ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
📖📖📖

NHẬP MÔN HỌC MÁY

BÁO CÁO ĐỒ ÁN 01

REGRESSION

-- Giảng viên lý thuyết --

TS. Bùi Tiến Lên

-- Giảng viên hướng dẫn thực hành --

TS. Nguyễn Tiến Huy

TP. Hồ Chí Minh, tháng 5 năm 2021

MỤC LỤC

I- THÔNG TIN NHÓM VÀ ĐÁNH GIÁ ĐỒ ÁN

1. Thông tin thành viên, phân công công việc và đóng góp

Số thứ tự nhóm: **10**

STT	MSSV	Họ và tên	Phân công công việc	Đóng góp
1	18120066	Bùi Đoàn Hữu Nhân	- Cài đặt 3 thuật toán máy học: Polynomial Regression, Non-parametric Kernel Regression, Parametric Kernel Regression.	100%
2	18120085	Nguyễn Tấn Thìn	- Giới thiệu và phân tích 2 thuật toán máy học: Polynomial Regression, KNN Regression. - Khám phá và phân tích dữ liệu.	100%
3	18120090	Phạm Nguyên Minh Thy	- Giới thiệu và phân tích 2 thuật toán máy học: Non-parametric Kernel Regression, Parametric Kernel Regression. - Báo cáo kết quả đạt được và nhận xét.	100%
4	18120097	Đinh Hữu Phúc Trung	- Giới thiệu và phân tích 2 thuật toán máy học: Linear Regression, Ridge Regression. - Tiền xử lý dữ liệu, hoàn thành báo cáo và tập tin nộp bài.	100%
5	18120649	Nguyễn Phạm Phúc Việt	- Cài đặt 3 thuật toán máy học: Linear Regression, Ridge Regression, KNN Regression.	100%

2. Đánh giá mức độ hoàn thành đồ án

STT	Yêu cầu	Mức độ hoàn thành
1	Đọc và phân tích các đặc trưng trong 2 tập tin được cung cấp. Trình bày thông tin hữu ích (insights) tác động đến chi phí y tế cá nhân	100%
2	Cài đặt các thuật toán máy học đã được học để dự đoán chi phí y tế cá nhân.	100%
3	Báo cáo kết quả đạt được sau quá trình phân tích và cài đặt. Từ đó nhận xét về các tác nhân ảnh hưởng mạng/yếu tới chi phí y tế cá nhân.	100%
Tổng		**100%**

II- NỘI DUNG

1. Đọc và khám phá dữ liệu

Các đặc trưng:

- Age: tuổi
- Sex: Giới tính

- BMI: Chỉ số khối cơ thể
- Children: Số lượng trẻ con/người phụ thuộc
- Smoker: Tình trạng hút thuốc
- Region: Khu vực sinh sống
- Charges: Chi phí y tế cá nhân

Dữ liệu tập train.csv gồm 10003 dòng và 7 cột như trên.

Dữ liệu tập test.csv gồm 335 dòng và 7 cột như trên.

Kiểu dữ liệu các cột như sau:

Cột	Kiểu dữ liệu
age	int64
sex	object
bmi	float64
children	int64
smoker	object
region	object
charges	float64

Các cột có kiểu dữ liệu số: age, bmt, children.

Các cột có kiểu dữ liệu object: sex, smoker, region.

Quan sát các cột có kiểu dữ liệu số:

	age	bmi	children
missing_ratio	0.0	0.00	0.0
min	18.0	15.96	0.0
lower_quartile	27.0	26.20	0.0
median	39.0	30.20	1.0
upper_quartile	51.0	34.40	2.0
max	64.0	53.13	5.0

Quan sát các cột có kiểu dữ liệu object:

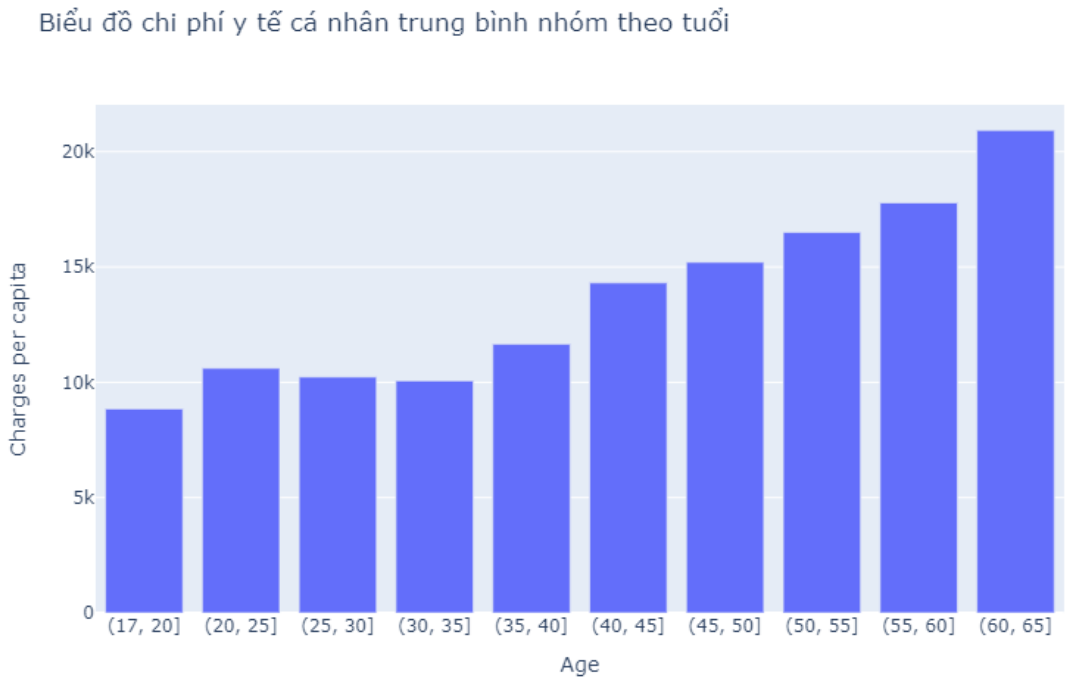
	sex	smoker	region
missing_ratio	0	0	0
num_values	2	2	4
value_ratios	{'male': 50.4, 'female': 49.6}	{'no': 79.5, 'yes': 20.5}	{'southeast': 25.8, 'northeast': 25.0, 'southwest': 24.6, 'northwest': 24.5}

2. Phân tích đặc trưng và tiền xử lý dữ liệu

2.1. Phân tích các đặc trưng

Các thông tin hữu ích (insights) tác động đến chi phí y tế cá nhân:

- **Age**

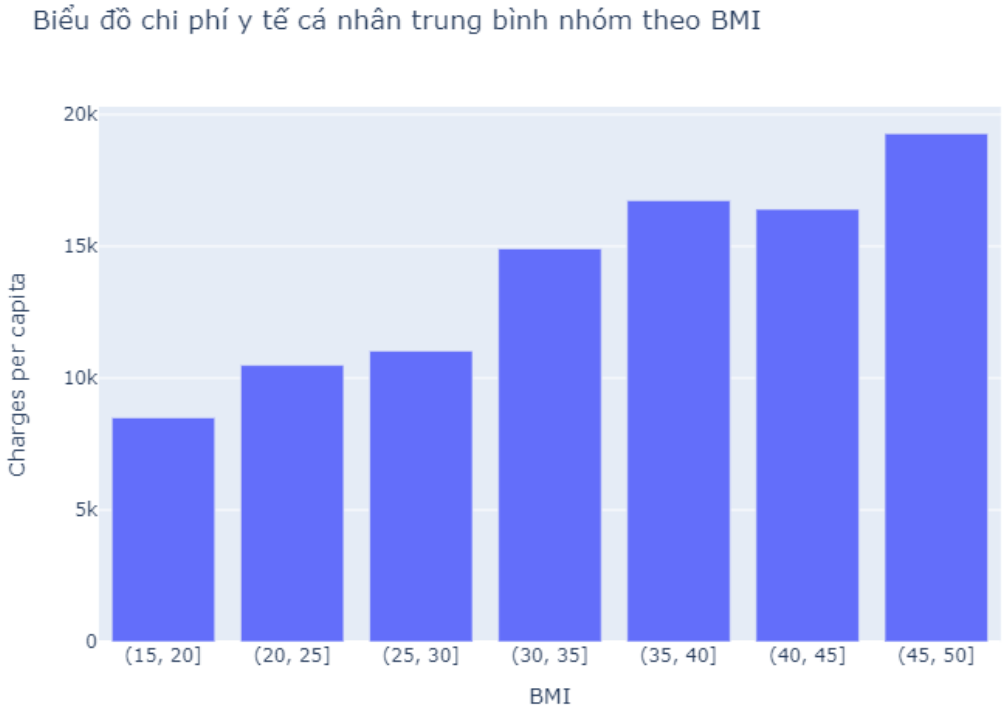


Nhận xét: Trong các nhóm tuổi từ 18 đến 65, chi phí y tế cá nhân trung bình có tỉ lệ thuận với nhóm tuổi, nghĩa là nhóm tuổi càng cao thì chi phí càng lớn, từ đó cho thấy được tuổi (age) có tác động tương đối đến chi phí y tế cá nhân.

- **Sex**

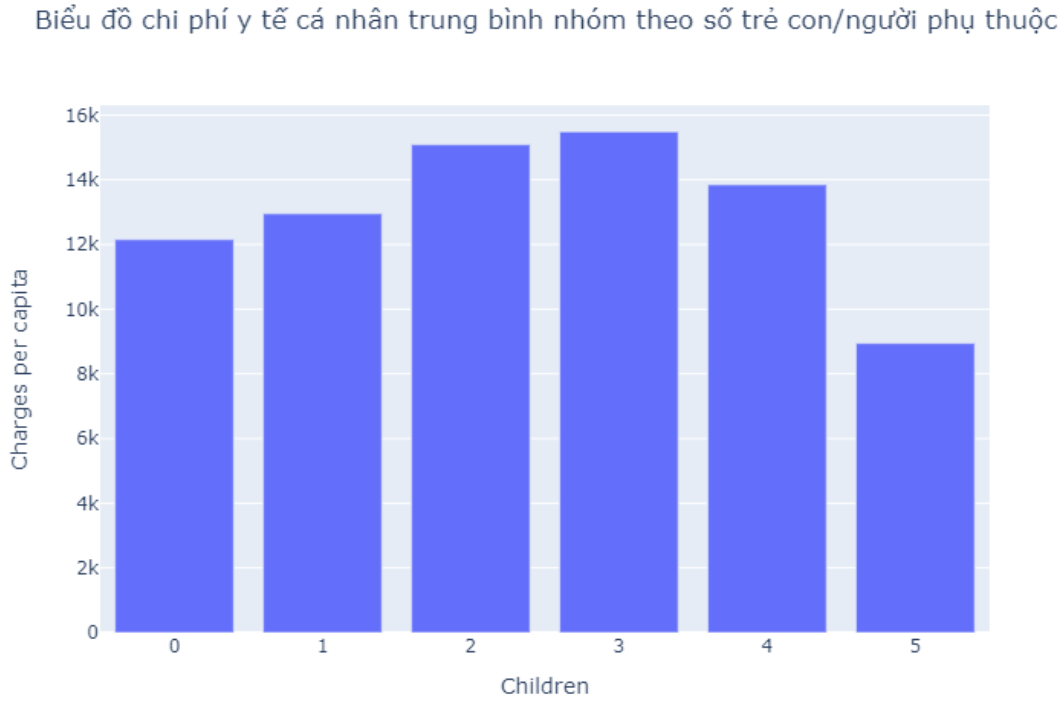
Nhận xét: Chi phí y tế bình quân trên đầu người của nam và nữ có chênh lệch không đáng kể → giới tính (age) có ảnh hưởng rất yếu tới chi phí y tế cá nhân.

- **BMI**



Nhận xét: Trong các nhóm BMI từ 15 đến 65, chi phí y tế cá nhân trung bình có tỉ lệ thuận với BMI, nghĩa là nhóm BMI càng cao thì phi phí càng lớn → chỉ số BMI có ảnh hưởng tương đối tới chi phí y tế cá nhân.

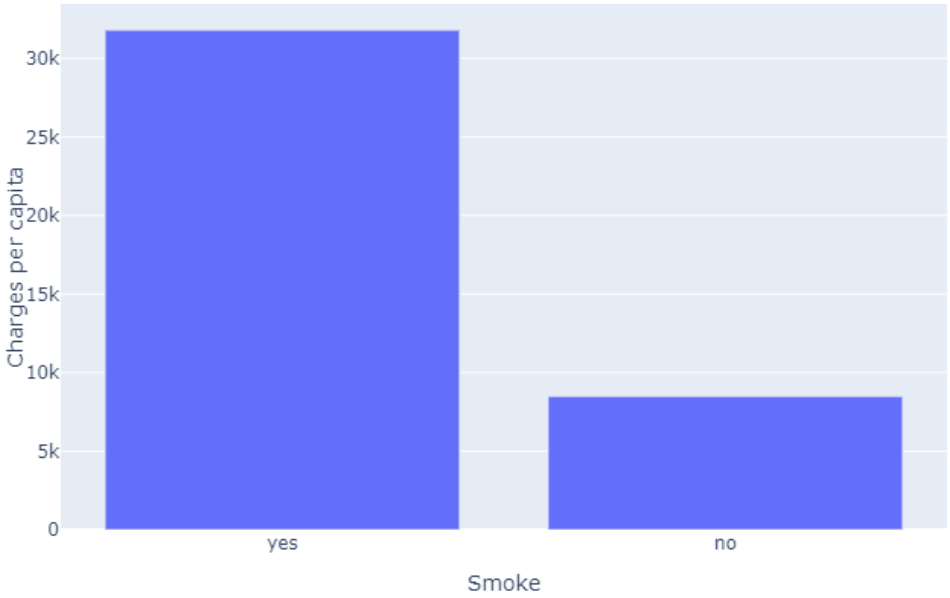
- **Children**



Nhận xét: Qua biểu đồ, phần nào cho ta thấy được việc có nhiều hay ít trẻ con/người phụ thuộc không có ảnh hưởng đến chi phí y tế cá nhân của người đó quá nhiều.

- **Smoker**

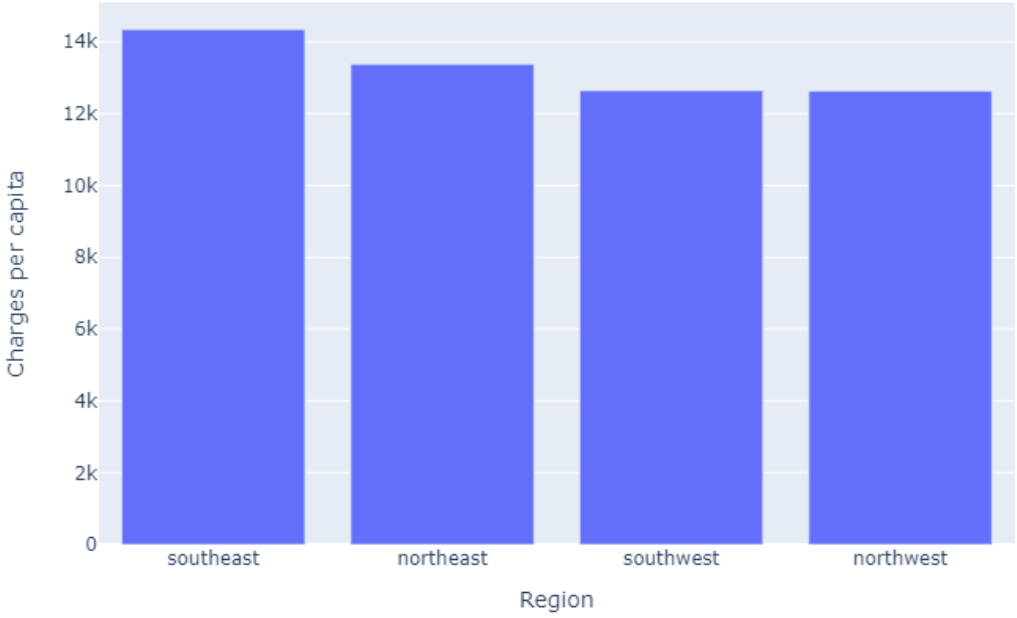
Biểu đồ chi phí y tế cá nhân trung bình nhóm theo tình trạng hút thuốc



Nhận xét: Chi phí y tế cá nhân trung bình của người có hút thuốc lớn hơn nhiều lần so với người không hút thuốc → tình trạng hút thuốc có ảnh hưởng mạnh đến việc dự đoán chi phí y tế cá nhân.

- **Region**

Biểu đồ chi phí y tế cá nhân trung bình nhóm theo khu vực



Nhận xét: Chi phí y tế cá nhân trung bình của mỗi khu vực có chênh lệch không đáng kể → khu vực (region) có ảnh hưởng khá yếu tới chi phí y tế cá nhân.

2.2. Tiền xử lý dữ liệu

Tiến hành phân tích thông số tương quan ảnh hưởng của các đặc trưng đối với cột `Charges` sau khi đã Encode các cột có dữ liệu object, thu được các chỉ số tương quan như sau:

```
age         0.293434
bmi         0.195828
children    0.079823
sex         0.043353
smoker      0.781981
region     -0.004034
dtype: float64
```

Dựa vào việc phân tích ở mục **2.1** và các chỉ số tương quan ở trên, ta nhận thấy được 3 cột `children`, `sex`, `region` có độ tương quan ảnh hưởng thấp đến `charges` nên ta sẽ bỏ đi 3 cột này và giữ lại các cột `age`, `bmi`, `smoker` để chạy các mô hình máy học dự đoán chi phí y tế cá nhân.

Sử dụng 1 pipeline để tiền xử lý dữ liệu:

- Encode các cột có kiểu dữ liệu object.
- Drop 3 cột `children`, `sex`, `region` vì có độ tương quan ảnh hưởng thấp.
- Biến đổi dữ liệu sao cho phân phối có giá trị trung bình là 0 và độ lệch chuẩn là 1 qua lớp StandardScaler.

3. Giới thiệu và phân tích các thuật toán máy học đã cài đặt

Các thuật toán Regression đạt được hiệu quả cao thì dữ liệu cần được phân bố một cách tuyến tính và ít đối tượng nhiễu.

3.1. Linear Regression

Linear Regression là một trong những thuật toán cơ bản nhất của Machine Learning. Đôi khi được gọi là Linear Fitting hoặc Linear Least Square.

Yêu cầu bài toán: nhận input là 1 vector $x \in \mathbb{R}^{D+1}$ sau đó dự đoán kết quả là 1 giá trị $y \in \mathbb{R}$. Có công thức tính giá trị dự đoán từ model - $\hat{y}$:

$$y \approx \hat{y} = w^T x$$

Với $w \in \mathbb{R}^{D+1}$ là 1 vector tham số của model.

Bài toán với N mẫu:

Có tập train – (X, y) gồm N mẫu $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ mỗi x có D thuộc tính.

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1D} \\ 1 & x_{21} & \cdots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{ND} \end{pmatrix} \begin{pmatrix} \mathrm{w}_0 \\ \mathrm{w}_1 \\ \vdots \\ \mathrm{w}_D \end{pmatrix}$$

Có công thức tính giá trị dự đoán từ model:

$$\hat{y} \approx Xw$$

Hàm mất mát - Độ lỗi Mean Squared Error MSE:

$$L(\mathrm{w}) = MSE_{train} = \frac{1}{N}\|\hat{y} - y\|^2 = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

Cần tìm giá trị w để hàm mát đạt giá trị nhỏ nhất:

$$\mathrm{w}^* = \arg\min_{\mathrm{w}}(L(\mathrm{w}))$$

Tiến hành đạo hàm mất mát theo w:

$$\frac{\partial L(\mathrm{w})}{\partial \mathrm{w}} = 2X^T(X\mathrm{w} - y)$$

$$\frac{\partial L(\mathrm{w})}{\partial \mathrm{w}} = 0 \rightarrow \mathrm{w} = (X^T X)^{-1} X^T y$$

Tìm được tham số w, ta tính được $\hat{y}$.

3.2. Ridge Regression

Ridge Regression là 1 dạng hồi quy tuyến tính giống Linear Regression nhưng khác là dùng L2 regularization.

Regularization nói chung là quá trình để tránh overfitting khi sử dụng model quá phức tạp. Dùng regularization giúp giảm độ phức tạp của model bằng cách thêm 1 complexity term để nó làm độ lỗi lớn lên khi dùng 1 model quá phức tạp.

Input và output của bài toán giống với Linear Regression với:

- Input: X
- Output: y
- tham số của model cần tìm: w

Hàm mất mát:

Hàm mất mát của Ridge Regression là sự biến đổi từ MSE khi kết hợp MSE và thêm 1 complexity term để penalize kích thước của tham số cần tìm

$$L_{ridge}(\mathrm{w}) = \|y - X\mathrm{w}\|^2 + \lambda\|\mathrm{w}\|^2$$

λ là regularization penalty. Cách chọn λ sẽ ảnh hưởng tới mức độ penalize model:

- Nếu $\lambda \to 0$: $w_{ridge} \to w_{OLS}$. Với OLS là Ordinary Least Square.
- Nếu $\lambda \to \infty$: $w_{ridge} \to 0$.

Vậy λ càng lớn thì càng giảm độ phức tạp của model.

Cần tìm giá trị *w* để hàm mát đạt giá trị nhỏ nhất:

$$w^* = \arg\min_w (L_{ridge}(w))$$

Tiến hành đạo hàm mất mát theo *w*:

$$w = (X^T X + \lambda I)^{-1}(X^T y)$$

Với I là identity matrix

Tìm được tham số *w*, ta tính được $\hat{y}$.

3.3. KNN Regression

K-Nearest Neighbor (KNN) là một thuật toán học máy phi tham số. Khác với các thuật toán học máy thông thường, kNN không loại bỏ bộ dữ liệu training sau khi mô hình đã được xây dựng, thay vào đó kNN lưu giữ toàn bộ tập dữ liệu trong bộ nhớ. Khi training, thuật toán này không học gì từ dữ liệu training (lazy learning). KNN còn được gọi là một thuật toán Instance-based hay Memory-based learning.

Trong bài toán Regresssion, đầu ra của một điểm dữ liệu sẽ bằng đầu ra của điểm dữ liệu đã biết gần nhất (nếu k=1), hoặc là trung bình có trọng số của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên khoảng cách tới các điểm đó. Tóm lại, kNN là thuật toán tìm đầu ra của một điểm dữ liệu mới bằng cách dựa trên thông tin của k điểm dữ liệu gần nó nhất (k-lân cận).

Khoảng cách của 2 điểm trong bài toán KNN được cho bởi một hàm khoảng cách. Hàm khoảng cách thường được sử dụng nhất là hàm khoảng cách Euclid:

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

Một lựa chọn phổ biến khác nữa là hàm tương tự cosin âm (negative cosine similarity), với hàm tương tự cosine được định nghĩa:

$$s(x_i, x_k) \overset{\text{def}}{=} \cos(\angle(x_i, x_k)) = \frac{\sum_{j=1}^{D} x_i^{(j)} x_k^{(j)}}{\sqrt{\sum_{j=1}^{D} (x_i^{(j)})^2} \sqrt{\sum_{j=1}^{D} (x_k^{(j)})^2}}$$

Hàm tương tự cosine thể hiện sự tương đồng về hướng giữa 2 vector. Để sử dụng hàm tương tự cosine cho khoảng cách giữa 2 điểm, ta đơn giản nhân nó với -1. Một số hàm khoảng cách tương đối phổ biến khác là Chebychev, Mahalanobis và Hamming.

3.4. Polynomial Regression

Mô hình Polynomial Regression là một nâng cấp của mô hình Linear Regression khi sử dụng hàm đa thức thay vì một hàm số tuyến tính.

Với những dữ liệu có xu hướng hơi hướng theo dạng đường "cong", thì hàm mô hình Linear Regression cho dù có cố gắng fit cũng không còn chính xác nữa. Mô hình Linear Regression không còn đáp ứng được sự biến thiên của dữ liệu nữa (underfitting). Lúc này, cái ta cần là mô hình của ta có thể uốn theo dạng đường cong của dữ liệu, vì vậy ta nâng hàm tuyến tính sang thành hàm đa thức. Dù đa thức có dạng bậc cao hơn thì phương pháp tối tiểu hóa $J(\theta)$ không thay đổi.

Khi ta cố gắng fit mô hình với dữ liệu càng lớn khi bậc càng cao, thậm chí mô hình có độ chính xác là 100%, nhưng sẽ có một vấn đề khác sẽ xảy ra đó là "overfitting". Vì vậy, khi triển khai Polynomial Regression, ta cần lựa chọn bậc đa thức cho phù hợp, tránh việc underfitting và overfitting.

Đa thức tổng quát:

$$y_i = \mathrm{w}_0 x_i^{\,0} + \mathrm{w}_1 x_i^{\,1} + \mathrm{w}_2 x_i^{\,2} + \cdots + \mathrm{w}_m x_i^{\,m} + \varepsilon_i$$

$$\vec{y} = X\vec{\mathrm{w}} + \vec{\varepsilon}$$

Input: Dữ liệu train gồm N mẫu $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$.

Output: Tìm vector tham số w của model để dự đoán y.

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1^1 & \cdots & x_1^m \\ 1 & x_2^1 & \cdots & x_2^m \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n^1 & \cdots & x_n^m \end{pmatrix} \begin{pmatrix} \mathrm{w}_0 \\ \mathrm{w}_1 \\ \vdots \\ \mathrm{w}_m \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

Hàm mất mát - Độ lỗi Mean Squared Error MSE:

$$L(\mathrm{w}) = MSE_{train} = \frac{1}{N}\|\hat{y} - y\|^2 = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

Cần tìm giá trị w để hàm mát đạt giá trị nhỏ nhất:

$$\mathrm{w}^* = \arg\min_{\mathrm{w}}(L(\mathrm{w}))$$

Tiến hành đạo hàm mất mát theo w:

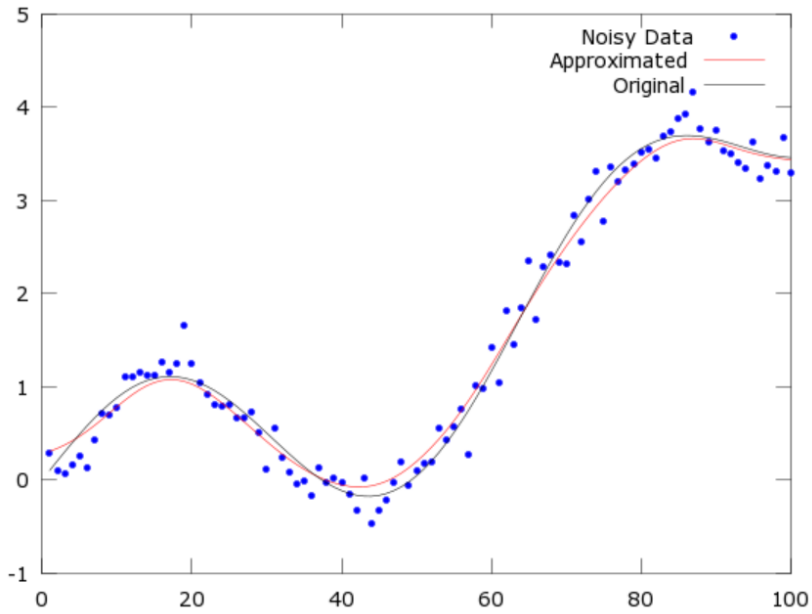
$$\frac{\partial L(\mathrm{w})}{\partial \mathrm{w}} = 2X^T(X\mathrm{w} - y)$$

$$\frac{\partial L(\mathrm{w})}{\partial \mathrm{w}} = 0 \rightarrow \mathrm{w} = (X^T X)^{-1} X^T y$$

3.5. Non-parametric Kernel Regression

Non-parametric Kernel Regression còn gọi là Gaussian Kernel Regression – GKR. Đây là 1 kĩ thuật hồi quy mà không dùng kĩ thuật lặp – iterative learning như gradient descent trong Linear Regression.

Gaussian Kernel Regression là 1 kĩ thuật cho Non-linear Regression.



Với dùng những điểm dữ liệu được phân bố như hình, GKR có thể xấp xỉ đường màu đỏ gần với hàm thực tế là đường màu xanh.

Input: Dữ liệu train gồm N mẫu $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$.

Bài toán: Dự đoán $\hat{y}$ với một x mới.

Công thức:

$$\hat{y} = \sum_{i=1}^{n} \alpha(x, x_i) y_i$$

Xác định α sử dụng Gaussian kernel:

$$\alpha(x, x_i) = \frac{e^{-\frac{1}{2}(x-x_i)^2}}{\sum_{j=1}^{n} e^{-\frac{1}{2}(x-x_j)^2}}$$

Vậy thu được công thức:

$$\hat{y} = \sum_{i=1}^{n} \alpha(x, x_i) y_i = \sum_{i=1}^{n} \frac{e^{-\frac{1}{2}(x-x_i)^2}}{\sum_{j=1}^{n} e^{-\frac{1}{2}(x-x_j)^2}} y_i$$

3.6. Parametric Kernel Regression

Parametric Kernel Regression tương tự như Non-parametric Kernel Regression.

Phương pháp này kết hợp thêm tham số học model – w.

Nhận thấy rằng khi thêm tham số học model – w vào thì kết quả cho ra được tốt hơn Non-parametric Kernel Regression.

Input: Dữ liệu train gồm N mẫu $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$.

Bài toán: Dự đoán $\hat{y}$ với một x mới.

Công thức:

$$\hat{y} = \sum_{i=1}^{n} \frac{e^{-\frac{1}{2}((x-x_i)w)^2}}{\sum_{j=1}^{n} e^{-\frac{1}{2}((x-x_j)w)^2}} y_i$$

4. Báo cáo kết quả và nhận xét

4.1. Báo cáo kết quả

Đánh giá các mô hình trên bằng độ lỗi R^2

Công thức:

Giá trị trung bình của dữ liệu quan sát

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

Total sum of squares

$$SS_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Residual sum of squares

$$SS_{res} = \sum_{i=1}^{n} (y_i - y_i)^2$$

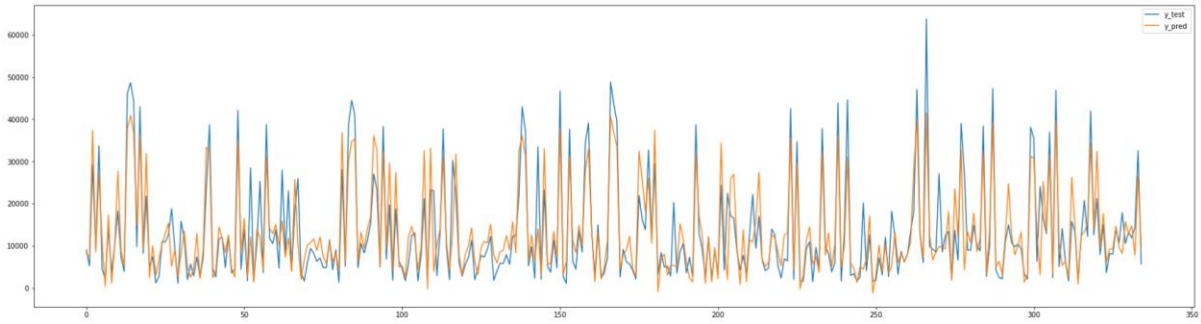
Độ lỗi R^2

$$R^2 = 1 - \frac{SS_{tot}}{SS_{res}}$$

- **Thuật toán Linear Regression**

  Độ chính xác trên tập train: 0.7423

  Độ chính xác trên tập test: 0.7623

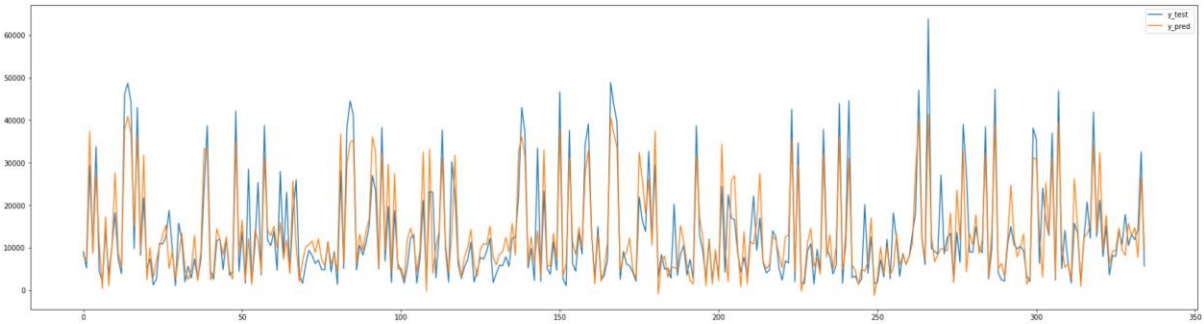


Biểu đồ so sánh giữa y_test và y_pred

- **Thuật toán Ridge Regression**

  Độ chính xác trên tập train: 0.7423

  Độ chính xác trên tập test: 0.7623

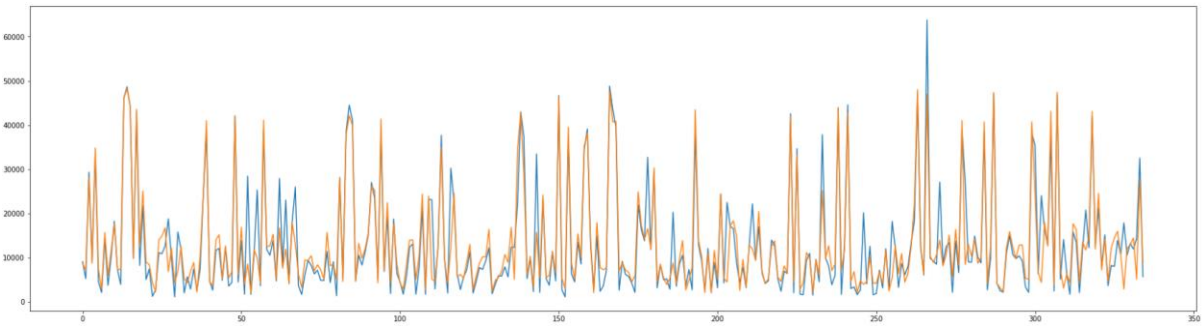


Biểu đồ so sánh giữa y_test và y_pred

- **Thuật toán KNN Regression**

  Độ chính xác trên tập train: 0.8643

  Độ chính xác trên tập test: 0. 8598

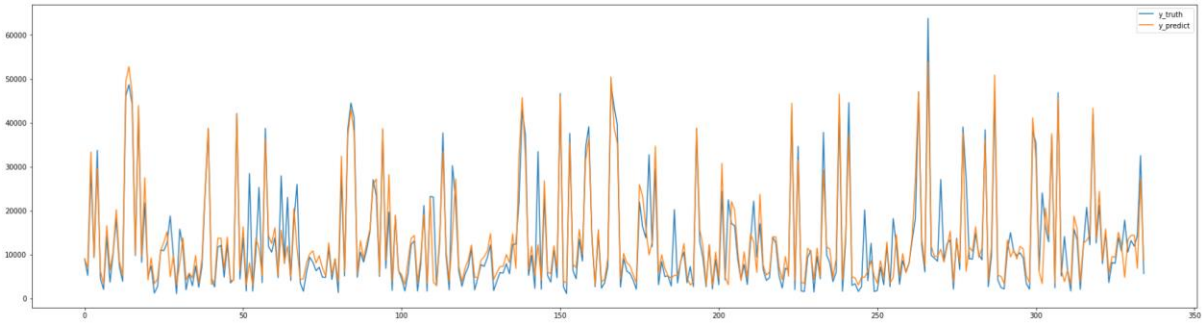


Biểu đồ so sánh giữa y_test và y_pred

- **Thuật toán Polynomial Regression**

  Độ chính xác trên tập train: 0.8377

  Độ chính xác trên tập test: 0.8512

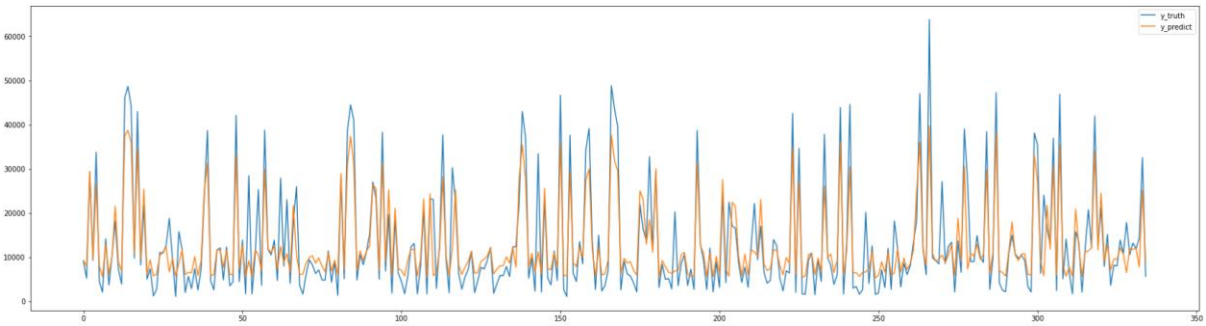


Biểu đồ so sánh giữa y_test và y_pred

- **Thuật toán Non-parametric Kernel Regression**

  Độ chính xác trên tập train: 0.7695

  Độ chính xác trên tập test: 0.7903

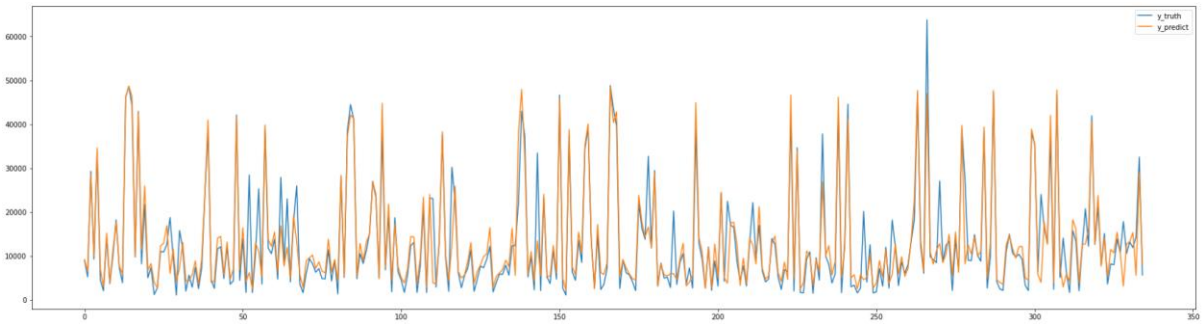


Biểu đồ so sánh giữa y_test và y_pred

- **Thuật toán Parametric Kernel Regression**

  Độ chính xác trên tập train: 0.8780

  Độ chính xác trên tập test: 0.8609



Biểu đồ so sánh giữa y_test và y_pred

4.2. Nhận xét

- **Các thuật toán máy học:**

Nhận thấy 2 thuật toán thuần về Hồi quy tuyến tính là Linear Regression và Ridge Regression có độ chính xác thấp hơn các thuật toán còn lại.

Linear Regression và Ridge Regression có độ chính xác bằng nhau do bản chất 2 thuật toán này giống nhau. Ridge Regression áp dụng thêm L2 regularization nhưng ở đây không có ý nghĩa nhiều lắm do mô hình bài toán này khá đơn giản.

Đối với mô hình KNN Regression, thuật toán chọn số lượng neighbors = 10 và trả ra kết quả khá tốt.

3 mô hình Polynomial Regression, Non-parametric Regression và Parametric Regression là các mô hình phi tuyến tính. Áp dụng 3 mô hình phức tạp hơn giúp bài toán trả về kết quả tốt hơn mô hình tuyến tính.

- **Các tác nhân ảnh hưởng:**

Qua quá trình phân tích, nhận thấy trong các tác nhân ảnh hưởng đến chi phí y tế cá nhân thuộc 2 tập tin được cung cấp, **smoker** có ảnh hướng mạnh nhất tới chi phí y tế cá nhân (người có hút thuốc có chi phí cao hơn nhiều so với người không hút thuốc); **children**, **sex**, **region** có ảnh hưởng yếu tới chi phí y tế cá nhân và các tác nhân còn lại có ảnh hướng tương đối.

III- TÀI LIỆU THAM KHẢO

[1]. (2019). In A. Burkov, The Hundred-Page Machine Learning Book (pp. 41-42). Quebec City, Canada.

[2]. Kernel Regression · Chris McCormick (mccormickml.com)

[3]. Ridge Regression Definition & Examples | What is Ridge Regression? (mygreatlearning.com)

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

[5]. Polynomial regression using scikit-learn (opengenus.org)

[6]. scikit-learn/_data.py at 15a949460dbf19e5e196b8ef48f9712b72a3b3c3 · scikit-learn/scikit-learn · GitHub

[7]. Kernel Regression from Scratch | Kunj Mehta | Towards Data Science