AI model validation report_GXS Agent Requirement

1. Project Background & Objectives

1.1 Background

- The current MVR "URN00001 – FlexiLoan & FlexiCard Application Scorecard v3.0" includes the following in **Section 3.3 Model Performance**:
  - 3.3.1 Information Value (IV)
  - 3.3.2 Discriminatory Power (AUROC / AR, by segment, by product, by monthly cohort)
  - 3.3.3 Risk Ranking (Bad rate & customer distribution by risk band)
- This content is currently handled manually by model development/validation colleagues: manual data extraction → manual code/notebook execution → manual compilation of tables and text → copy-pasting into the Word report.

1.2 Objectives

☐ The goal of this project is to build a **Model Validation Agent**, given configurations and data, **fully automates the calculation and report generation for Section 3.3 Model Performance**, including:

  a. **Automatically loading user-uploaded training and validation samples** (refer to `training_data_20251007_v4.xlsx`).

  b. **Reusing the feature engineering, model, and metric calculation logic from the existing model development code** `mtb_scorecard_3.pkl`

(Note: If the agent studio cannot read the model file format, this logic needs to be embedded into the chatflow using predefined metric calculation code).

  c. **Automatically calculating**:
    - IV for each segment (MTB / YNTB) for both training & combined datasets.
    - AUROC & AR (for Train/Test/Combined sets, for FlexiLoan and FlexiCard).
    - Monthly cohort AUROC (for FlexiLoan & FlexiCard).
    - Risk Ranking tables by risk band (including population percentage & bad rate).

   d. **Automatically populating a standardized template based on the generated metrics, producing Section 3.3 content ready for direct insertion into the MVR** (including tables + textual descriptions + charts).

- ☐ The system needs to support a **conversational, multimodal Model Validation Assistant**, enabling users (Data Scientists, Risk Control, PMs) to interact with the Agent via natural language to complete the following tasks:
   a. Professional model Q&A
   b. Triggering metric calculations (automated model performance runs)
   c. Triggering report generation
   d. Multimodal input analysis (text, tables, charts)
   e. Responding to open-ended / exploratory questions

1.3 Scope (Phase 1)

Focuses only on Section 3.3 Model Performance within the MVR document.

1. **QA Engine (Question-Answering Engine)**
   - Input: User's natural language questions.
   - Process: Parsing → Constructing answers from cached metric results/model configurations.
2. **Metric Engine (Metric Calculation Engine)**
   - Executes the existing performance pipeline.
   - Supported Customer Segments:
     - MTB (Mature to Bureau)
     - YNTB (Young/No to Bureau)
   - Supported Products:
     - FlexiLoan (development sample + monitoring sample)
     - FlexiCard (out-of-time validation sample)
3. **Report Generator (Report Text Generation + Template Filling)**
   - Automatically produces:
     - 3.3.1 IV tables + textual description.
     - 3.3.2 AUROC/AR tables (overall, by segment, by monthly cohort) + textual description.
     - 3.3.3 Risk Ranking tables + key descriptions (including rank order break detection).
     - Necessary charts (ROC curves, cohort AUROC trends, risk band bad rate charts, etc.) as PNG files.

        - A structured text fragment (Markdown / HTML / Docx) that can be directly pasted into the corresponding Section 3.3 of the [Approved] document.

4. **Multimodal Parser (Multimodal Parsing)**
    - Supports users uploading files in different formats: Excel, PNG, Docx, CSV, etc.

2. Structured Input and Output Design

2.1 Input (User Input Items)

1. Training Sample Input: `training_data_20251007_v4.xlsx` (Training Data)

    Contains fields (Key field requirements):

    - Customer & Loan Identifiers
        - `customer_id`
        - `application_id`
    - Segment Identifier
        - `segment` (Values: MTB / YNTB)
    - Product Type
        - `product_type` (FlexiLoan / FlexiCard)
    - Time Field
        - `origination_month` (for cohort analysis, format yyyyMM/date)
    - Target Variable (Label)
        - `bad_30dpd_15_25m` (proxy default flag)
    - Feature Columns (consistent with features in the `s02_model_training-MRM` model file)
        - Examples: `property_type`, `a_latest_cpf_cntri`, `POSTBUREAU_NUM_ENQ_UNSEC_NEW_APP_L12M`, `CUO_CUL`, `PAYLATER_EVER_DPD_1_OR_CB_DPD30_L6M_CNT`, `POSTBUREAU_AGE`, `k_mob_unsec`, `r_unsecbal_l3m_l6m`, `FE_CC_USAGE_PATTERN_GRAB`, `r_ibunsec`, `PAYLATER_DSR_LIMIT`, `num_enq_3_6`, `POSTBUREAU_ACCOUNT_COUNT`, and corresponding features for YNTB.
2. Validation Sample Input: `df_base_credit_monitoring` (Validation / Monitoring Sample)
    - Also contains the fields listed above, but:
        - Spans a longer time period (Jan 2023 – Aug 2025).
        - Includes both FlexiLoan + FlexiCard.
    - Requires additional fields:

    - `product_type` (FlexiLoan / FlexiCard)
    - If a 90dpd flag is available, it can be recorded as `bad_90dpd_12m` (reserved for future expansion).

(Note: Implement schema validation during development – if key columns are missing, the agent must fail fast and provide clear error messages.)

2.2 Output

2.1 Matrices Calculation Output（Dashboard）

a. IV

   i. outputs/MTB_features_IV.csv

   ii. outputs/YNB_features_IV.csv

   Field example：

   - variable_name
   - iv_training / iv_combined
   - iv_flag_below_threshold（True/False）

b. AUROC/AR

   i. outputs/MTB_cohort_auroc.csv

   ii. outputs/YNB_cohort_auroc.csv

   Field example：

   - set_type（Train/Test/Combined）
   - cohort_month
   - total_obs
   - total_bad
   - bad_rate
   - AUROC / AR

c. Risk Ranking

   i. outputs/MTB_risk ranking.csv

   ii. outputs/YNB_risk ranking.csv

   Field example：

   - band_segments
   - score_min / score_max or pd_min / pd_max
   - population_count / population_pct

        - bad_count
        - bad_rate
        - cumulative_population_pct
        - cumulative_bad_pct
        - rank_order_break_flag（True/False）

2.2 Chart Output

a. ROC Curve：
    i. plots/urn00001_v3_roc_flexiloan_MTB.png
    ii. plots/urn00001_v3_roc_flexiloan_YNTB.png
    iii. plots/urn00001_v3_roc_flexicard_MTB.png
    iv. plots/urn00001_v3_roc_flexicard_YNTB.png
b. Cohort AUROC Trend：
    i. plots/urn00001_v3_cohort_auc_flexiloan_MTB.png
    ii. plots/urn00001_v3_cohort_auc_flexiloan_YNTB.png
    iii. plots/urn00001_v3_cohort_auc_flexicard_MTB.png
    iv. plots/urn00001_v3_cohort_auc_flexicard_YNTB.png
c. Risk Ranking **Bad debt ratio bar chart**（option）。

2.3 Report Chapter Output

Paste-ready Text Version:

- `outputs/urn00001_v3_mvr_3_3.md` or `.docx`
- The content structure must be consistent with the existing Section 3.3 of the MVR
- The text should follow a template + data-driven approach, similar to:

"MRM finds that most of the selected features are passing the IV threshold of 0.02, except for ... in MTB and ... in YNTB. However, all selected features passed the threshold in the training set, and the model's overall performance is strong." The feature names & values should be automatically replaced based on the latest calculation results.

3. Workflow Design

Step 0. Initiation & Run

User trigger (via natural language or command).

Step 1. Load Configuration & Data

a. Load data from user-uploaded files.

b. Validate the existence of required fields. If any are missing, throw an exception and provide a detailed explanation in the logs.

c. Perform type validation based on the data dictionary (refer to `training_data_20251007_v4`):

    i. Check if numeric columns can be converted to float.

    ii. Check if date columns can be parsed as dates.

d. Apply exclusion rules (if any):

    i. Reuse filtering logic from `s02_model_training-MRM` (e.g., filter out unsuccessful applications, no drawdowns, etc.).

Step 2. Feature Engineering & Model Loading

a. **Feature Engineering**

- Ensure consistency with the features used in the development model:
    - Identical binning & WOE transformation.
    - Identical missing value handling.
    - Identical variable filtering.

b. **Model Loading / Reconstruction**

    i. Option 1 (if PKL file upload is possible): Load the saved model object (e.g., `models/mtb_scorecard_3.pkl`).

    ii. Option 2: Reconstruct the Logistic Regression model in code using coefficient fields & intercept (solution TBD).

    Coefficients can be read and solidified from the model development documentation.

Step 3. Metric Calculation

Use the predefined metric calculation logic within the chatflow, with a unified interface. Refer to the appendix document for calculation code. [`model_performance_metrics_code`]

3.1 IV Calculation (3.3.1)

a. Calculate separately for each segment:

    i. Training set: `compute_iv(df_train_segment)`

    ii. Combined set: `compute_iv(df_combined_segment)`

b. Output: Structure as defined in section 2.2 Output.

    i. (Note: Flag low IV features based on thresholds: green, yellow, red).

3.2 Discriminatory Power (AUROC & AR) (3.3.2)

3.2.1 FlexiLoan

Use data where `product_type=FlexiLoan`.

  - Train/Test Split Logic: Reuse the original training code's split (or directly read the existing identifier column `is_train`).
  - For MTB / YNTB: Calculate AUROC & AR for Train/Test/Combined using `compute_auc_ar(df, label_col, score_col)`.
  - Output: Structure as defined in the AUC/AR section of 2.2 Output.

3.2.2 FlexiCard

Use monitoring data where `product_type=FlexiCard`.

Same process as 3.2.1 above.

3.2.3 Monthly Cohort AUROC (FlexiLoan & FlexiCard)

  - Group by: `cohort_month` + `segment`.
  - For each group: If `total_bad` or `total_good` samples are too few (e.g., < 30), record AUROC as NA.
  - Output populated into the corresponding cohort metrics table.

(Note: Mark cohort AUROC level based on thresholds: green, yellow, red).

3.3 Risk Ranking (3.3.3)

- Based on model output PD or score: Sort separately for each segment.
- Rules:
  - Minimum 5% population per band.
  - Maximum band count of 10 (configurable).
- Output table fields as defined in the structure in section 2.2 Output.

Step 4. Generate Structured Files

Step 5. Interaction – Answer User Questions

Step 6. Generate Report

4. User Interaction Design

1 Model Q&A (QA Interaction)

Scenario 1: User inquires about model performance (already calculated)

User:

> How has the recent YNTB segment performance been? Has the AUROC dropped?

Agent（Intent: ModelPerformanceQuery）

> The AUROC for the YNTB Segment in the last three months was 0.661, 0.654, and 0.658 respectively. The month 2025-06 fell into the Amber zone, while the other months were in the Green. The overall trend is stable, with no significant decline observed.

Scenario 2: User inquires about feature stability

User:

> Which features have the most severe drift in recent cohorts?

Agent（Intent: ModelPerformanceQuery）

> In the last three cohorts, the PSI for `POSTBUREAU_AGE` and `PAYLATER_EVER_DPD_1_OR_CB_DPD30_L6M_CNT` were both > 0.1, indicating moderate drift; it is recommended to focus on changes in the sample structure.

Scenario 3: User inquires about risk ranking (rank order break)

User:

> Is there any rank order break in the MTB segment?

Agent（Intent: ModelPerformanceQuery）

> A slight rank order break was detected in the MTB Segment at Band 7 (bad rate 2.91% > Band 6's 2.75%). However, the sample size is small, and the impact is acceptable.

---

2 Triggering Metric Calculation (Actionable Commands)

Scenario 4: User triggers one-click model performance run

User:

> Please re-run the model performance for URN00001

Agent（Intent: RunPerformanceComputation）

> OK, I am loading the configuration `urn00001_v3_model_perf.yaml` and calculating all metrics (IV / AUROC / AR / Cohort / Risk Ranking)...
>
> Calculation completed. Results have been saved to the directory: `/outputs/20250215_1500/`. Would you like me to generate the 3.3 report?

Scenario 5: Run cohort AUROC (with parameter parsing)

User:

> Generate AUROC for FlexiCard for the last 6 months.

Agent（Intent：RunCohortPerformance + ParameterExtraction）

> Identified Product=FlexiCard, Window=Last 6 months.
>
> Cohort AUROC has been calculated for MTB/YNTB Segments respectively, and trend charts have been generated. Charts have been output to `/plots/cohort_auc_flexicard.png`.

---

3 Triggering Report Generation (Report Mode)

Scenario 6: One-click generation of Section 3.3 report

User：

> Please generate the report fragment for section 3.3, in Word format.

Agent

> The report fragment has been generated using the latest metrics and populating the template. The document has been output to: `/reports/urn00001_3_3_20250215.docx`.
>
> You can copy it directly into the MVR document."

---

4 Multimodal Input (Multimodal)

Scenario 7: User uploads Excel for Agent analysis

User：

> (Uploads Excel file)
>
> This is the bad-rate data provided by our external partner. Can you compare it with the cohort's bad-rate calculated by the system to see if they are consistent?

Agent（Intent：FileUpload + DataComparison）

> Identified the uploaded file as Excel, containing cohort and bad-rate fields. I have completed the comparison:
>
> - 2025-06: Partner 3.21%, System 3.18% (Consistent)
> - 2025-07: Partner 3.31%, System 3.85% (Slight discrepancy, need to confirm definitions)
>
> Do you need me to further analyze the source of the discrepancy?"

Scenario 8: User uploads an image (PNG chart)

User：

> (Uploads PNG)
>
> Can you check if there are any issues with this trend?

Agent（Intent：ImageAnalysis）

> The chart shows that the bad-rate for the MTB Segment has been continuously rising over the last 7 months (approx. 2.1% → 3.2%), with increased volatility. It is recommended to further check model stability by combining it with cohort AUROC analysis.

---

5 Exploratory Questions (General Q&A)

Scenario 9: User asks an open-ended question

User：

> Is the model's performance degrading severely? Does it need retraining?

Agent（Intent：GeneralOpenQ）

> Based on current metrics:
>
> - AUROC remains above the Green/Amber thresholds.
> - No significant Feature Stability Drift.
> - Only minor breaks in Risk Ranking.
>
> Retraining is not currently necessary. Recommend continued monitoring of drivetrain features and structural changes in the YNTB segment.

5. Project plan

Content	Dependencies	Key Deliverables
1 Requirement & Data Alignment		

1.1 Indicator definition	1. Input data(training data\df_base_credit_monitoring\base_by_custfac_w_waterfall_bad_20250923_v4)
	2. Data dictionary
		Data dictionary;
		Indicator list;
		Input-output schema

1.2 Data mapping completed		
1.3 Requirements clarification		Requirements document（lark）
2 Analytics Agent		

2.1 Core metric engine	indicator definitions，
	python scripts
		IV/PSI/AR/AUC/KS modules;
		agent_v1;
		unit tests

2.2 Segment metrics validated		
2.3 Output format aligned		
2.4 Rating rules complete	Metric outputs from Analytics Agent; MRM thresholds	Green/Amber/Red rule engine;
2.5 Override mechanism implemented		
3 Dialogue Platform		
3.1 Parameter mapping stable	Analytics Agent API, Parameter schema	Dialogue interface; Query→parameter parser
3.2 Query success rate		
note：phase connection with GXS *Periodic delivery, GXS tests whether the results of the interactive dialogue are what they need, such as the acc*		
4 Report Writing Agent		
4.1 Auto-report v1	Assessment outputs; Report template	Model Performance chapter draft
4.2 Tone/style aligned		
4.3 95% completeness		

Appendix

📍 **Matrices Calculation logic /notebook**

W model_performance_metrics_code.docx

76.69KB

📍 **Model validation report-GXS**

W [Approved] URN00001 - FlexiLoan & FlexiCard Application⋯

1.93MB