 thuộc tính và data 4 thuộc tính

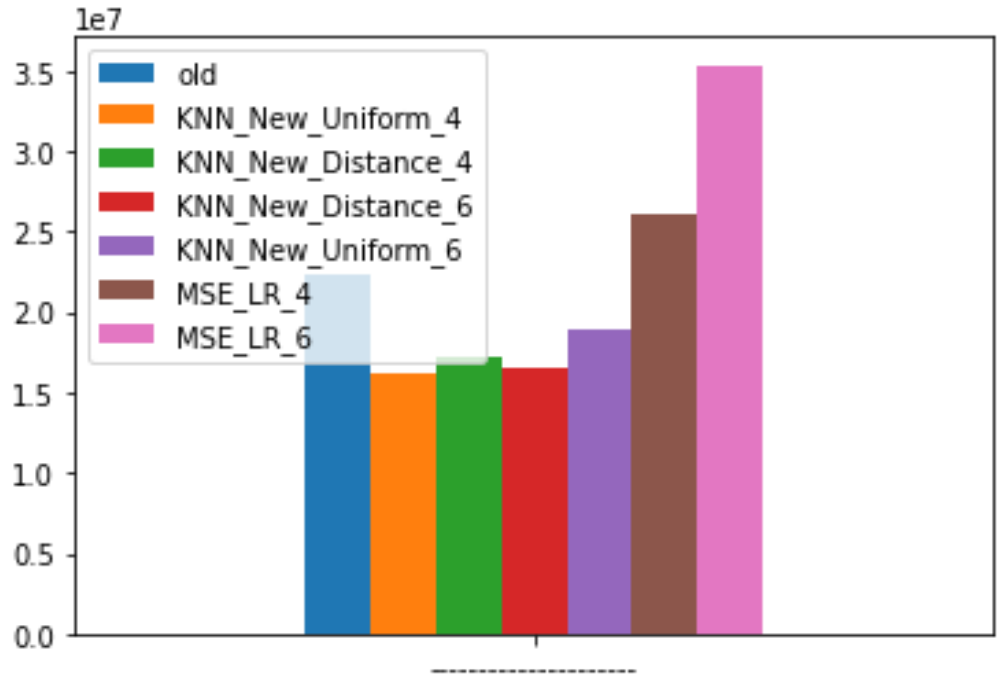




Ta chọn threshold 0.004 và remove tất cả các điểm dữ liệu có distance >0.004 thì remove khoảng 3% lượng dữ liệu, sau đó ta tiếp tục thử train dữ liệu như config ở trên.

Sau khi train, ta thu được MSE của LR_6 là 35346221.74168072 và LR_4 là 26101174.396164242 -> sau khi remove outlier LR_4 tốt lên khá nhiều trong khi LR_6 không đổi? Với KNN thì distance_6 và uniform_6 với K=5 là tốt nhất và MSE lần lượt là 16560297.35620579 và 18866170.713416085 còn KNN distance-4 và uniform_4 K=17 và MSE lần lượt là 17187063.662323218 và 16938245.525235213. Thấy KNN sau khi remove outlier model đã tốt lên. KNN với 4 features không tốt bằng 6 features. MSE tốt nhất hiện giờ là 16560297 và RMSE là khoảng 4000 chênh với MEAN khoảng 30%.

So sánh 6 model mới sau remove outlier với model tốt nhất trước khi remove outlier ta được bảng sau:



CHƯƠNG 3: TỔNG KẾT

Trên đây là một số mô hình cơ bản nên độ chênh lệch dự đoán với ground truth khá cao. Mô hình có độ chính xác lớn nhất hiện là KNN với 6 features tính weight bằng distance và sau khi remove outlier với độ lệnh so với trung bình là 30%. Từ các kết quả thu được cho thấy. Với mỗi kiến trúc mô hình khác nhau, cho ta một độ chính xác thu được khác nhau đáng kể. Việc chọn những features phù hợp hay việc loại bỏ những điểm ngoại lai cũng làm thay đổi độ chính xác đầu ra.

Qua quá trình thực hiện project, ta đã hiểu kỹ hơn về việc tiền xử lý, huấn luyện model cũng như các kỹ thuật cải tiến model và phân tích dữ liệu. Trên đây là tất cả những gì bọn em đã tìm hiểu trong quá trình làm đồ án môn học.

CHƯƠNG 4: TÀI LIỆU THAM KHẢO

[1] Các thư viện có sẵn như scikit-learn, ols, numpy, pandas, matplotlib, seaborn, statsmodels

[2] https://users.soict.hust.edu.vn/khoattq/ml-dm-course/

[3] Machine Learning cơ bản (machinelearningcoban.com)

[4] Bài viết mới nhất - Viblo