ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN



MACHINE LEARNING

LAB 1: REGRESSION

*Lớp*: Nhập môn học máy 21

*Nhóm thực hiện*:

18120078 – Ngô Phù Hữu Đại Sơn

18120253 – Mai Ngọc Tú

18120201 ­– Nguyễn Bảo Long

MỤC LỤC

[A. Thông tin khái quát 2](#_Toc71921326)

[I. Thông tin nhóm 2](#_Toc71921327)

[II. Bảng phân công công việc 2](#_Toc71921328)

[B. Nội dung 3](#_Toc71921329)

[I. Mục tiêu của đồ án: 3](#_Toc71921330)

[II. Yêu cầu: 3](#_Toc71921331)

[III. Triển khai 4](#_Toc71921332)

[1. Phân tích đặc trưng dữ liệu được cung cấp 4](#_Toc71921333)

[2. Mô hình học máy 9](#_Toc71921334)

[3. Báo cáo kết quả và nhận xét 14](#_Toc71921335)

[IV. Đánh giá đồ án 15](#_Toc71921336)

[VII. Nguồn tham khảo 15](#_Toc71921337)

# Thông tin khái quát

## I. Thông tin nhóm

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Họ tên** | **Vai trò** |
| 18120078 | Ngô Phù Hữu Đại Sơn | Nhóm trưởng |
| 18120253 | Mai Ngọc Tú | Thành viên |
| 18120201 | Nguyễn Bảo Long | Thành viên |

## II. Bảng phân công công việc

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Công việc phụ trách** | **Thời gian thực hiện** |
| 18120078 |  |  |
| 18120253 |  |  |
| 18120201 |  |  |

# B. Nội dung

## I. Mục tiêu của đồ án:

Tìm hiểu và vận dụng các kỹ thuật:

* **Nhận diện bài toán:** Sinh viên cần tìm hiểu bài toán và dữ liệu được giao nhằm xác định nội dung và ý nghĩa bài toán thực tế cần giải quyết. Thông qua đó, sinh viên có khả năng ánh xạ vấn đề thực tế sang bài toán lập trình.
* **Giải quyết vấn đề:** Sinh viên được yêu cầu đưa ra các giải pháp và hướng tiếp cận nhằm giải quyết được yêu cầu bài toán thực tế.
* **Xử lý và phân tích dữ liệu**: Sinh viên có khả năng xử lý các công cụ phântích dữ liệu tự động nhằm tìm ra các thông tin hữu ích, các đặc trưng tiềm ẩn ảnh hưởng để mục tiêu bài toán.
* **Thiết kế và cài đặt các thuật toán máy học:** Sinh viên được yêu cầu có khả năng đề xuất, triển khai và giải thích các thuật toán máy học nhằm giải quyết bài toán được giao.

## II. Yêu cầu:

1. Đọc và phân tích các đặc trưng trong 2 tập tin được cung cấp. Trình bày các thông tin hữu ích (insights) tác động đến chi phí y tế cá nhân.
2. Cài đặt các thuật toán máy học đã được học để dự đoán chi phí y tế cá nhân.
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 mạnh/yếu tới chi phí y tế cá nhân.

## III. Triển khai

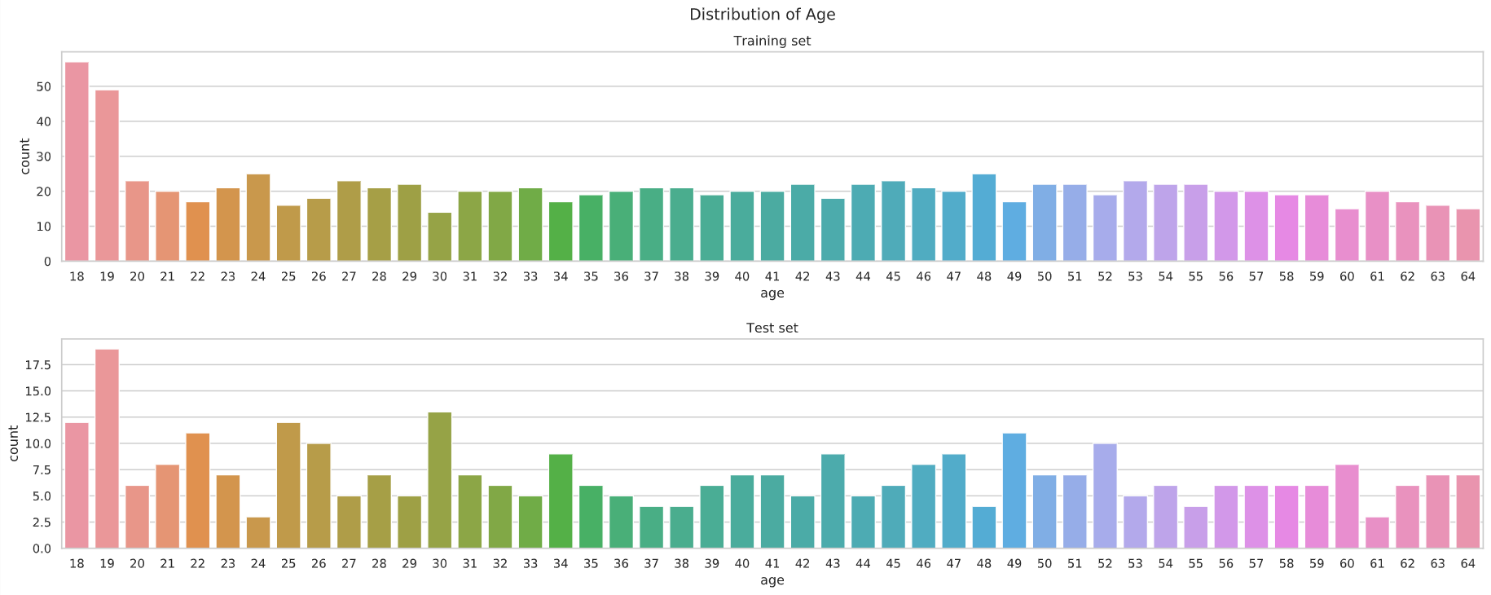
### Phân tích đặc trưng dữ liệu được cung cấp

* Thuộc tính input: Age, Sex, BMI, Region, Children, Smoker
* Thuộc tính output: Charges

1. **Phân tích phân bố từng thuộc tính**

* **Age** (độ tuổi):

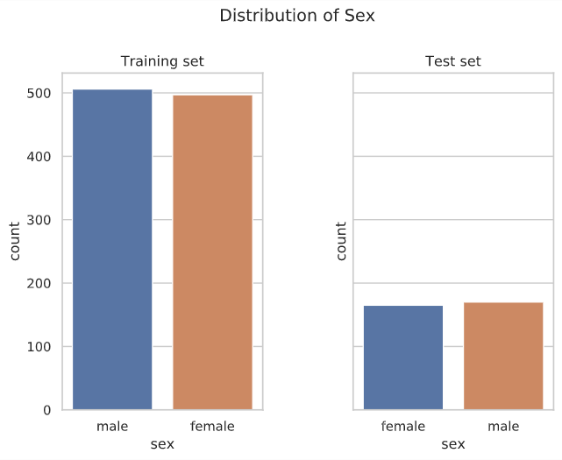
+ Biểu diễn phân bố dữ liệu:



+ Nhận xét: Dữ liệu về độ tuổi các đối tượng được cho phân bố khá đều. Riêng nhóm tuổi dưới 20 có số lượng nhiều hơn các nhóm còn lại

* **Sex** (giới tính):

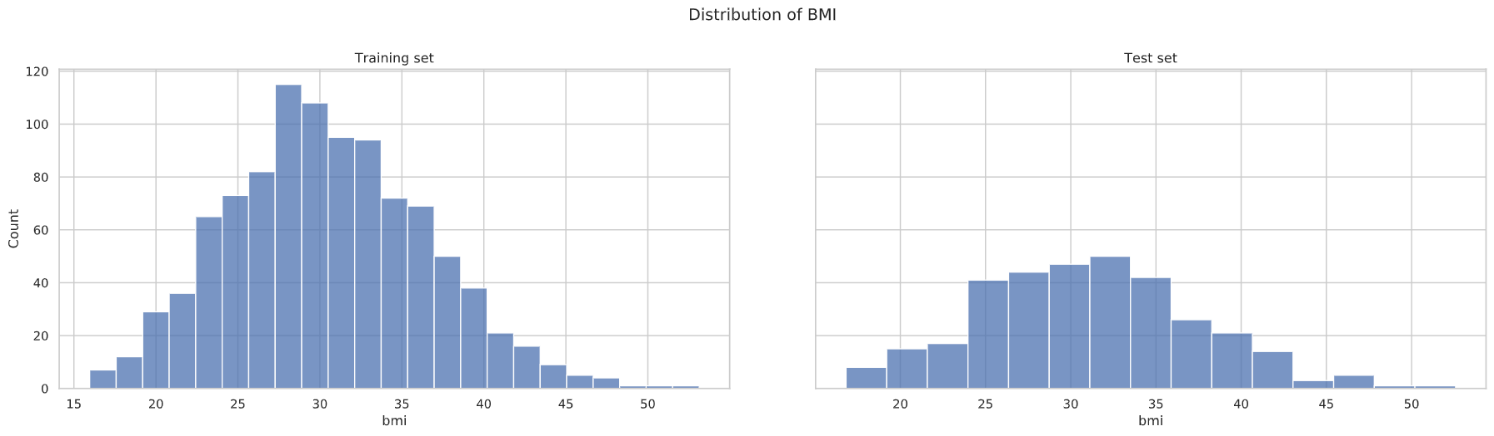
+ Biểu diễn phân bố dữ liệu:



+ Nhận xét: dữ liệu về giới tính của các đối tượng được cho có 2 giá trị và phân bố đều nhau (xấp xỉ 1:1)

* **BMI** (chỉ số khối cơ thể):

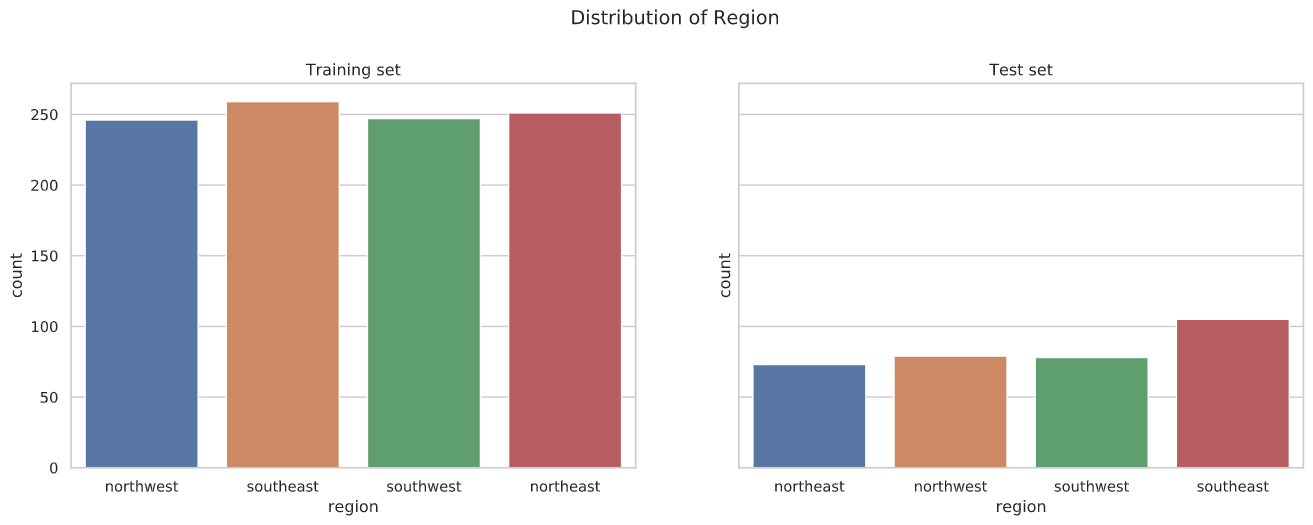
+ Biểu diễn phân bố dữ liệu:



+ Nhận xét: dữ liệu về chỉ số BMI của các đối tượng phân bố theo hình tháp chuông, có xu hướng hơi lệch về bên trái

* **Region** (vùng miền):

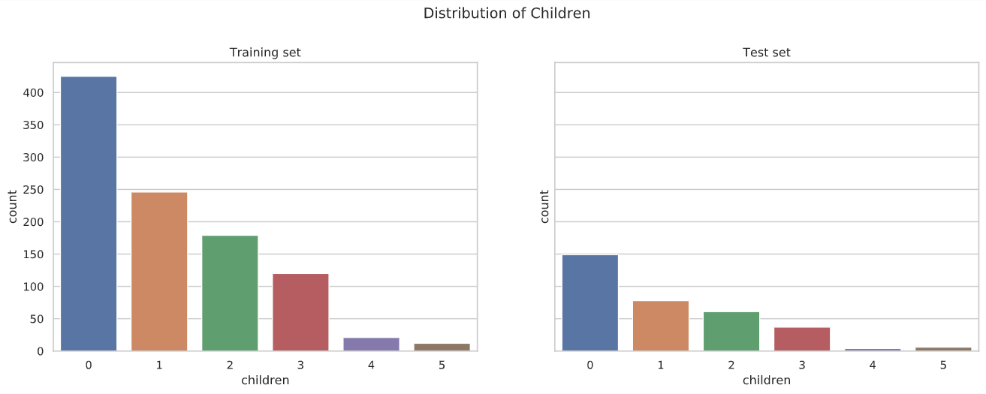
+ Biểu diễn phân bố dữ liệu:



+ Nhận xét: phân bố dữ liệu vùng miền của các đối tượng gồm 4 giá trị và phân bố khá đồng đều nhau.

* **Children** (số lượng con cái):

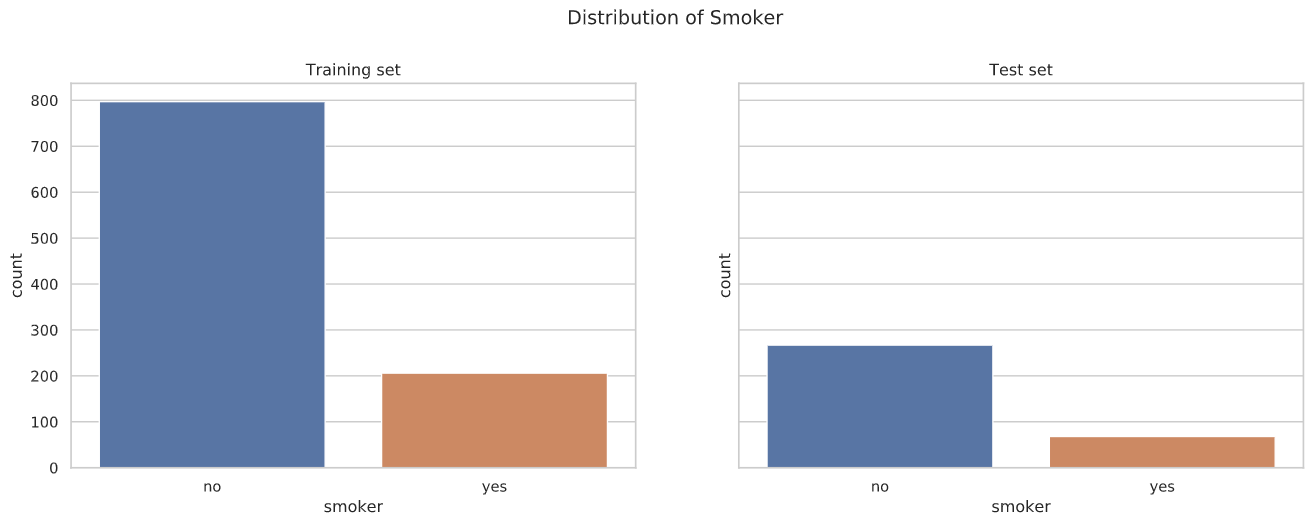
+ Biểu diễn phân bố dữ liệu:



+ Nhận xét: Phân bố về dữ liệu số lượng con cái (phụ thuộc) vào đối tượng có xu hướng giảm dần khi số lượng con tăng . Hầu hết các đối tượng có số lượng con nhỏ hơn 4

* **Smoker** (hút thuốc):

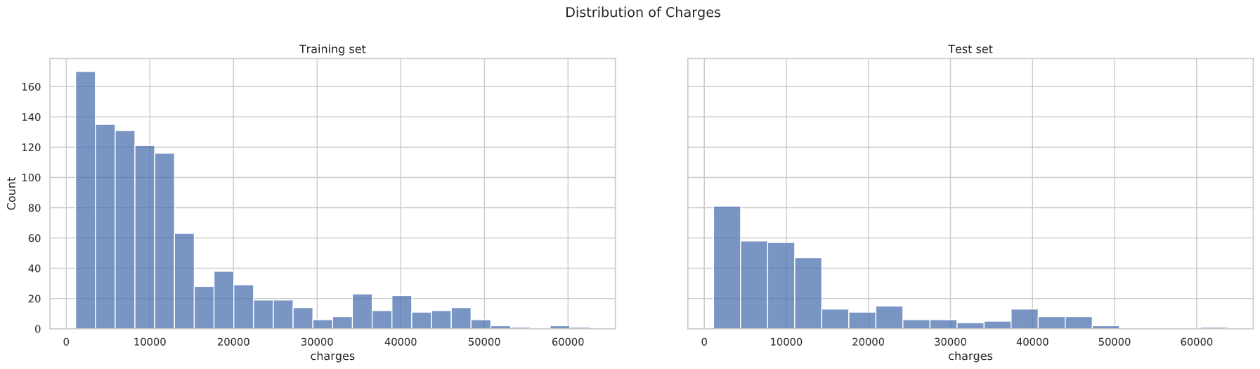
+ Biểu diễn phân bố dữ liệu:



+ Nhận xét: Số lượng đối tượng không hút thuốc nhiều hơn số lượng đối tượng có hút thuốc rất nhiều (gấp 4 lần)

* **Charge** (chi phí y tế):

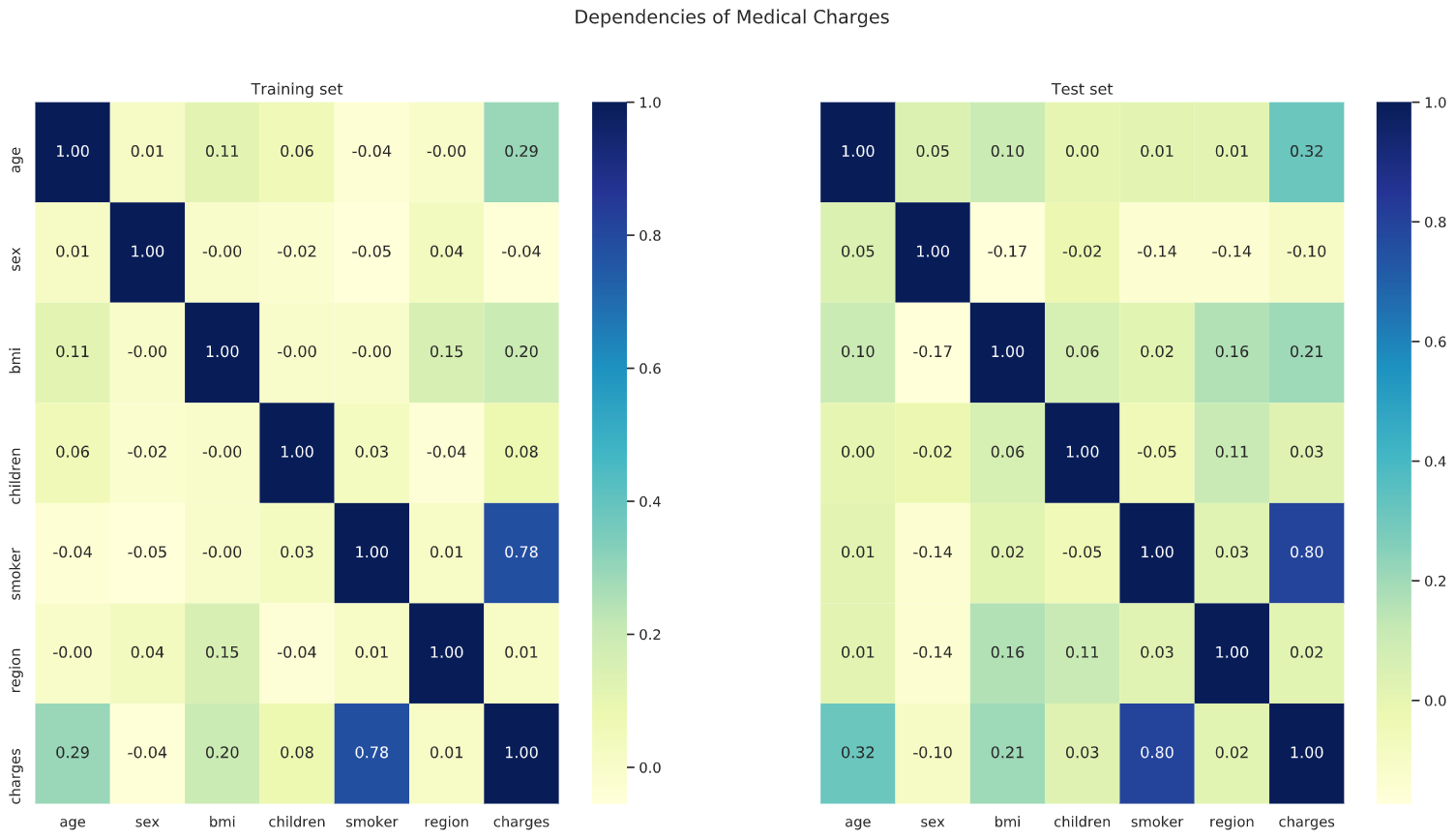
**+** Biểu diễn phân bố dữ liệu:

****

+ Nhận xét: Hầu hết các đối tượng có chi phí y tế nhỏ hơn 15000. Tuy nhiên ta không thể xem phần còn lại là ngoại lệ vì chúng có thể bị ảnh hưởng bởi độ lệch của các thuộc tính trước đó như: Smoker và Children

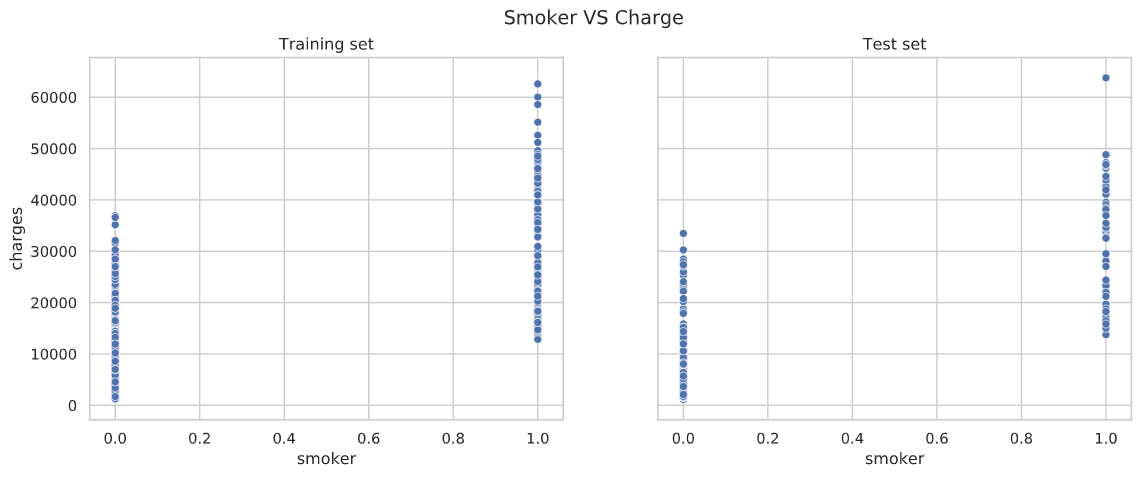
1. **Thông tin tác động đến thuộc tính “Charges”**

* Ma trận tương quan giữa các thuộc tính:



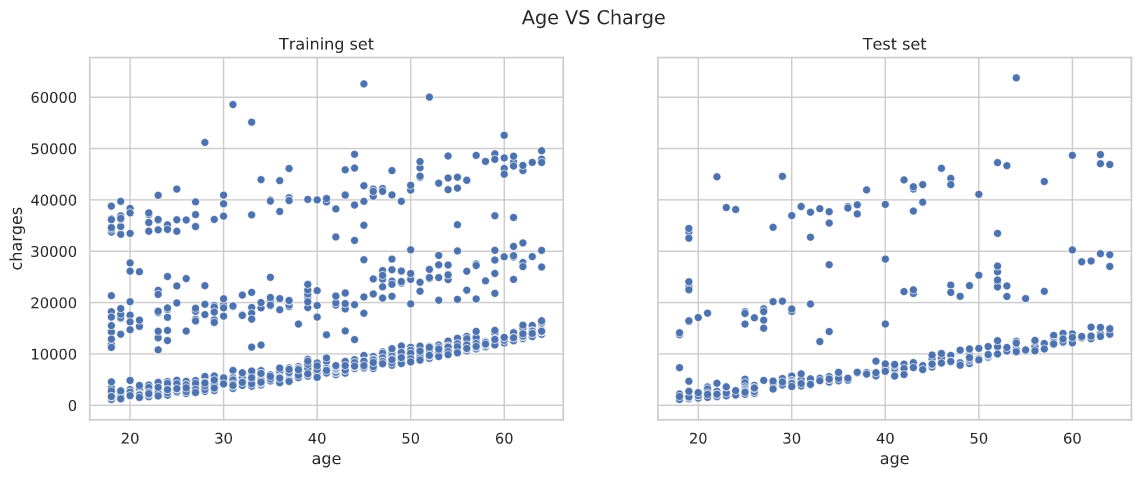
+ Nhận xét:

* Các thuộc tính có tương quan cao với thuộc tính Charges: Age (0.29), BMI (0.20) và đặc biệt là Smoker (0.78)
* Các thuộc tính còn lại rất ít tương quan với Charges: Sex, Children, Region
* Tương quan giữa Smoker và Charges:



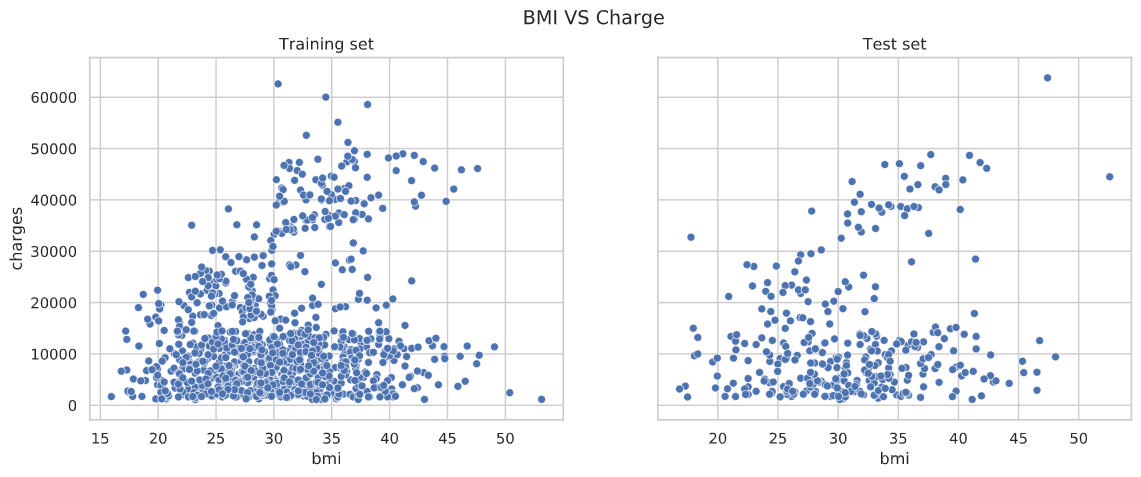
+ Nhận xét: Các đối tượng hút thuốc có xu hướng phải trả chi phí y tế cao hơn những đối tượng không hút thuốc. Trong đó các đối tượng có chi phí y tế cao hơn 10000 và nhỏ hơn 40000 vừa có khả năng là hút thuốc hoặc không hút thuốc

* Tương quan giữa Age và Charges:



+ Nhận xét: độ tuổi càng cao thì chi phí y tế phải trả có xu hướng tăng theo. Ngoài ra có thể thấy độ tuổi chia làm 3 nhóm tăng khác nhau và có hệ số tăng khá tương đồng.

* Tương quan giữa BMI và Charges:



+ Nhận xét: có xu hướng tăng nhưng không rõ ràng. Hầu hết các đối tượng vẫn có chi phí y tế nhỏ hơn 20000

### Mô hình học máy

* Sử dụng các input về thông tin y tế để dự đoán output là viện phí phải trả
* Nhóm sử dụng 3 mô hình học máy: Linear Regression, Random Forest Regression, Gradient Boost Regressor

1. **Linear Regression**

* Sử dụng mô hình Linear Regression để tính output dự đoán

1. **Random Forest Regression**

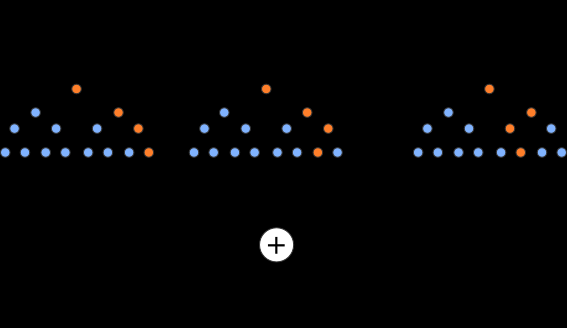
* Model học dựa trên kỹ thuật Bootstraping (thuộc nhánh Ensemble learning)
* Các bước thực hiện:

+ Tạo các tập huấn luyện (*bags*): Chọn ngẫu nhiên có hoàn lại hoặc không hoàn lại các mẫu trong tập huấn luyện và bỏ vào từng *bag* để tiến hành học

+ Tại mỗi *bag*, chọn random các thuộc tính để tiến hành xây dựng cây hồi quy.

+ Siêu tham số cần xác định trước khi chạy (Cần thử nhiều giá trị để chọn ra số cây tốt nhất):

* Số cây được sinh ra
* Số lượng thuộc tính sử dụng khi sinh một cây hồi quy.
* Quá trình xây dựng cây hồi quy trên mỗi *bag* (áp dụng cho cả thuộc tính liên tục và thuộc tính rời rạc):



+ Một vài siêu tham số:

* Số các thuộc tính được chọn để phân lớp: Có thể bằng số thuộc tính hiện có hoặc bằng căn bậc 2 số thuộc tính hiện có
* Chiều cao tối đa của cây: Cần thử nhiều giá trị để chọn ra giá trị tốt nhất
* Số lượng mẫu dữ liệu tối thiểu ở node lá: Cần thử nhiều giá trị để chọn ra giá trị tốt nhất

+ Đối với dữ liệu bao gồm 1 thuộc tính input và 1 thuộc tính output:

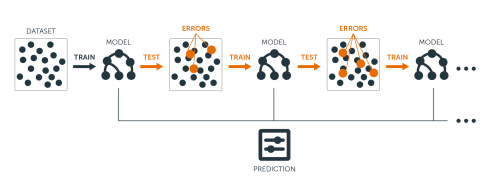
* Vẽ biểu đồ scatter plot biểu diễn dữ liệu
* Tại mỗi giá trị *x\_i* dọc theo trục hoành, tiến hành chia đôi dữ liệu thành 2 tập con lần lượt chứa các phần tử *x* ở bên phải *x\_i* và *x* ở bên trái *x\_i*
* Tại mỗi điểm chia, ta được tạo ra được 1 cây hồi quy. Lấy trung bình giá trị output tại mỗi cây con, ta được output chính thức của cây
* Tính độ lỗi MSE trên các cây hồi quy vừa thu được sau đó chọn cây (điểm phân đôi dữ liệu) có độ lỗi nhỏ nhất. Đó là cây hồi quy cần tìm.

+ Đối với dữ liệu bao gồm nhiều thuộc tính input và 1 thuộc tính output:

* Xét từng thuộc tính input với thuộc tính output, tìm được điểm chia dữ liệu tốt nhất của thuộc tính đó bằng cách thực hiện các bước đã trình bày ở phần "dữ liệu bao gồm 1 thuộc tính input và 1 thuộc tính output"
* Sau khi thu được các điểm tốt nhất tại mỗi thuộc tính, so sánh độ lỗi của các điểm này với nhau và chọn ra thuộc tính với độ lỗi thấp nhất. Tới đây, ta thu được node root của cây quyết định.
* Lặp lại quá trình trên với các thuộc tính còn lại đến khi thỏa điều kiện về chiều cao tối đa và số mẫu dữ liệu tối thiểu ở node lá thì dừng thuật toán
* Quá trình dự đoán: Với input x, ta đưa input này vào tất cả các cây thu được trong quá trình huấn luyện và thu được một số lượng output nhất định. Lấy trung bình cộng các output này, ta được output cần tìm

1. **Gradient Boost Regressor**

* Model học dựa trên kỹ thuật *Boosting* (thuộc nhánh *Ensemble learning*)



* Các bước thực hiện:

1. Tính giá trị trung bình của thuộc tính cần dự đoán (*charges*) và gọi đây là kết quả dự đoán lần thứ 0: *pred0 = meanCharges*
2. Tính giá trị *residuali = actualValue – predi* (i chạy từ 0)
3. Tạo cây hồi quy dự đoán giá trị *residual* (phương pháp đã được trình bày ở trên)
4. Tính giá trị dự đoán lần tiếp theo *predi+1*, (i chạy từ 0)

* Mỗi điểm dữ liệu trong tập train được đưa qua cây hồi quy bên trên. Ta thu được giá trị ước lượng của residual gọi là *predResiduali+1*, (i chạy từ 0)
* *predi+1 = meanCharges + learningRate \* predResiduali*

1. Tính giá trị *residuali+1 = actualValue – predi+1*
2. Lặp lại các bước từ 3 đến 5 cho đến khi số lần lặp vượt quá *max\_iter* hoặc giá trị cần ước lượng không đổi.
3. **Chọn các siêu tham số**

* Sử dụng kỹ thuật Cross Validation để chọn siêu tham số. *k = 3 folds*
* Tạo không gian siêu tham số

+ Số cây sinh ra: *n\_estimators = [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]*

+ Số lượng thuộc tính lớn nhất được sử dụng khi xây dựng cây hồi quy: *max\_features = ['auto', 'sqrt']*

* *auto*: Tương đương số lượng thuộc tính của cả bộ dữ liệu
* *sqrt*: Tương đương căn bậc 2 (làm tròn số) của số lượng thuộc tính của cả bộ dữ liệu

*+* Chiều cao lớn nhất của cây hồi quy: *max\_depth = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None]* - *None* biểu thị chiều cao không giới hạn

*+* Số lượng mẫu tối thiểu ở node lá: *min\_samples\_leaf = [1, 2, 4]*

+ Learning rate (đối với Gradient Boosting): *learning\_rate = [0.01, 0.05, 0.1]*

* Vì không gian tìm kiếm là quá lớn, nên ta chỉ tiến hành tìm kiếm trong *max\_iter=100* lần. Nói cách khác, ta lặp lại *max\_iter=100* lần các công việc sau:

+ Chọn ngẫu nhiên giá trị siêu tham số từ không gian tìm kiếm

+ Tính toán độ lỗi đối với bộ tham số vừa chọn

* Kết thúc bước này, ta chọn được bộ tham số có chất lượng tốt nhất trong số 100 bộ tham số vừa duyệt qua:

+ Đối với Random Forest:

* *n\_estimators: 600,*
* *min\_samples\_split: 2,*
* *min\_samples\_leaf: 4,*
* *max\_features: 'auto',*
* *max\_depth: 40*

+ Đối với Gradient Boosting:

* *n\_estimators: 1000,*
* *min\_samples\_split: 2,*
* *min\_samples\_leaf: 1,*
* *max\_features: 'sqrt',*
* *max\_depth: 10,*
* *learning\_rate: 0.01*
* Giả định rằng các bộ tham số lân cận của nó cũng có chất lượng gần tương đương, tiến hành tìm kiếm vét cạn các lân cận với hy vọng cải thiện chất lượng của bộ tham số hiện tại:

+ Đối với Random Forest:

* *n\_estimators: [600, 800, 1000],*
* *min\_samples\_split: [2, 4, 5],*
* *min\_samples\_leaf: [4, 8, 12],*
* *max\_features: ['auto'],*
* *max\_depth: [30, 40, 50, 60, None]*

+ Đối với Gradient Boosting:

* *n\_estimators: [3000, 5000],*
* *min\_samples\_split: [2, 5],*
* *min\_samples\_leaf: [1, 2],*
* *max\_features: ['auto', 'sqrt'],*
* *max\_depth: [60, 80, 100, None],*
* *learning\_rate: 0.01*
* Kết quả thu được sau khi tìm kiếm trong không gian lân cận sẽ được dùng để dự đoán tập test:

+ Đối với Random Forest:

* *n\_estimators: 1000,*
* *min\_samples\_split: 2,*
* *min\_samples\_leaf: 1,*
* *max\_features: 'sqrt',*
* *max\_depth: 10,*
* *learning\_rate: 0.01*

+ Đối với Gradient Boosting:

* *learning\_rate: 0.01,*
* *max\_depth: 60,*
* *max\_features: 'sqrt',*
* *min\_samples\_leaf: 1,*
* *min\_samples\_split: 2,*
* *n\_estimators: 5000*

### Báo cáo kết quả và nhận xét

* Kết quả kiểm thử trên tập test:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **MAE** | **RMSE** | **R2** | **Accuracy** |
| Linear Regression | 4243.654 | 5926.024 | 0.767 | 0.5553 |
| Random Forest Regression | 2592.958 | 4532.961 | 0.864 | 0.7244 |
| Gradient Boost Regressor | 2545.602 | 4830.882 | 0.845 | 0.7201 |

* Đối với Gradient Boost Regressor (GB), nhóm có được độ quan trọng của các thuộc tính input đối với thuộc tính output sắp xếp giảm dần:

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Từ bảng trên ta có được các thuộc tính input có độ quan trọng cao: Smoker, Sex, BMI, Age
* Xét các thuộc tính quan trọng, kết quả kiểm thử trên tập test:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **MAE** | **RMSE** | **R2** | **Accuracy** |
| Linear Regression | 4243.654 | 5926.024 | 0.767 | 0.5553 |
| Random Forest Regression | 2637.747 | 4571.786 | 0.861 | 0.7208 |
| Gradient Boost Regressor | 2575.32 | 4863.763 | 0.843 | 0.7307 |

## IV. Đánh giá đồ án

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Mức độ hoàn thành công việc** | **Đóng góp** |
| 18120078 |  |  |
| 18120253 |  |  |
| 18120201 |  |  |

## VII. Nguồn tham khảo

* [Random Forest Regression](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html)
* [Gradient Boosting Regression](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html)
* [The Element of Statistical Learning](https://web.stanford.edu/~hastie/ElemStatLearn/)
* [Randomforest2001.pdf](https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf)
* [Medical Cost Prediction](https://github.com/SahilChachra/Medical-Cost-Prediction)
* [Hyperameter tuning the Random Forest in Python](https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74)