Câu 3 – Thiết kế giải pháp phát hiện khách hàng "thiếu tiền"

Giải pháp được thiết kế theo Cross Industry Standard Process for Data Mining (CRISP-DM). Áp dụng mô hình thiết kế này, người viết đưa ra giải pháp theo mindset: quá trình xây dựng 1 giải pháp/mô hình data-driven là 1 vòng lặp cải tiến liên tục, với các bước lựa chọn feature, lựa chọn model, đánh giá sẽ luôn có điểm để nâng cao hiệu suất của giải pháp(point of improvement). Do đó giải pháp dưới đây sẽ không fix cứng ở 1 feature set hoặc 1 model.

1. Business Understanding

Mục tiêu:

- Xây dựng mô hình phát hiện **khách hàng có khả năng thiếu tiền mặt**, phục vụ cho hoạt động **bán chéo sản phẩm vay**.

Loại bài toán:

- Đây là bài toán **classification** (phân loại nhị phân):
  - **Label đầu ra**: khách hàng **thiếu tiền (1)** hay **không thiếu tiền (0)**
  - Hoặc có thể xây dựng mô hình **dự đoán xác suất (probability)** thiếu tiền để xếp hạng khách hàng theo tiềm năng có thể cross-sale

Vì sao chọn classification (probability):

- Dễ áp dụng trong hệ thống marketing / CRM để gợi ý danh sách khách hàng mục tiêu
- Có thể linh hoạt theo ngưỡng xác suất để tối ưu số lượng khách hàng tiếp cận
- Có thể đánh giá hiệu quả mô hình bằng các chỉ số cụ thể như Precision, Recall

---

2. Data Understanding

Dưới đây là các **biến đầu vào (features)** khả dĩ được trích xuất từ các nhóm dữ liệu đã có:

Nhân khẩu học:

Nhóm người trẻ, có công việc chưa ổn định có thể có nhu cầu vay - Tuổi - Nghề nghiệp - Khu vực địa lý - Trình độ học vấn - Thu nhập khai báo

Tài chính & sản phẩm trong bank:

ví dụ như số dư tài khoản thấp có khả năng cần tiền, sản phẩm lắm giữ nhiều có thể tự tin vay hơn - Số dư trung bình 30 ngày gần nhất, 90 ngày gần nhất - Dư nợ tín dụng hiện tại - Sản phẩm đang nắm giữ (tiền gửi, thẻ tín dụng, khoản vay,…) - Lịch sử trả nợ đúng hạn / trễ hạn (có thể dưới dạng boolean: trả nợ đúng hạn hay không, hoặc xếp hạng mức nợ xấu theo CIC)

Hành vi giao dịch:

Ví dụ: người có số dư tài khoản thấp, lại có tỉ lệ rút tiền hoặc tiền ra cao hơn tỉ lệ nạp tiền/tiền vào, thì có khả năng cho vay cao hơn. - Tổng giá trị tiền vào / tiền ra hàng tháng - Tỷ lệ rút tiền / nạp tiền - Tần suất giao dịch rút tiền - Ngày gần nhất số dư tài khoản $<$ 1 triệu VND

Hành vi trên app:

- Lượt đăng nhập gần đây
- Số lần click vào trang vay vốn
- Thời gian dừng ở module vay / tín dụng
- Tìm kiếm các từ khóa như "vay", "tín dụng", "hỗ trợ tiền mặt"

Việc lựa chọn các biến cũng như quá trình chuẩn bị dữ liệu ở bước sau sẽ phụ thuộc vào mức độ available của dữ liệu và model sẽ sử dụng, và luôn có thể quay lại bước này để thực hiện lại feature selection nhằm nâng cao hiệu suất mô hình.

3. Data Preparation

Một số pre-processing dữ liệu cần thiết:

- **Xử lý missing value**: điền giá trị trung bình, median, hoặc gán nhãn "unknown", hoặc Multiple Imputing tùy theo loại data
- **Biến đổi định dạng**: chuyển đổi dữ liệu thời gian (timestamp), tính các biến động trong khoảng 30/90 ngày
- **Chuẩn hóa biến số liên tục** (StandardScaler, MinMaxScaler)
- **Encoding biến phân loại**: One-Hot Encoding hoặc Target Encoding cho các cột như nghề nghiệp, khu vực
- **Tạo biến tương tác**: ví dụ `tỷ lệ chi tiêu / thu nhập`, `rút tiền / số dư`
- **Giảm chiều**: PCA nếu số feature nhiều và mô hình cần đơn giản

---

4. Modeling

Dưới đây là 1 số gợi ý các model phù hợp cho bài toán classification nhằm dự đoán khả năng **khách hàng cần vay tiền**, được xếp theo thứ tự từ đơn giản đến phức tạp.

Thuật toán tuyến tính

Tiêu biểu là **Logistic Regression**. - Ưu điểm: đơn giản, nhanh, dễ triển khai, dễ hiểu, dễ giải thích rõ từng hệ số ảnh hưởng đến xác suất thiếu tiền. - Nhược điểm: phải giả định là xác suất có thể bán được sản phẩm vay cho khách hàng là tuyến tính với các biến đầu vào, do đó kém hiệu quả với dữ liệu phi tuyến. Ngoài ra, phải xử lí pre-processing để impute missing data

Với các ưu điểm của mình, model logistic regression có thể được training ở bước đầu nhằm tạo baseline model để so sánh hiệu quả cho các model phức tạp hơn

---

Ensemble Models

Là nhóm các model phát triển dựa trên việc kết hợp nhiều decision tree model, với ưu điểm chung là hiểu quả với dữ liệu phi tuyết tính và không cần xử lí missing data

Bagging Là nhóm các model được thực hiện bằng cách chạy song song decision tree trên nhiều sample con của dataset, các decision tree này sẽ cùng vote để có được giá trị dự đoán cuối cùng cho đầu ra. Tiêu biểu của nhóm model này là **Random Forest**

- Ưu điểm: kháng overfitting tốt, hoạt động hiệu quả trên nhiều loại dữ liệu, robust với outliers.
- Nhược điểm: model yêu cầu khả năng tính toán của máy tính cao do chạy song song nhiều model, ít trực quan hóa mối quan hệ biến – mục tiêu hơn so với các model tuyến tính do mỗi tree có 1 tệp sample khác nhau và đánh giá tầm quan trọng của feature khác nhau -> khó diễn đạt

---

Boosting (Sequential Weak Learners) Như tên gọi, đây là nhóm ensemble model được xây dựng bằng cách chạy nhiều lần, lần lượt decision tree trên training set, với mỗi decision tree sau sẽ tập trung vào các weak point (các entry bị dự đoán sai và tăng weight để các decision tree sau coi trọng các entry này hơn). Một số model tiêu biểu có thể ứng dụng là:

- **XGBoost**

  - Ưu điểm: hiệu năng cao, kiểm soát overfitting tốt, hỗ trợ regularization, xử lý missing value tự nhiên.
  - Nhược điểm: dễ overfit với dữ liệu nhỏ, cần tuning nhiều.
- **CatBoost**
  - Ưu điểm: không cần encode biến phân loại, hoạt động tốt với tập nhỏ và không cần tuning nhiều.
  - Nhược điểm: ít phổ biến hơn, cần GPU cho huấn luyện nhanh.

Neural Networks (Deep Learning)

Neural Network là mô hình học sâu mô phỏng cấu trúc não bộ, gồm nhiều lớp liên kết (layers) giúp học các mối quan hệ phức tạp, đặc biệt tốt trong việc mô hình hóa các mối quan hệ phi tuyến. Ưu điểm chung của nhóm model này là: Khả năng tiếp nhận các biến dữ liệu phi tuyến, tự động học và tìm được đặc trưng ẩn (hidden features), Linh hoạt trong việc mở rộng và kết hợp với các kỹ thuật embedding, attention, … Tuy nhiên, NN có nhược điểm sau: Yêu cầu dữ liệu lớn để học tốt, cần tài nguyên máy tính để có thể training. Cần data scientist phải có hiểu biết về tuning hyperparameter và xử lý overfitting. Khó giải thích so với các mô hình truyền thống do có các lớp đặc trưng ẩn mà mô hình tự khám phá (hidden layer)

Một số model ANN tiêu biểu là - **Feedforward Neural Network (MLP)**
Phù hợp với dữ liệu dạng bảng khi cần khai thác quan hệ phi tuyến

- **TabNet**
  Mô hình deep learning chuyên biệt cho dữ liệu dạng bảng (tabular), sử dụng attention để tăng khả năng giải thích.
- **Entity Embedding + MLP**
  → Dùng khi có nhiều biến phân loại – embedding giúp mô hình hiểu các mối quan hệ tiềm ẩn giữa các giá trị phân loại, ví dụ: ngành nghề, khu vực, sản phẩm.

Dù lựa chọn model nào, thì quá trình tạo dựng giải pháp là 1 vòng lặp của việc thử nghiệm, đánh giá, so sánh giữa các model cũng như các yếu tố như thời gian, chi phí train model, khả năng giải thích cho end-user cũng như deploy vào hệ thống

5. Evaluation (Đánh giá mô hình)

Vì đây là bài toán phân loại nhị phân với mục tiêu **xác suất** (probability classifier), nên các chỉ số đánh giá cần phản ánh chất lượng phân loại theo xác suất.

- **Accuracy (Độ chính xác)**
  Tỷ lệ dự đoán đúng trên tổng số mẫu. Phù hợp khi dữ liệu **cân bằng**.

Tuy nhiên, dễ gây hiểu nhầm nếu tập dữ liệu **mất cân bằng** (ví dụ: 95% khách hàng không cần vay).

- **Precision (Độ chính xác trong các giá trị positive)**
  Trong các khách hàng được dự đoán là thiếu tiền (positive), có bao nhiêu là đúng (true positive). Quan trọng trong việc giảm **chi phí cho mỗi positive case** (ví dụ: marketing nhắm sai người).
- **Recall (Tỷ lệ phát hiện)** Trong số các khách hàng có thể cho vay, mô hình phát hiện được bao nhiêu. Chỉ tiêu này càng cao, độ phủ của mô hình càng lớn, ko bỏ sót các false negative (khách hàng có nhu cầu vay nhưng model nhận định là không).
- **F1-Score**
  Trung bình điều hòa giữa 2 chỉ Precision và Recall – giúp cân bằng các yếu tố liên quan đến 2 chỉ số. Ví dụ như việc tối ưu Recall bằng cách tăng positive instance, dẫn đến phình to chi phí marketing, và ngược lại tối ưu Precision (đồng thời tối ưu chi phí marketing) có thể làm giảm positive instance, gây bỏ sót các khách hàng tiềm năng
- **ROC-AUC (Area Under ROC Curve)**
  Đo lường tương quan giữa Recall (tỉ lệ dự đoán đúng trên tổng các trường hợp positive) và tỉ lệ cảnh báo sai (số lượt dự đoán positive sai trên tổng negative). Tỉ lệ tốt nhất AUC = 1 với ý nghĩa Recall = 1 (tỉ lệ dự đoán đúng 100%) và False positive rate = 0 (tỉ lệ dự đoán sai là 0). Hệ số AUC ngưỡng dưới là 0.5 (2 tỉ lệ này bằng nhau, cứ 1 lần đoán đúng thì 1 lần đoán sai, ngang rate tugn đôngf xu)
- **Log Loss (Binary Cross Entropy)**
  Đo độ lệch giữa xác suất dự đoán và nhãn thực tế. Phù hợp cho bài toán xác suất vì phản ánh **chất lượng xác suất** hơn là nhãn cứng.

Ngoài ra còn có các chỉ tiêu về thời gian training, chi phí computing, độ phức tạp trong deploy,... Các chỉ số trên sẽ được so sánh giữa các model - feature set nhằm đánh giá hiệu suất của các model để tiến đến model tối ưu nhất —

6. Deploy (Triển khai mô hình)

Các bước triển khai điển hình:

- Model packaging
- Triển khai bằng API hoặc batch job
- Kết nối với hệ thống kinh doanh: Push kết quả dự đoán (probability) về CRM, dashboard nội bộ hoặc gửi danh sách cho đội kinh doanh.
- Giám sát hiệu suất mô hình (Monitoring), đánh giá vào cảnh báo khi performance giảm sút.
- Cập nhật & retrain (thủ công hoặc tự động với pipeline)

4. Doanh số chi tiêu thẻ tháng này giảm mạnh 20% — Giải pháp từ góc độ Data Scientist

Phân tích tình huống:

Doanh số chi tiêu thẻ tụt mạnh có thể bắt nguồn từ thay đổi hành vi khách hàng, vấn đề nội bộ, hoặc yếu tố thị trường. Cần phân tích nguyên nhân để từ đó đưa ra các hành động cụ thể.

Đặt ra hypothesis:

- Khách hàng có xu hướng giảm chi tiêu cá nhân do yếu tố kinh tế vĩ mô hoặc thu nhập giảm.
- Tăng trưởng sử dụng các ví điện tử và nền tảng thanh toán khác thay thế thẻ ngân hàng.
- Khách hàng gặp trải nghiệm không tốt với app hoặc dịch vụ của ngân hàng nên giảm sử dụng thẻ.
- Có thay đổi chính sách hạn mức hoặc phí liên quan đến thẻ khiến khách hàng ít sử dụng hơn.
- Các chiến dịch khuyến mãi hoặc ưu đãi từ ngân hàng dừng lại làm giảm động lực chi tiêu.

Kiểm chứng hypothesis dựa trên data

- Khách hàng có xu hướng giảm chi tiêu: Kiểm tra lịch sử giao dịch thẻ của khách hàng, so sánh timeline với thay đổi trong chỉ số giá CPI hoặc các chỉ số đánh giá nền kinh tế, dòng tiền vào của tài khoản khách hàng.
- Tăng trưởng sử dụng ví điện tử/nền tảng thanh toán/ngân hàng khác: Kiểm tra thay đổi về số lượng và giá trị giao dịch của các giao dịch chuyển khoản/nạp tiền vào ví điện tử, có thể đi kèm với competitor analysis (các chương trình khuyến mãi của đối thủ gần đây làm breakpoint)
- Trải nghiệm không tốt với app: log hành vi app, runtime thời gian thanh toán online bằng thẻ tín dụng, số lần lỗi app, số khiếu nại của khách hàng.
- Thay đổi chính sách phí thẻ/hạn mức/khuyến mãi: Kiểm tra lịch sử giao dịch thẻ với với thời điểm thay đổi các chính sách là breakpoint.

Một số nhu cầu kinh doanh có thể giải quyết bằng data science:

1. **Phân khúc khách hàng có doanh số chi tiêu giảm rõ rệt**
   - **Bài toán DS**: Clustering khách hàng theo hành vi chi tiêu qua các tháng.
   - **Feature**: tổng chi tiêu, số lần giao dịch, loại giao dịch, ngành hàng sử dụng.
   - **Giải pháp**: Dùng KMeans hoặc DBSCAN để phân cụm, xác định nhóm sụt giảm.

- **Mục tiêu cuối**: Xác định nhóm khách hàng có đặc điểm đang và sẽ giảm chi tiêu, thực hiện các campaign marketing về lợi ích của việc dùng thẻ.

2. **Dự báo doanh số chi tiêu và cảnh báo sớm sụt giảm**
   - **Bài toán DS**: Time-series forecasting.
   - **Feature**: lịch sử chi tiêu thẻ tín dụng, biến động chi tiêu giữa các tháng và các lần thay đổi chính sách
   - **Giải pháp**: ARIMA, LSTM cho time-series
   - **Mục tiêu cuối**: Lường trước sụt giảm có thể xảy ra khi thay đổi chính sách, từ đó cân nhắc chiến lược.
3. **Phân tích nguyên nhân sụt giảm**
   - **Bài toán DS**: Causal inference hoặc feature importance analysis.
   - **Feature**: thời gian sử dụng thẻ, thay đổi chính sách, tương tác CSKH.
   - **Giải pháp**: Xử lý bằng model explainability (SHAP, LIME), kiểm định thống kê.

---

5. Mục tiêu tăng huy động vốn thêm 20% — Giải pháp từ góc độ Data Scientist

Phân tích tình huống:

Mục tiêu tăng 20% tổng huy động vốn trong năm cần hỗ trợ bằng các chiến lược cụ thể về khách hàng, sản phẩm, và kênh phân phối.

Một số nghiệp vụ kinh doanh có thể được giải quyết bằng DS:

1. **Tìm tập khách hàng tiềm năng có khả năng gửi thêm tiền** Tương tụ như bài toán tìm khách hàng cho vay, nhưng với nhóm khách hàng có thể bán thêm sản phẩm huy động vốn
   - **Bài toán DS**: classification – likelihood of deposit.
   - **Feature**: số dư hiện tại, lịch sử giao dịch gửi tiền, thu nhập, độ tuổi, các sản phẩm đang sở hữu
   - **Giải pháp**: Logistic Regression, Random Forest, XGBoost, hoặc ANN
2. **Phân nhóm khách hàng để đưa ra chiến lược gói sản phẩm huy động vốn** Nhằm định hình đặc điểm của nhóm khách hàng, từ đó đưa ra chiến lược bán hàng, marketing, gói sản phẩm phù hợp với từng segment.
   - **Bài toán DS**: Customer segmentation.
   - **Feature**: mục tiêu tài chính, thu nhập, sản phẩm nắm giữ.
   - **Giải pháp**: KMeans hoặc Hierarchical Clustering để phân nhóm và đưa ra chính sách phù hợp.

3. **Tối ưu hóa kênh phân phối sản phẩm huy động** Cải tiến các model đang có nhằm nâng cao trải nghiệm của khách hàng với các sản phẩm huy động vốn online.
    - **Bài toán DS**: Recommendation system hoặc uplift modeling.
    - **Feature**: hành vi trên app, lịch sử sản phẩm nắm giữ, thu nhập
    - **Giải pháp**: Collaborative filtering hoặc uplift tree modeling.
4. **Dự báo huy động theo khu vực hoặc nhóm khách hàng** Mục tiêu là dự đoán được nhóm khách hàng, khu vực khách hàng sẽ có tăng trưởng về vốn cá nhân trong thời gian tới, từ đó thúc đẩy phòng kinh doanh tiếp cận các khách hàng này.
    - **Bài toán DS**: Forecasting.
    - **Feature**: dữ liệu lịch sử huy động theo phân khúc/khu vực/thời gian, lịch sử thu nhập
    - **Giải pháp**: ARIMA, Prophet hoặc LSTM.

“‘