Báo cáo Giữa kỳ Nhận dạng mẫu

Loan Application Approval Prediction

Thành viên nhóm
Võ Thành Đạt - 22280010
Vũ Ngọc Phương - 22280072
Trần Tấn Tiến - 22280095

I. Giới thiệu về bộ dữ liệu

Bộ dữ liệu này chứa thông tin chi tiết về khách hàng muốn vay tiền, phục vụ cho việc đánh giá quyết định duyệt hoặc từ chối các đơn xin vay. Mục tiêu của bài toán là xây dựng một mô hình học máy có khả năng dự đoán quyết định chấp nhận hoặc từ chối đơn xin vay dựa trên các dữ liệu của khách hàng có trong bộ dữ liệu.

Các đặc điểm này có thể bao gồm thông tin cá nhân, tình trạng tài chính, lịch sử tín dụng và các yếu tố khác có liên quan đến khả năng trả nợ của khách hàng. Mô hình học máy sẽ phân tích các yếu tố này để đưa ra dự đoán chính xác về việc liệu một đơn xin vay sẽ được chấp nhận hay từ chối, giúp các tổ chức tài chính đưa ra quyết định nhanh chóng và hiệu quả.

Thông qua việc áp dụng các thuật toán học máy, chúng ta hy vọng có thể tối ưu hóa quy trình xét duyệt đơn vay, giảm thiểu rủi ro và cải thiện chất lượng dịch vụ cho khách hàng.

II. Model Development

1. Data Preprocessing

a) Mô tả dữ liệu

Bộ dữ liệu bao gồm 39,717 quan trắc và 111 biến (features). Trong đó có 87 biến kiểu numerical và 24 biến kiểu object. Trong đó, biến mục tiêu là `loan_status` (trạng thái của khoản vay), trong khi các biến đặc trưng bao gồm thông tin về khách hàng như thu nhập, điểm tín dụng, lịch sử tín dụng, tình trạng hôn nhân, nghề nghiệp, và các yếu tố khác.

b) Loại bỏ các biến không sử dụng

- Biến Entity như `id`, `member_id`. Đây là các biến mang chức năng nhận dạng, hay nói cách khác chúng không mang tính dự đoán nên ta có thể loại bỏ chúng đi.
- Các biến NULL mang giá trị NULL ở toàn bộ dữ liệu cũng là các biến cần xóa đi.
- Ngoài ra còn có một số biến không có ý nghĩa hoặc không đóng góp vào việc xây dựng model như:
  - Biến chỉ mang một giá trị duy nhất trên toàn bộ tập dữ liệu. Ví dụ như `pymnt_plan`, `collections_12_mths_ex_med`, v.v.
  - Biến `desc` và `title` lần lượt là mô tả và tiêu đề của đơn xin vay nợ. Cả 2 biến đều nói về lý do muốn vay nợ và thường được viết khác nhau giữa mỗi người. Bởi vì ta đã có biến `purpose` để phân loại mục đích vay nợ và được định dạng về 1 format giống nhau, dễ sử dụng nên hoàn toàn có thể thay thế 2 biến trên nên ta sẽ loại bỏ chúng.

c) Chuyển đổi kiểu dữ liệu

- **Các biến** `int_rate` và `revol_util`, hiện đang ở dạng **string** và chứa ký hiệu %, sẽ được chuyển đổi sang kiểu **float** và định dạng số thập phân.
- **Các biến** `issue_d`, `last_pymnt_d`, `last_credit_pull_d`, và `earliest_cr_line`, cũng đang ở dạng **string**, sẽ được chuyển đổi sang kiểu dữ liệu **datetime**. Sau đó, các biến này sẽ được tách thành các thành phần tháng và năm.

d) Encoding

- **Label Encoding**: Áp dụng cho các biến `term`, `sub_grade`, `emp_length`, và `purpose`.
- **Phân nhóm địa lý**:
  - Phân `addr_state` thành bốn khu vực: **Northeast**, **Midwest**, **South**, và **West**.
  - Tạo biến mới `region` và loại bỏ `addr_state`.
- **One Hot Encoding**: Áp dụng cho các biến `next_pymnt_d`, `verification_status`, `home_ownership`, và `region`.
- **Loại bỏ biến không cần thiết**: Loại bỏ các biến `grade`, `emp_title`, và `zip_code` vì chúng có ý nghĩa tương tự với các biến khác và không đóng góp nhiều cho mô hình.

e) Xử lý Outliers

Trong các đặc trưng, có hai biến chứa nhiều giá trị ngoại lai nghiêm trọng: `installment`, `annual_inc`.

Biến `installment` có giá trị **trung bình** chênh lệch lớn so với **trung vị** do xuất hiện nhiều giá trị ngoại lai. Biến `annual_inc` có một vài giá trị ngoại lai quá lớn, làm phân phối bị lệch nghiêm trọng.

Chúng ta sẽ xử lý các giá trị ngoại lai bằng cách sử dụng Z-Score, chọn các giá trị trong khoảng 5% và 95%. Phân phối của các biến sau khi loại bỏ outliers đã về gần với phân phối chuẩn hơn. Ta có thể thực hiện các bước tiếp theo.

f) Xử lý Missing Values

Trong quá trình phân tích dữ liệu, chúng ta nhận thấy một số biến có giá trị bị thiếu với tỷ lệ khác nhau.

- **Các biến có tỷ lệ khuyết trên 60%:**

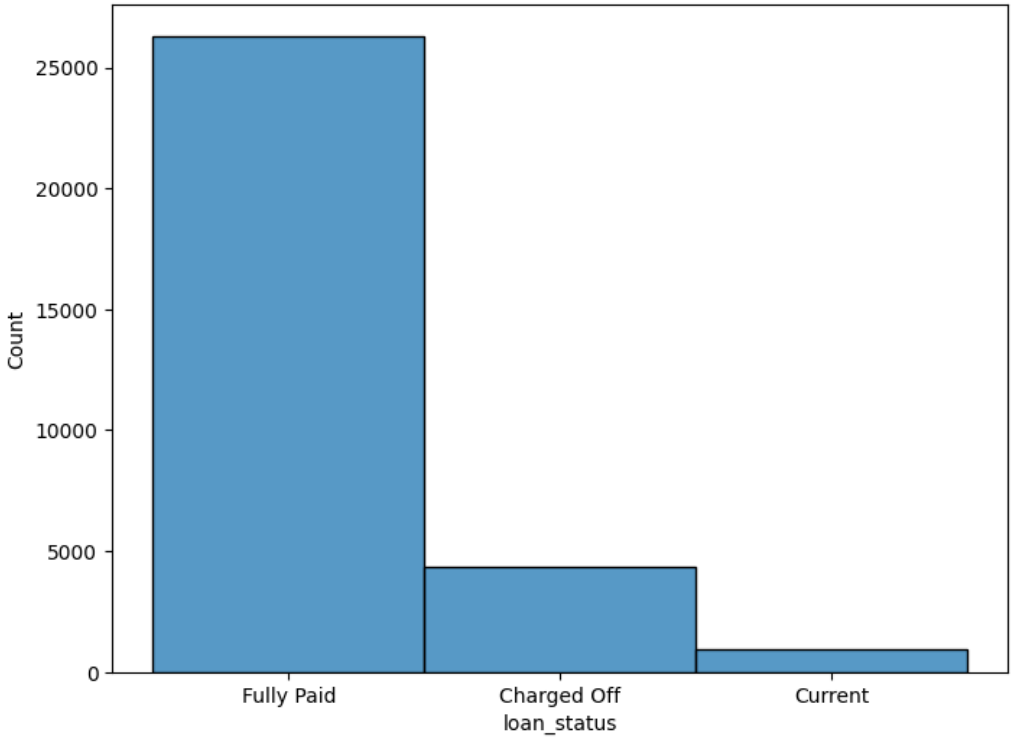
  Hai biến `mths_since_last_delinq` và `mths_since_last_record` có tỷ lệ khuyết trên 60% dữ liệu. Cả hai đều là biến **numerical**, do đó không thể xử lý bằng các phương pháp tương tự như các biến **categorical**. Vì vậy, chúng ta quyết định loại bỏ hai biến này để không ảnh hưởng đến mô hình.

- **Các biến có tỷ lệ khuyết thấp:**

Biến	**Tỷ lệ khuyết (%)**
`last_credit_pull_d`	$\approx 0\%$
`revol_util`	$\approx 0\%$
`last_pymnt_d`	0.12%
`pub_rec_bankruptcies`	1.66%

  Với các biến có tỷ lệ khuyết dữ liệu rất thấp như trên, chúng ta có thể loại bỏ các quan trắc này mà không làm thay đổi đáng kể phân phối của bộ dữ liệu.

2. Exploratory Data Analysis



Hình 1: Phân bố của biến target (`loan_status`)

Ta có thể nhận xét thấy biến mục tiêu `loan_status` có 3 giá trị là:

- **Fully Paid**: Người vay này đã thanh toán toàn bộ khoản vay, bao gồm cả gốc và lãi, theo đúng điều khoản vay.
- **Current**: Người vay đang trả nợ đúng hạn tại thời điểm hiện tại.
- **Charged Off**: Người vay không trả nợ trong một khoảng thời gian dài, nợ xấu dẫn tới người cho vay đã từ bỏ việc thu hồi khoản nợ.

⇒ Để đơn giản hóa quá trình phân loại và xây dựng mô hình, ta sẽ gộp các giá trị Fully Paid và Current thành nhóm Accept (nên cho vay) và nhóm Charged Off thành nhóm Reject (không nên cho vay). Mã hóa Accept là 1 và Reject là 0.

Tuy có tới 53 biến đặc trưng (feature) nhưng chỉ có khoảng 10-11 biến có tương quan đáng chú ý với biến mục tiêu `loan_status` gồm: `recoveries`, `total_rec_prncp`, `total_pymnt`, `total_pymnt_inv`, `last_pymnt_amnt`, `last_pymnt_d_year`, `collection_recovery_fee`, `int_rate`, `sub_grade`, `total_rec_late_fee`, `term`.

Từ đồ thị phân tán sau khi PCA, ta có thể thấy 2 lớp **Accept** và **Reject** của lớp `loan_status` được chia thành 2 phần tách biệt nhau rất rõ ràng ⇒ Đây là một tín hiệu tốt, ta kỳ vọng các mô hình có thể phân loại tốt 2 lớp với bộ dữ liệu này. Thậm chí, các mô hình có thể không cần quá phức tạp để có thể phân loại tốt 2 lớp trên.

3. Model Development and Comparison

Trong bài toán này dùng 3 mô hình là Decision Tree, Logistic Regression và Gradient Boosting

Cả 3 mô hình đều cho ra kết quả rất tốt mặc dù chưa hypertuning, cả 4 metric **accuracy**, **precision**, **recall** và **F1 score** đều trên 0.99 và chỉ có **AUC** của mô hình **Decision Tree** đạt

	Accuracy	Precision	Recall	F1 Score	AUC-ROC	Train Time
Decision Tree	0.993036	0.993007	0.993036	0.993004	0.980469	0.277485
Logistic Regression	0.999842	0.999842	0.999842	0.999842	0.999909	0.180520
Gradient Boosting	0.997942	0.997947	0.997942	0.997936	0.992308	1.965021

dưới 0.99 - là 0.98. Điều này cho thấy các mô hình trên là phù hợp với bộ dữ liệu hiện tại, đồng thời ảnh hưởng của việc không cân bằng giữa 2 lớp cần phân loại gần như không có.

Dù có chênh lệch về metric giữa các mô hình nhưng chỉ rất nhỏ, trong khoảng 0.019 cho **AUC** và khoảng 0.007 cho các metric còn lại.

Nổi trội nhất có thể thấy là mô hình Logistic Regression với sự thời gian train thấp nhất nhưng tất cả các metrics đều cao nhất trong cả 3 mô hình. Điều này giống với những gì ta kỳ vọng khi quan sát bộ dữ liệu ở trên (Bộ dữ liệu đã phân tách khá tốt nên có thể một phương pháp đơn giản, truyền thống cũng có thể đạt được kết quả tốt). ⇒ Ta chọn Logistic Regression để Hyperparameter Tuning, giải thích các biến quan trọng liên quan và xây dựng script dự đoán.

III. Interpretability

1. Math Behind Feature Importance

Logistic Regression

Trong **Logistic Regression**, mức độ quan trọng của một đặc trưng được đo bằng độ lớn tuyệt đối của hệ số β_i trong mô hình. Đặc trưng có hệ số lớn sẽ có ảnh hưởng lớn hơn đến dự đoán của mô hình. Công thức tính feature importance là:

$$\text{Importance of } X_i = |\beta_i|$$

Decision Tree

Trong **Decision Trees**, mức độ quan trọng của một đặc trưng được tính bằng sự giảm đi của impurity (Gini hoặc Entropy) sau mỗi lần phân chia (split). Sự giảm impurity này được cộng dồn qua các node trong cây, và feature importance là tổng sự giảm impurity mà đặc trưng đóng góp. Công thức tính là:

$$\text{Importance of } X_i = \sum_{t \in T} \left(\frac{|t|}{|T|} \times \Delta Gini(t) \right)$$

SHAP trong Gradient Boosting

SHAP (Shapley Additive Explanations) sử dụng **Shapley value** để đo lường đóng góp của mỗi đặc trưng vào dự đoán cuối cùng của mô hình. Shapley value tính toán sự khác biệt trong dự đoán khi một đặc trưng được thêm vào tất cả các tập con của các đặc trưng khác. Công thức tính Shapley value là:

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left[f(S \cup \{i\}) - f(S) \right]$$

2. Explainability

Từ các model đã huấn luyện ta vẽ lên các đồ thị cho thấy Feature Importance và đồ thị SHAP (trong file code), và rút ra nhận xét như sau

Trong mô hình Logistic Regression, các biến có tác động lớn là:

- `funded_amnt`: Có hệ số âm lớn nhất, cho thấy khi số tiền được cấp càng cao, khả năng được duyệt khoản vay sẽ càng giảm.
- `total_rec_prncp`: Có hệ số dương mạnh, cho thấy khi số tiền gốc đã trả càng cao, khả năng được duyệt khoản vay sẽ càng tăng. Điều này đúng với thực tế vì người vay trả nợ càng nhiều thì uy tín của họ càng cao, tỉ lệ được cho vay tiếp càng cao.
- `out_prncp` và `out_prncp_inv`: Hai đặc trưng này tượng trưng cho số tiền gốc còn phải trả cho bên cho vay - cũng có hệ số dương, nghĩa là khi giá trị này tăng lên, khả năng xảy ra sự kiện sẽ tăng. Tuy nhiên, vì bộ dữ liệu này ghi nhận 2 giá trị này bằng 0 khi khách hàng **'Charged Off'** - giống với khi khách hàng **'Fully Paid'** - và chỉ có khách hàng nào đang trả nợ **'Current'** mới có 2 giá trị này lớn hơn 0. Điều này dẫn tới tuy có hệ số cao nhưng 2 biến này chỉ dự đoán được các khách hàng **'Current'**, nếu bên ngoài tệp khách hàng đó 2 biến này sẽ không có ích.
- Các biến `funded_amnt_inv`, `total_rec_int`, `total_pymnt_inv` và `recoveries` có hệ số giảm dần, tuy không lớn nhưng vẫn có ảnh hưởng đáng kể.

Trong mô hình Decision Tree, các biến có tác động lớn là:

- `recoveries`: Là đặc trưng quan trọng nhất, có mức độ ảnh hưởng lớn hơn nhiều so với các đặc trưng còn lại. Tuy nhiên, biến này là số tiền thu hồi được sau khi xóa nợ (hay số tiền bị đòi nợ sau khi trốn nợ), tức là nếu biến này có giá trị dương thì chắc chắn khách hàng này đã trốn nợ - **'Charged Off'**, còn nếu biến này có giá trị bằng 0 thì tỉ lệ rất cao là khách hàng đã trả nợ **'Fully Paid'** hoặc đang trả nợ **'Current'**. Minh chứng là có tới 33477 giá trị 0 và là giá trị duy nhất ở lớp **'Accept'** và chỉ có 1347 giá trị 0 ở lớp textbf'Reject'.
- `total_rec_prncp`: Là đặc trưng quan trọng thứ hai, giống với mô hình Logistic Regression.
- `funded_amnt`: Là đặc trưng quan trọng thứ ba, cũng giống với Logistic Regression.

Trong mô hình Gradient Boosting, các biến có tác động lớn là:

- `total_rec_prncp`: Là đặc trưng quan trọng nhất. Các giá trị SHAP màu đỏ đều là giá trị dương, tức là số tiền gốc đã trả càng lớn thì tỉ lệ duyệt đơn xin vay càng tăng. Điều này là đúng với thực tế và giống với phân tích ở phần Logistic Regression.
- `recoveries`: Quan trọng thứ hai. Các giá trị SHAP màu đỏ đều là giá trị âm, tức là số tiền thu hồi được sau khi xóa nợ càng lớn thì tỉ lệ duyệt đơn xin vay càng giảm. Điều này là đúng và phù hợp với phân tích ở phần Decision Tree.
- `funded_amnt`: Là đặc trưng quan trọng thứ ba. Các giá trị SHAP màu đỏ đều là giá trị âm, tức là số tiền được cấp/xin vay càng lớn thì tỉ lệ duyệt đơn xin vay càng giảm. Điều này là đúng với thực tế và giống với phân tích ở phần Logistic Regression.

Nhìn chung, các biến có tác động lớn nhất xuất hiện trong cả 3 mô hình là `funded_amnt` (số tiền được cấp/xin vay) và `total_rec_prncp` (số tiền gốc đã trả). Trong thực tế, hai yếu tố

trên cũng đóng vai trò quan trọng trong quyết định cho vay, do đó có thể coi các mô hình đã huấn luyện cho ra kết quả hợp lý và có thể giải thích được.

Ngoài ra, để xây dựng một mô hình mới đơn giản hơn, ta sẽ chọn 8 biến có tác động lớn nhất tới mô hình Logistic Regression để huấn luyện mô hình mới, bao gồm: `funded_amnt`, `total_rec_prncp`, `out_prncp`, `out_prncp_inv`, `funded_amnt_inv`, `total_rec_int`, `total_pymnt_inv` và `recoveries`.

IV. Xây dựng mô hình mới

Tạo một bộ dữ liệu mới gồm biến target `loan_status` và 8 biến đặc trưng quan trọng nhất: `funded_amnt`, `total_rec_prncp`, `out_prncp`, `out_prncp_inv`, `funded_amnt_inv`, `total_rec_int`, `total_pymnt_inv`, `recoveries`, `loan_status`

	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC-ROC**
Logistic Regression	0.999842	0.999842	0.999842	0.999842	0.999909
Tuned Logistic Regression	0.997468	0.997475	0.997468	0.997457	0.990533

⇒ Ta thấy việc đơn giản hóa từ mô hình gồm 53 biến thành mô hình 8 biến không ảnh hưởng nhiều đến kết quả của bài toán, các metric chỉ giảm nhẹ khoảng 0.009 ở AUC và 0.002 ở các metric còn lại.