

**Project môn học**

Lập trình nâng cao trong khoa học dữ liệu

**Nhóm thực hiện**

Tạ Văn Nhân

Nguyễn Hoàng Nam

Phạm Trung Hiếu

Nguyễn Quang Huy

**Đề tài**

Ứng dụng python trong tự động mô hình hóa thống kê

**Repository**

https://github.com/namnhGru/auto-modelling-ds-vnu.git

Table of Contents

[I. Mở đầu 3](#_Toc5445915)

[II. Nội dung dự án 4](#_Toc5445916)

[1. Thiết kế 4](#_Toc5445917)

[a. Chức năng từng class 5](#_Toc5445918)

[2. Hoạt động 9](#_Toc5445919)

[III. Kết quả thực hiện 10](#_Toc5445920)

[IV. Đánh giá ưu nhược điểm 12](#_Toc5445921)

[V. Những mục chưa làm được 12](#_Toc5445922)

[VI. Lời cảm ơn 13](#_Toc5445923)

# Mở đầu

Thưa bạn đọc,

Trong thời gian học tập tại HUS, chuyên ngành khoa học dữ liệu, nhóm nghiên cứu nhận thấy rằng việc mô hình hóa thống kê không chỉ đơn thuần giải quyết về mặt toán học, mà còn là những thao tác dữ liệu trên phần mềm với hiệu năng cao, đem lại nhiều lợi ích trong tính toán.

Đặc biệt, nhóm nhận thấy ngôn ngữ Python hay các ngôn ngữ lập trình nói chung có khả năng xây dựng nên các kịch bản cũng như tạo ra các sản phẩm mang tính nhất quán, giúp giảm thiểu nhiều thao tác lặp lại khi tính toán trên số liệu, cụ thể là những kiểm định, ước lượng thống kê hay các thao tác đọc ghi dự đoán dữ liệu thông thường. Nhờ có Python, các thao tác trên dữ liệu của nhóm đã nhất quán hơn cũng như tránh bị lặp lại ở thời điểm tương lai

Dự án này là một nỗ lực trong việc tổng hợp các thao tác trên dữ liệu lặp như vừa nêu, cũng như là tiền đề để nghiên cứu những ứng dụng và công nghệ cao hơn trong việc tự động hóa mô hình cũng như học máy, giảm các tải phải bỏ ra của các nhà khoa học dữ liệu để đem lại sản phẩm một cách nhanh và chính xác nhất. Dự án cũng hướng đến như một “hộp đen” xử lý dữ liệu, giúp cho những người sắp bước vào con đường dữ liệu có một công cụ mô tả được quá trình xử lý đơn giản nhất, có được cái nhìn tổng quan nhất khi theo đuổi ngành khoa học dữ liệu.

Hà Nội, 04/6/2019,

Nhóm dự án ứng dụng Python trong tự động mô hình hóa

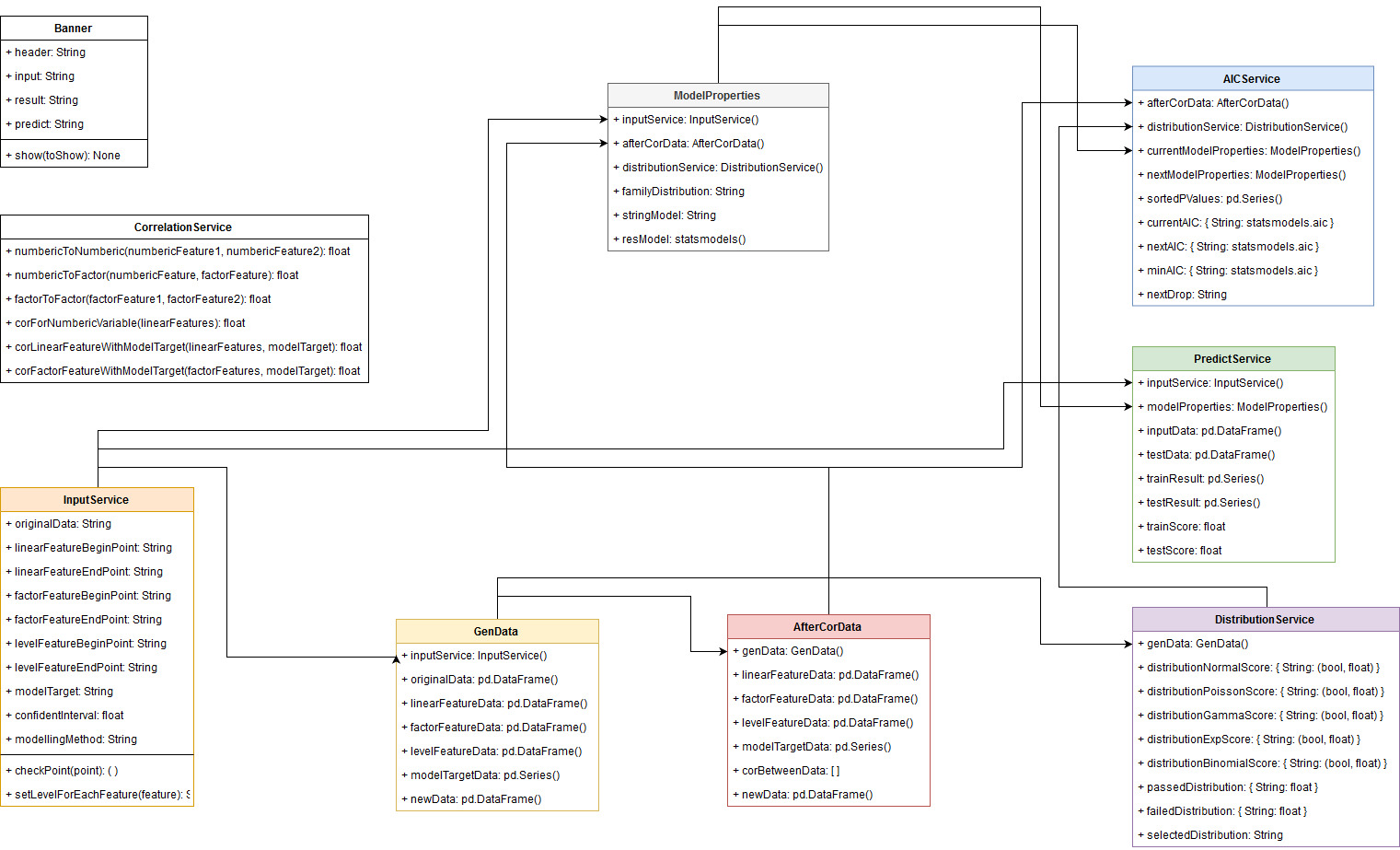
# Nội dung dự án

## Thiết kế

Dự án được thiết kế với các mục tiêu sau:

* Tối thiểu nhân lực cho việc triển khai coding
* Code có khả năng sử dụng lại khi cần thiết
* Code có khả năng mở rộng khi cần thiết
* Code có khả năng bảo trì dễ dàng khi cần thiết

Với mục tiêu đó, dự án được thiết kế như sau:

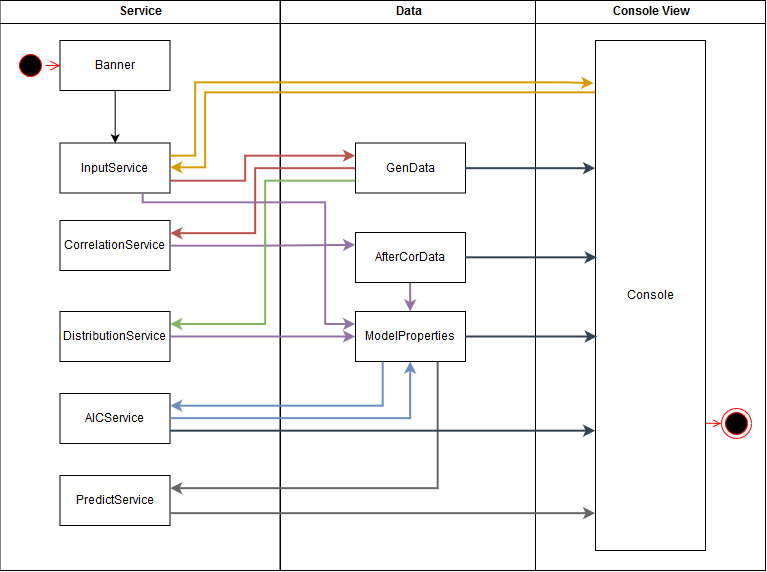


*Hình 1: Thiết kế class trong dự án*

### Chức năng từng class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Class** | **Mục đích** | **Trường** | **Phương thức** |
| 1 | InputService | * Tiếp nhận Input từ người dùng. * Khối này không hoặc có ít nhất có thể tính năng xử lý các đầu vào người dùng nhập | 1. OriginalData: nhận đường dẫn dữ liệu 2. linearFeatureBegin/EndPoint: nhận khoảng dữ liệu với dạng đặc trưng là tuyến tính 3. factorFeatureBegin/EndPoint: nhận khoảng dữ liệu với dạng đặc trưng là category đều 4. levelFeatureBegin/EndPoint: nhận khoảng dữ liệu với dạng đặc trưng là category phân cấp 5. modelTarget: nhận dữ liệu quy định là response của mô hình 6. confidentInterval: nhận độ tin cậy trên toàn dự án 7. modellingMethod: nhận phương thức lên mô hình (lm hay glm) | 1. CheckPoint: nhận input String khoảng dữ liệu, trả về cận của khoảng dữ liệu 2. SetLevelForEachFeature: nhận input string category phân cấp, trả về thứ tự phân cấp 3. Ngoài ra là các phương thức get/ set xử lý cho trường |
| 2 | GenData | * Xử lý các input và trả về bảng dữ liệu đã được quy định trường và thuộc tính | 1. InputService: nhận object InputService để lấy dữ liệu 2. originalData: dữ liệu gốc đã được xử lý 3. linearFeatureData: dữ liệu tuyến tính đã được xử lý 4. factorFeatureData: dữ liệu category đều đã được xử lý 5. levelFeatureData: dữ liệu category phân cấp đã được xử lý 6. modelTargetData: dữ liệu response đã được xử lý 7. newData: bảng dữ liệu mới đã được xử lý | Các phương thức get/ set xử lý dữ liệu cho trường |
| 3 | AfterCorData | * Lấy các dữ liệu đã xử lý corTest, ANOVATest, trả về bảng dữ liệu thu gọn/ bổ sung sau khi loại bỏ các trường không liên quan và sinh các trường mới tới response | 1. genData: nhận object GenData để lấy dữ liệu cần xử lý 2. linearFeatureData: dữ liệu tuyến tính đã được xử lý corTest 3. factorFeatureData: dữ liệu category đều đã được xử lý ANOVA Test 4. levelFeatureData: dữ liệu category phân cấp đã được xử lý ANOVA Test 5. modelTargetData: dữ liệu response đã được xử lý 6. corBetweenData: dữ liệu interact giữa các đặc trưng 7. newData: bảng dữ liệu mới đã được xử lý | Các phương thức get/ set xử lý dữ liệu cho trường |
| 4 | CorrelationService | * Xử lý CorTest, ANOVA Test cho dữ liệu | Không có | 1. numbericToNumberic: xử lý cor giữa các biến dạng tuyến tính 2. numbericToFactor:   xử lý cor giữa các biến dạng category và dạng tuyến tính   1. factorToFactor: xử lý cor giữa các biến dạng category 2. corForNumbericVariable: xử lý cor giữa các feature tuyến tính 3. corLinearFeatureWithModelTarget: xử lý cor giữa các feature tuyến tính và response 4. corFactorFeatureWithModelTarget: xử lý cor giữa các feature category và response |
| 5 | DistributionService | * Tìm phân phối cho response thông qua các kiểm định | 1. genData: nhận object GenData để lấy dữ liệu response 2. selectedDistribution: phân phối của response 3. passedDistribution: các phân phối sau kiểm định thỏa mãn 4. failedDistribution: các phân phối sau kiểm định không thỏa mãn 5. Distribution”X”Score: điểm kiểm định quyết định tính thỏa mãn của các phân phối | Các phương thức get/ set xử lý dữ liệu cho trường |
| 6 | ModelProperties | * Xử lý việc lên model cho dữ liệu | 1. inputService: nhận object InputService để lấy dữ liệu chọn mô hình 2. afterCorData: nhận object AfterCorData để lấy dữ liệu đã điều chỉnh sau kiểm định tương quan 3. distributionService: nhận object DistributionService để lấy dữ liệu phân phối của response 4. familyDistribution: xác định link function nếu mô hình được chọn là glm 5. stringModel: biểu thức của mô hình 6. resModel: mô hình được sinh ra | Các phương thức get/ set để xử lý dữ liệu cho trường |
| 7 | AICService | * Lựa chọn mô hình tốt nhất từ mô hình đã sinh | 1. afterCorData: nhận object afterCorData để xử lý 2. distributionService: nhận object distributionService để xử lý 3. currentModelProperties: nhận object ModelProperties từ mô hình đã tính 4. nextModelProperties: nhận object ModelProperties sinh ra sau mỗi lần so sánh AIC 5. sortedPValues: nhận danh sách thứ tự các pvalue của các đặc trưng 6. nextDrop: tên đặc trưng sẽ loại bỏ tiếp theo trong mô hình 7. currentAIC: AIC của mô hình hiện tại 8. nextAIC: AIC của mô hình tiếp theo 9. minAIC: AIC của mô hình tốt nhất | Các phương thức get/ set để xử lý trường |
| 8 | PredictService | * Xử lý dự đoán dữ liệu train và test | 1. inputService: nhận object InputService để lấy thông tin dữ liệu đầu vào 2. modelProperties: nhận object ModelProperties để lấy thông tin mô hình tốt nhất 3. testData: dữ liệu kiểm định 4. trainResult: dữ liệu train sau khi dự đoán 5. testResult: dữ liệu test sau khi dự đoán 6. trainScore: điểm dự đoán dữ liệu train 7. testScore: điểm dự đoán dữ liệu test | Các phương thức xử lý get/ set cho trường |
| 9 | Banner | * Đề mục, chỉ mục cho hiển thị trên console | 1. headerBanner: Đề mục đầu trang 2. inputBanner: chỉ mục nhập dữ liệu 3. resultBanner: chỉ mục hiển thị kết quả 4. predictBanner: chỉ mục hiển thị dự báo | toShow: hiển thị banner tùy ý |

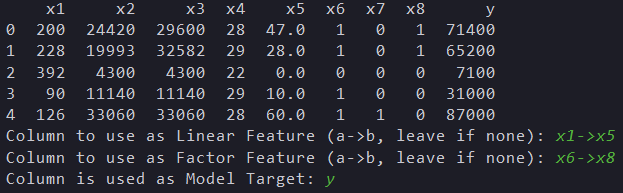
## Hoạt động



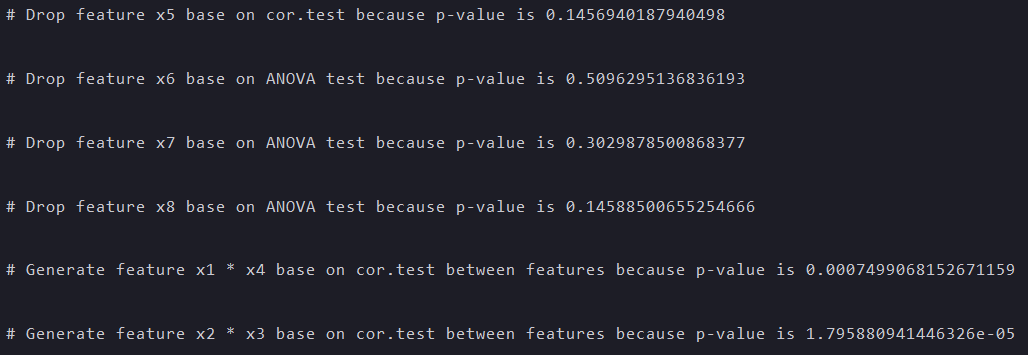
*Hình 2: Dòng chảy hoạt động của project*

Hoạt động của sản phẩm được mô tả kỹ càng như trên hình vẽ, với các flow cùng màu là cùng một quá trình vào ra của dữ liệu cũng như xử lý. Bạn đọc có thể bắt đầu từ điểm Begin ở trên cùng bên trái Service Layer, lần theo dấu mũi tên màu đến điểm End ở dưới cùng bên phải ConsoleView Layer để kết thúc flow của sản phẩm

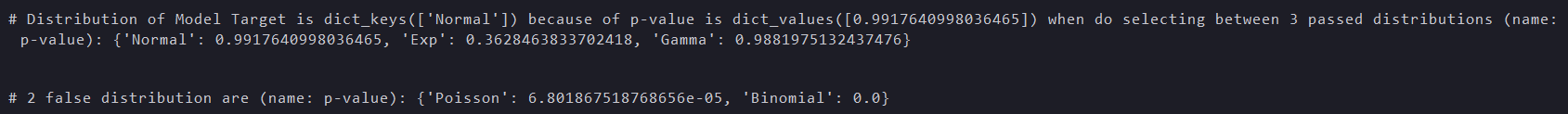
# Kết quả thực hiện



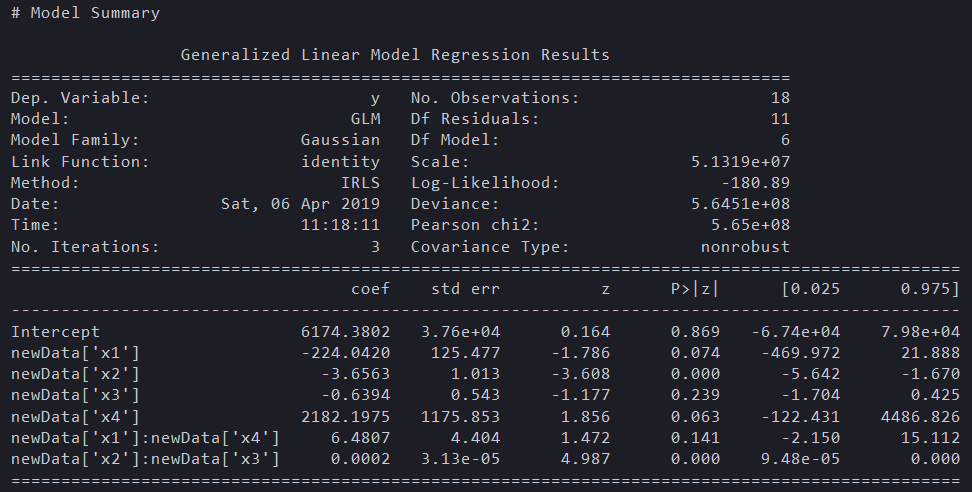
*Hình 3: InputService thực thi*



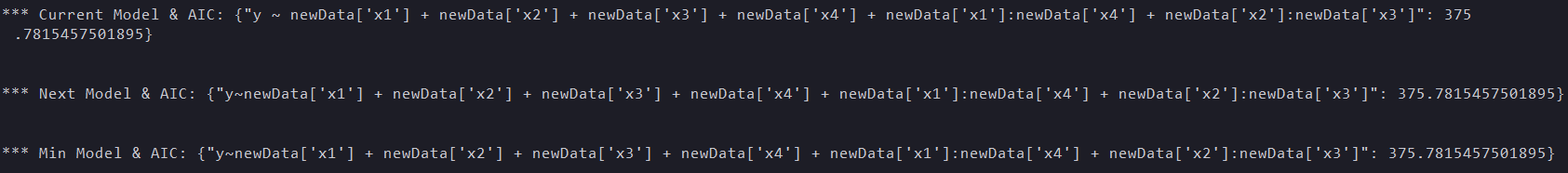
*Hình 4: CorrelationService thực thi*



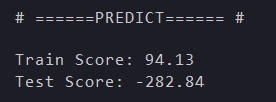
*Hình 5: DistributionService thực thi*



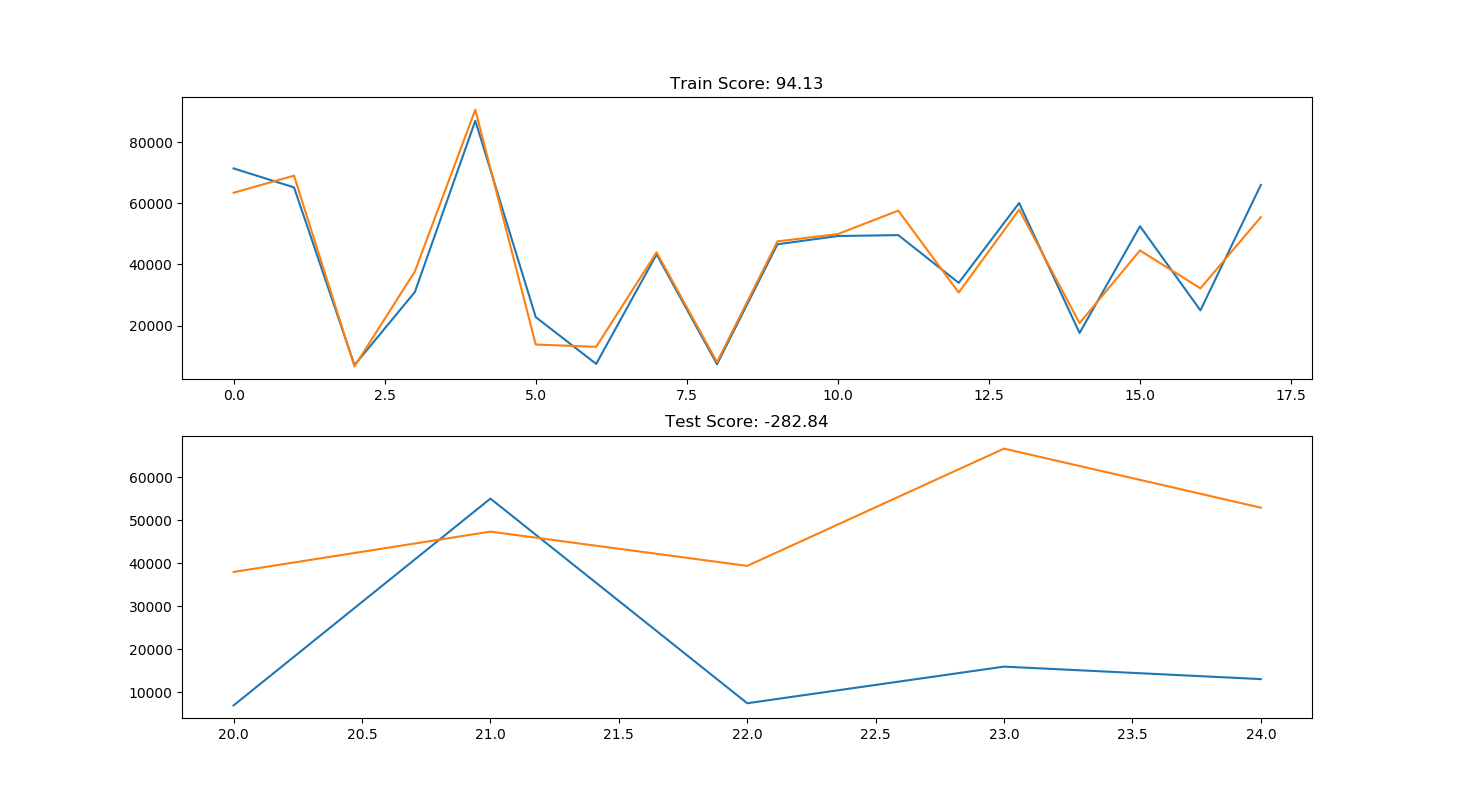
*Hình 6: ModelProperties thực thi*



*Hình 7: AIC Service thực thi*



*Hình 8: PredictService thực thi*



*Hình 9: Đồ thị dự đoán*

# Đánh giá ưu nhược điểm

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| Loại bỏ các thao tác lặp thông thường của nhà phân tích dữ liệu như đọc dữ liệu, kiểm định thống kê | Phụ thuộc vào 1 kịch bản chung |
| Phù hợp để làm quy trình sản xuất phần mềm | Logic code phức tạp |
| Kết quả mô hình được tính toán trong thời gian ngắn | Bảo trì khó khăn vì phải phiên dịch từ thuật toán toán học sang ngôn ngữ phần mềm phụ thuộc vào thiết kế |
|  | Kịch bản kiểm định còn hạn chế |

# Những mục chưa làm được

1. Xử lý trường category phân cấp chưa được ổn định nên chưa đưa vào flow
2. Chưa thử nghiệm đủ nhiều mô hình để đánh giá hiệu năng
3. CorrelationService đang có vấn đề, tuy nhiên là một module quan trọng nên hiện vẫn để trong flow
4. Chưa có giao diện đơn giản cho việc nhập liệu để tránh lặp lại

# Lời cảm ơn

Trong quá trình xây dựng sản phẩm, nhóm đã gặp không ít khó khăn về công nghệ, tuy nhiên nhờ sự chỉ bảo tận tình của thầy Vũ Tiến Dũng và giảng viên Nguyễn Tiến Hưởng mà đã có thể tương đối hoàn thành sản phẩm. Xin cảm ơn thầy và bạn rất nhiều.

Ngoài ra, những khó khăn về kiến thức thống kê khi làm sản phẩm cũng được các thành viên trong lớp giải đáp nhiệt tình, xin được cám ơn các bạn.

Lời cuối, cám ơn các thành viên trong nhóm dự án đã không kể thời gian, công việc cá nhân cố gắng hoàn thành dự án đúng hạn bảo vệ.

Xin cảm ơn!

Hà Nội, 04/6/2019,

Nhóm dự án ứng dụng Python trong tự động mô hình hóa