Báo cáo Dự án Machine Learning: Dự đoán Chi phí Y tế Cá nhân

Tác giả: Nguyễn Khôi Nguyên

1. Giới thiệu bài toán và Dữ liệu

Dự án sử dụng bộ dữ liệu **"Medical Cost Personal Datasets"**. Đây là bộ dữ liệu kinh điển, phù hợp để thực hành các quy trình xử lý và xây dựng mô hình. Dữ liệu bao gồm các cột sau:

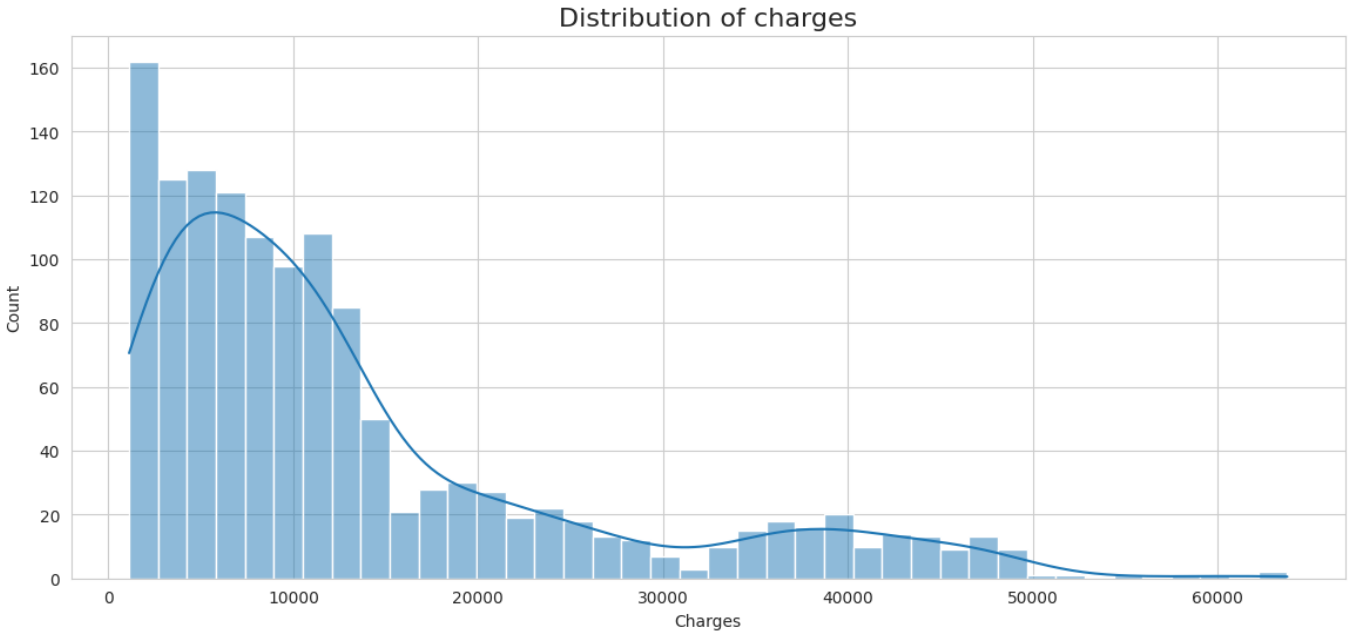
- `age`: Tuổi
- `sex`: Giới tính (male, female)
- `bmi`: Chỉ số cơ thể
- `children`: Số con
- `smoker`: Tình trạng hút thuốc (yes, no)
- `region`: Khu vực sinh sống
- `charges`: Chi phí y tế (đây là biến mục tiêu cần dự đoán).

2. Phân tích và Khám phá Dữ liệu (EDA)

Giai đoạn này giúp hiểu rõ hơn về đặc điểm của dữ liệu và mối quan_hệ giữa các biến, từ đó định hướng cho việc xây dựng mô hình.

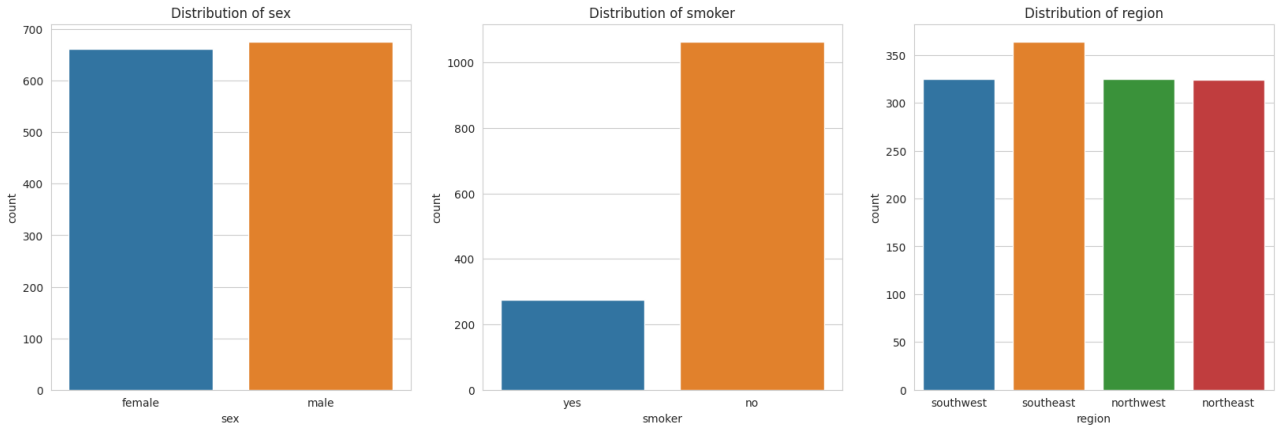
2.1. Phân tích biến mục tiêu `charges`

Biểu đồ phân phối của `charges` cho thấy dữ liệu bị **lệch phải (right-skewed)**. Phần lớn chi phí tập trung ở mức thấp, nhưng có một số trường hợp với chi phí rất cao. Điều này gợi ý rằng việc áp dụng phép biến đổi Logarithm sẽ giúp mô hình hoạt động hiệu quả hơn.

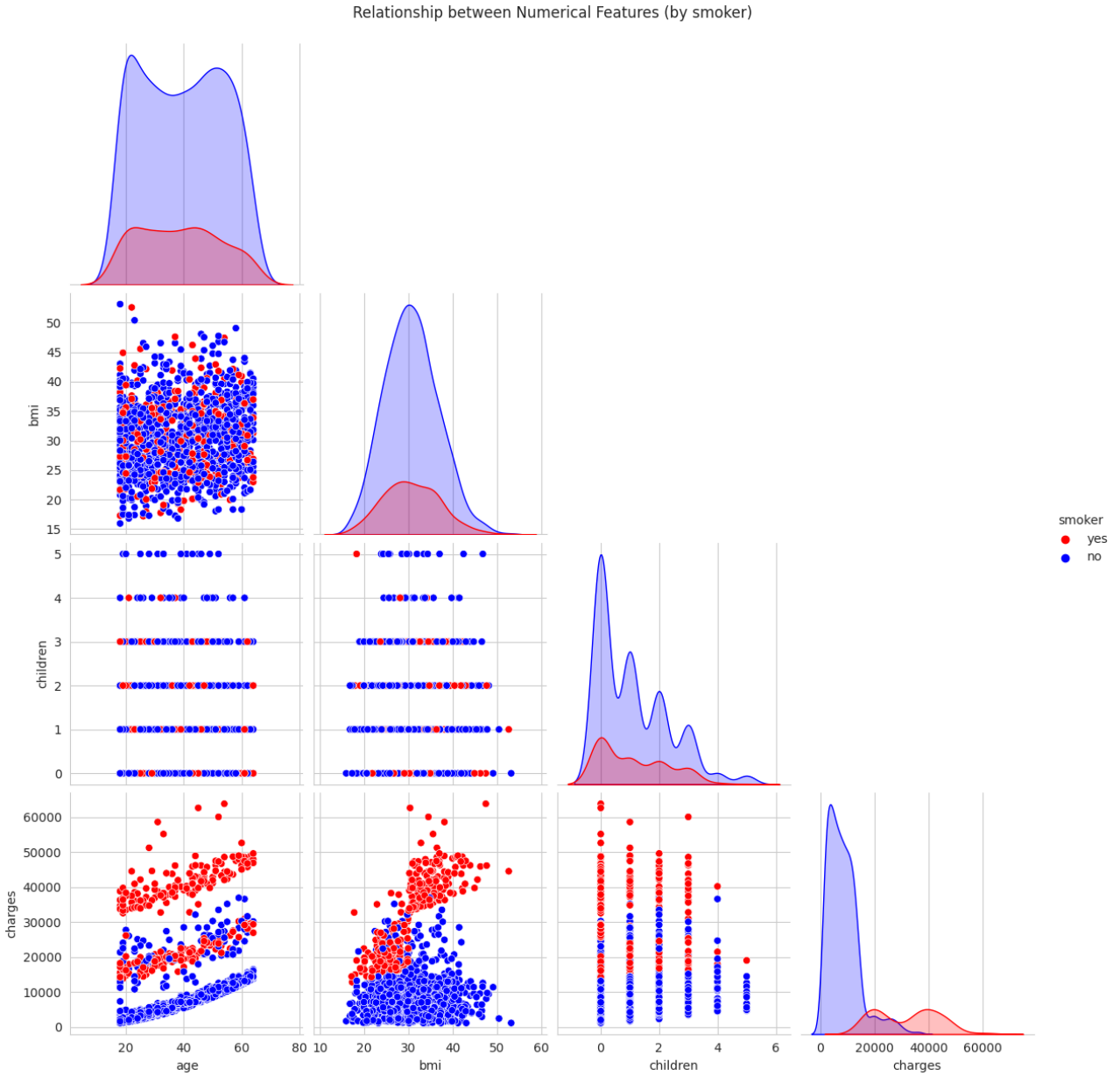


2.2. Phân tích các biến đầu vào

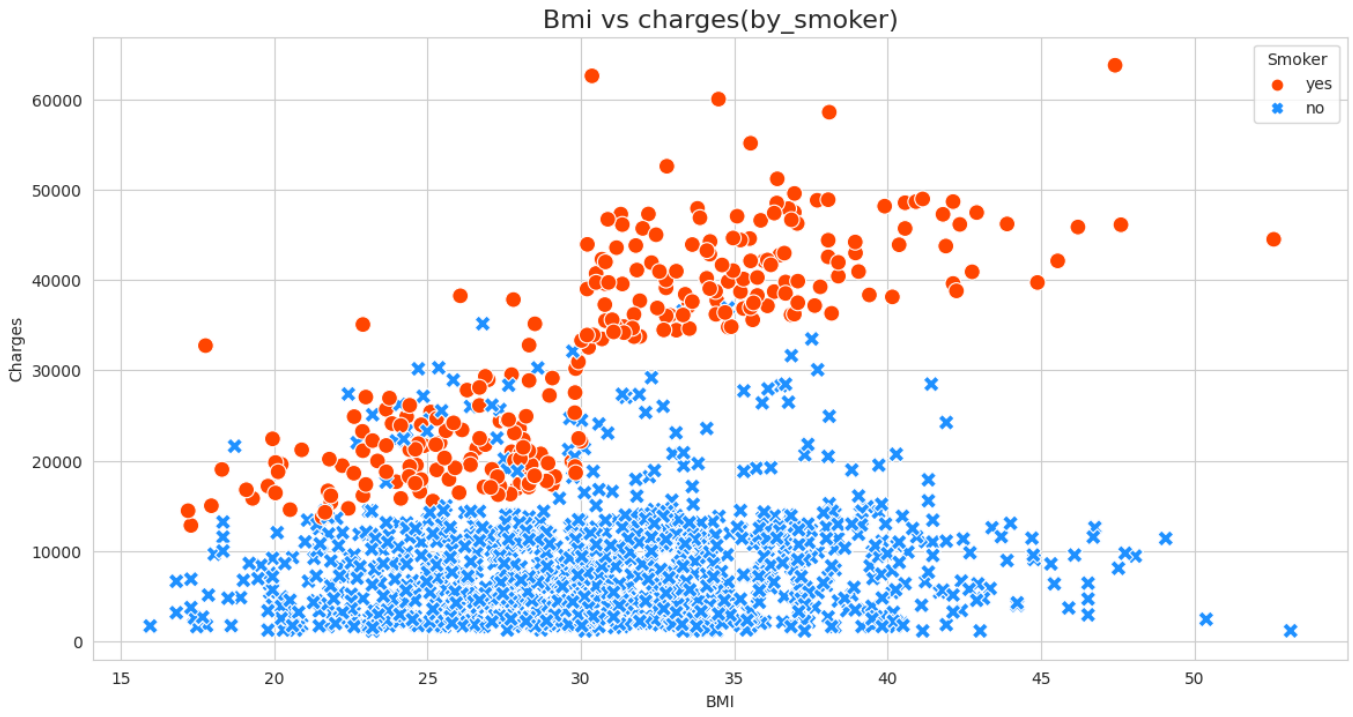
- **Biến phân loại (`sex`, `smoker`, `region`):** Dữ liệu cho thấy tỷ lệ nam/nữ khá cân bằng. Đa số người trong bộ dữ liệu không hút thuốc, và số lượng người từ các khu vực cũng tương đối đồng đều.



- **Mối quan hệ giữa các biến:**
    - Biểu đồ `pairplot` cho thấy `smoker` **(hút thuốc) là yếu tố có ảnh hưởng mạnh mẽ nhất** đến chi phí y tế. Nhóm người hút thuốc có chi phí cao hơn hẳn.
    - `age` **(tuổi)** có tương quan dương rõ rệt, chi phí có xu hướng tăng theo tuổi.
    - `bmi` **(chỉ số cơ thể)** cũng có ảnh hưởng, đặc biệt là ở nhóm người hút thuốc. Với những người hút thuốc, khi chỉ số BMI tăng cao, chi phí y tế tăng vọt.



Biểu đồ dưới đây thể hiện rõ sự tương tác mạnh mẽ giữa `bmi` và `smoker`.



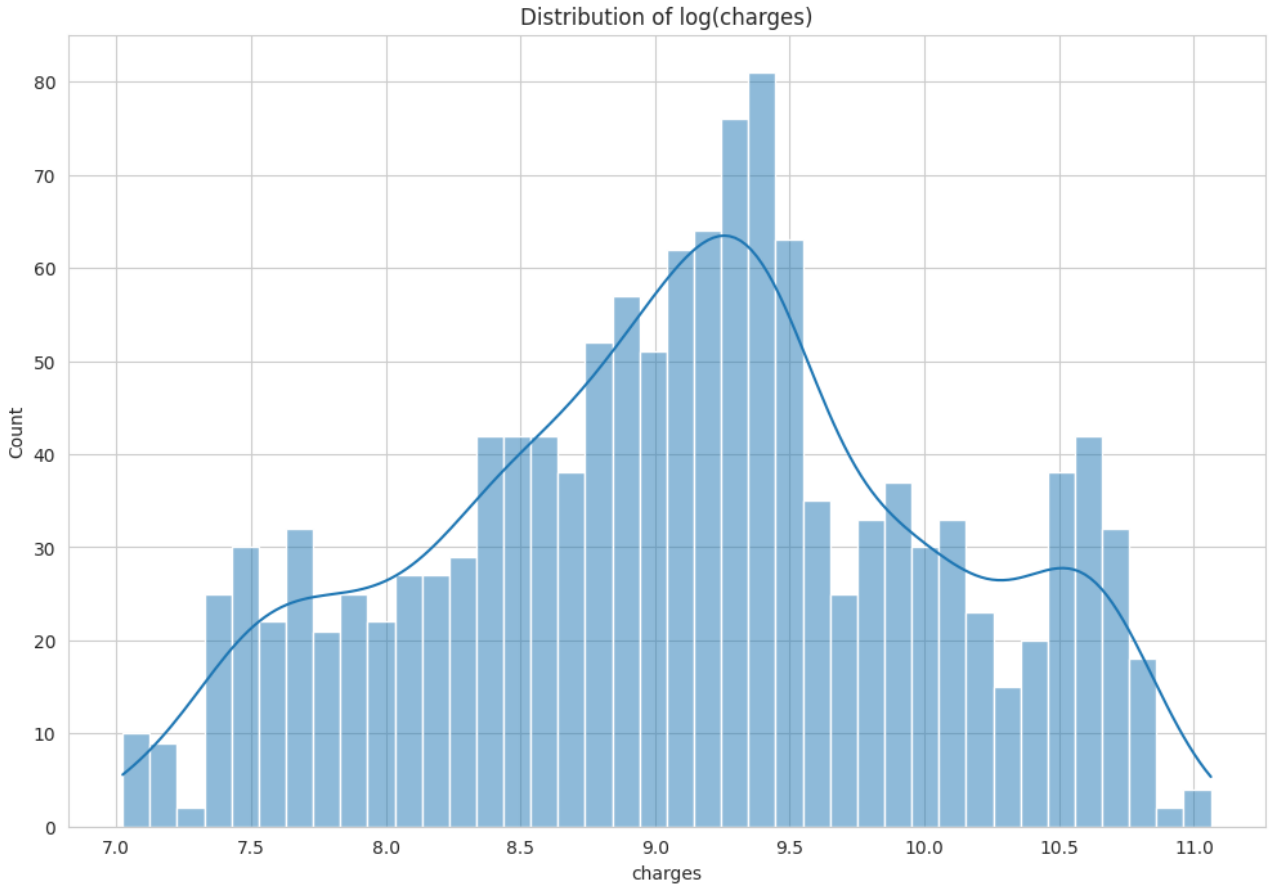
---

3. Tiền xử lý dữ liệu và Xây dựng Pipeline

3.1. Tiền xử lý dữ liệu

Các bước tiền xử lý sau đã được thực hiện để tối ưu hóa dữ liệu đầu vào cho mô hình:

1. **Biến đổi Logarithm:** Áp dụng phép biến đổi `log1p` cho cột `charges` để xử lý độ lệch.



2. **Mã hóa biến phân loại:** Sử dụng `OneHotEncoder` để chuyển đổi các cột `sex`, `smoker`, `region` thành dạng số.
3. **Chuẩn hóa dữ liệu:** Sử dụng `StandardScaler` cho các cột số (`age`, `bmi`, `children`) để đưa chúng về cùng một thang đo.

3.2. Kỹ thuật đặc trưng (Feature Engineering)

Để tăng hiệu quả của mô hình, các đặc trưng mới đã được tạo ra từ những đặc trưng có sẵn:

- `bmi_obese`: Một biến nhị phân cho biết một người có bị béo phì hay không (khi `bmi` >= 30).
- `age_smoker`: Biến tương tác giữa `age` và `smoker` để mô hình nắm bắt được mối quan hệ phức tạp đã thấy trong EDA.

---

4. Kết quả và Đánh giá Mô hình

Dự án đã xây dựng và so sánh hai mô hình để lựa chọn ra phương pháp hiệu quả nhất.

4.1. Mô hình Baseline: Hồi quy Tuyến tính (Linear Regression)

Đây là mô hình cơ sở được xây dựng với các bước tiền xử lý cơ bản.

- **Kết quả:**
  - **R^2 Score:** 0.6067
  - **MAE (Sai số tuyệt đối trung bình):** $3,888.44

4.2. Mô hình Cải tiến: XGBoost

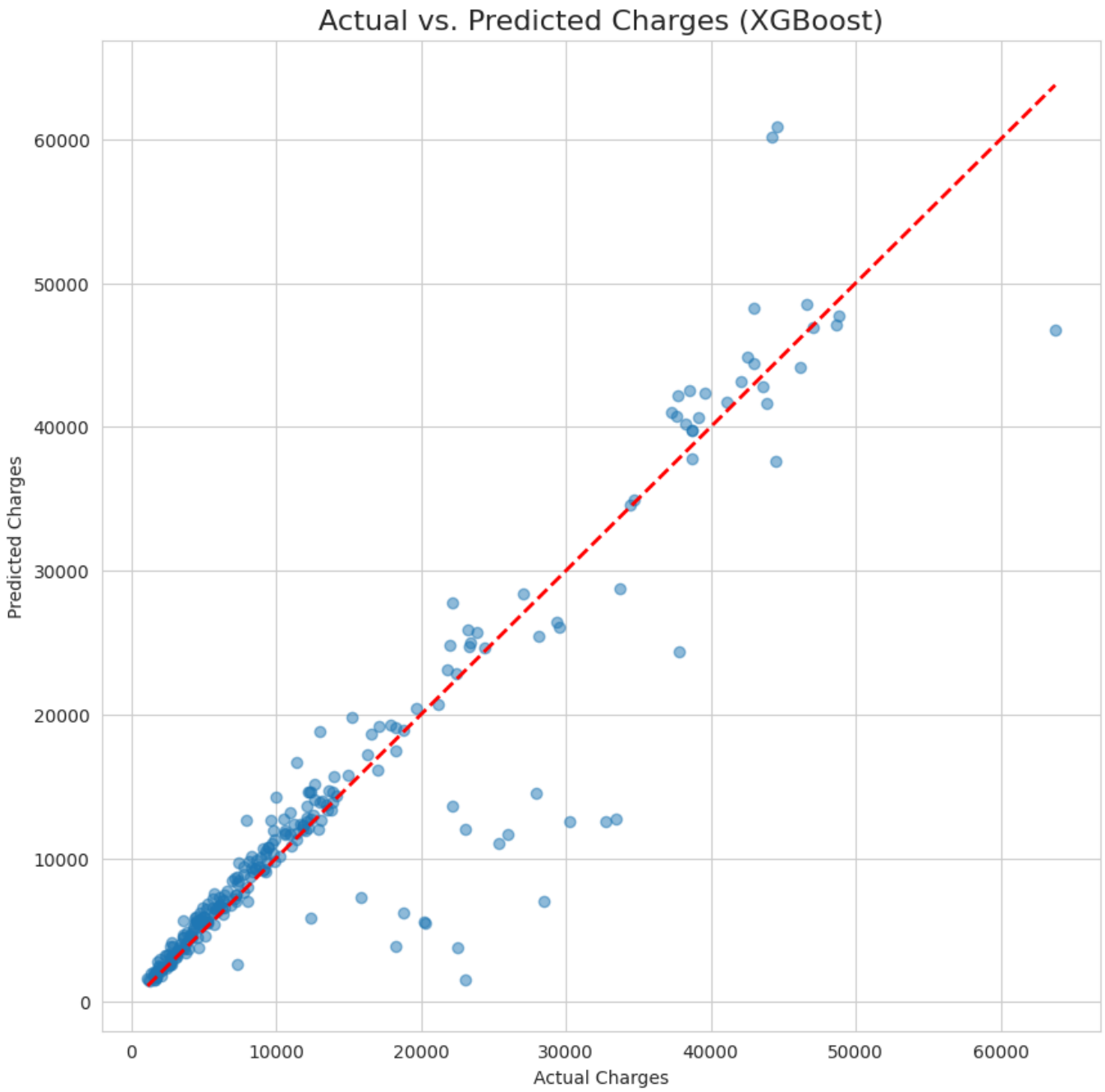
Mô hình này sử dụng thuật toán XGBoost và các đặc trưng đã được tạo thêm.

- **Kết quả:**
  - **R^2 Score:** 0.8622
  - **MAE (Sai số tuyệt đối trung bình):** $2,171.02

4.3. So sánh và Lựa chọn

Mô hình	R^2 Score	MAE (Sai số tuyệt đối trung bình)
Hồi quy Tuyến tính	0.6067	$3,888.44
XGBoost (Cải tiến)	**0.8622**	**$2,171.02**

Lựa chọn: Dựa trên kết quả so sánh, mô hình **XGBoost** cho hiệu suất vượt trội với R^2 cao hơn và sai số MAE thấp hơn đáng kể. Vì vậy, mô hình này đã được chọn để triển khai.



---

5. Triển khai mô hình

Mô hình XGBoost tốt nhất đã được lưu lại và triển khai dưới dạng một ứng dụng web hoàn chỉnh, bao gồm hai thành phần chính:

5.1. API với FastAPI

Một API (Application Programming Interface) đã được xây dựng bằng FastAPI (file `main.py`). API này đóng vai trò là "bộ não" của ứng dụng:

- Tải pipeline mô hình đã được huấn luyện (`.joblib`).
- Tạo một endpoint `/predict` để nhận dữ liệu đầu vào (tuổi, giới tính, bmi,...) dưới dạng JSON.
- Thực hiện các bước tiền xử lý và kỹ thuật đặc trưng tương tự như khi huấn luyện.
- Trả về kết quả dự đoán chi phí.

5.2. Giao diện Web với Streamlit

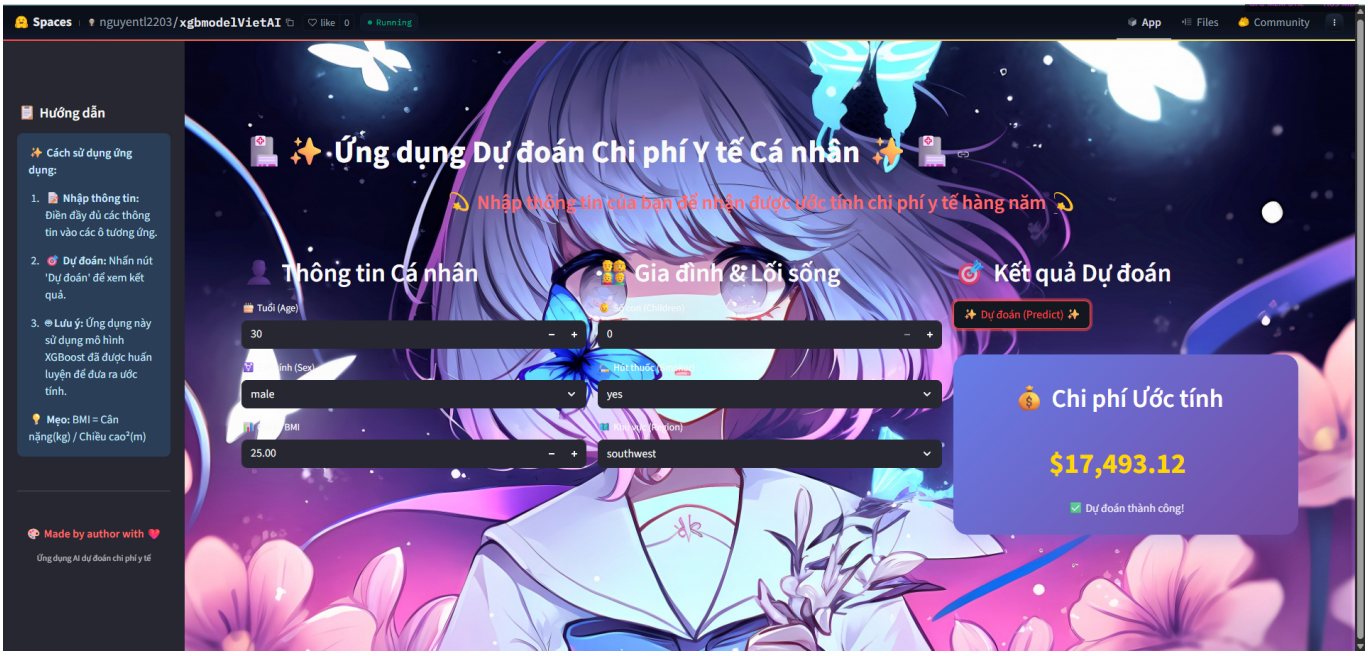
Một giao diện web thân thiện với người dùng đã được xây dựng bằng Streamlit (file `app.py`). Ứng dụng này cho phép:

- Người dùng nhập thông tin cá nhân qua các ô nhập liệu trực quan.
- Khi nhấn nút "Dự đoán", giao diện sẽ gửi yêu cầu đến API FastAPI.
- Hiển thị kết quả dự đoán chi phí mà API trả về một cách rõ ràng.

5.3. Cách truy cập và sử dụng (Hugging Face Spaces)

Ứng dụng đã được triển khai công khai và có thể được truy cập dễ dàng qua các bước sau:

- **Bước 1:** Truy cập vào địa chỉ Spaces trên Hugging Face: https://huggingface.co/spaces/nguyentl2203/XgbmodelVietAI
- **Bước 2:** Trên giao diện web, nhập các thông tin cá nhân, gia đình và lối sống vào các ô tương ứng.
- **Bước 3:** Nhấn nút "✨ **Dự đoán (Predict)** ✨". Kết quả chi phí y tế ước tính sẽ được hiển thị ngay lập tức ở mục "🎯 **Kết quả Dự đoán**".



---

6. Kết luận và Hướng phát triển

6.1. Kết luận

Dự án đã thành công trong việc xây dựng một quy trình Machine Learning hoàn chỉnh, từ phân tích dữ liệu, huấn luyện, so sánh mô hình cho đến triển khai sản phẩm cuối. Mô hình XGBoost đã chứng tỏ hiệu quả cao trong việc dự đoán chi phí y tế và được triển khai thành công thành một ứng dụng web tương tác.

6.2. Hướng phát triển

Trong tương lai, dự án có thể được cải thiện thêm bằng các phương pháp:

- **Tinh chỉnh tham số (Hyperparameter Tuning):** Sử dụng các kỹ thuật như `GridSearchCV` hoặc `RandomizedSearchCV` để tìm ra bộ tham số tối ưu nhất cho mô hình XGBoost.
- **Thử nghiệm các mô hình khác:** So sánh với các mô hình mạnh mẽ khác như LightGBM, CatBoost hoặc các mô hình Deep Learning.
- **Cải thiện giao diện:** Nâng cấp front-end bằng các framework như React, Vue.js để tăng trải nghiệm người dùng.