Credit Risk Scoring Extension: PD Modelling, IFRS 9 Staging and Scenario-Weighted ECL

Ashara Dangallage Don

1 What this project does

This project builds a simple but complete credit risk workflow using public LendingClub loan data [3]. The goal was not to build a production banking model, but to put together a clean, end-to-end pipeline that covers the main moving parts of an IFRS 9-style expected credit loss framework [1]:

- a **PD model** (probability of default),
- a **SICR staging rule** (to separate Stage 1, Stage 2 and Stage 3),
- basic **EAD** and **LGD** assumptions,
- and a final **scenario-weighted ECL** output.

I treated this as a technical implementation exercise but also as an extension of my works with a ML-based credit scoring model.

2 Dataset and setup

The source dataset is the LendingClub accepted loans file hosted on Kaggle [3]. However, to make the workflow manageable, I reduced the raw dataset to a modelling subset, created a default flag and downsampled non-defaults for faster iteration. A `sample_weight` column was kept throughout the pipeline so that later summaries could still reflect the implied portfolio mix.

After applying the maturity cutoff used for modelling, the working sample contained:

- 541,736 loans
- issue dates from **2011-01-01** to **2016-12-01**

One important note: the target used here is a **default proxy** based on final loan status (charged off / default related statuses), not a clean point-in-time 12-month default observation. That matters for interpretation later, especially when looking at lifetime PD and ECL.

3 Stage 1: Data preparation

The first stage handles the fundamental data preparation:

- select a compact set of origination variables,
- parse and sort by `issue_d`,
- create the default indicator from loan status,
- filter to post-2011 vintages,
- downsample non-defaults to reduce runtime,
- store `sample_weight` so portfolio-level totals can be recovered later.

This step is simple, but it matters because it sets up the rest of the pipeline. The downsampling makes the notebooks practical to run, while the weights stop the later results from becoming misleading.

4 Stage 2: PD modelling

4.1 Model structure

The PD model is a logistic regression built with a scikit-learn pipeline [2]. I kept it deliberately lightweight, but made the preprocessing fairly disciplined so the whole thing stays reproducible.

The main input features (v1 feature set) were:

- `loan_amnt`
- `term`
- `int_rate`
- `sub_grade`
- `purpose`
- `annual_inc`
- `emp_length`
- `home_ownership`
- `verification_status`
- `dti`
- `delinq_2yrs`
- `open_acc`
- `pub_rec`
- `revol_util`
- `revol_credit`
- `loan_to_income`

The preprocessing pipeline included:

- median imputation for numeric variables,
- most-frequent imputation for categorical variables,
- standard scaling for numeric variables,
- one-hot encoding for nominal categories,
- ordinal encoding for ordered categories (`sub_grade`, `emp_length`, `term`),
- quantile binning for variables with clearer monotonic risk patterns.

This kept the model easy to run and easy to explain, while still handling missing values and mixed variable types properly.

4.2 Validation design

The split was time-based rather than random. I also applied a 24-month maturity cutoff before modelling to reduce right-censoring caused by the status-based label.

That choice mattered a lot. Without the cutoff, the most recent loans looked artificially safe because they had not had enough time to move into a charged-off/default status. Cutting off the final 24 months made the train/validation/test periods much more stable.

4.3 Operating threshold

Instead of using a default threshold of 0.5, I chose the operating point under a simple policy rule:

$$\text{weighted FPR} \leq 0.20$$

This gave a threshold that is easier to justify than a standard 0.5 cutoff.

4.4 PD model results

Three logistic variants were tested: a plain L2 model, a class-weighted L2 model and an elastic-net version. The plain **BASELINE_L2** model came out marginally ahead on weighted validation PR-AUC, so that became the main model.

Table 1: Weighted PD model results

Model / Split	ROC-AUC	PR-AUC	Threshold
BASELINE_L2 (Validation)	0.7015	0.2937	0.2219
BASELINE_L2 (Test)	0.6790	0.2666	0.2219
CALIBRATED (Validation)	0.7015	0.2934	0.2211
CALIBRATED (Test)	0.6790	0.2664	0.2211

At the chosen operating point, the calibrated model produced:

- weighted validation ROC-AUC: **0.7015**
- weighted validation PR-AUC: **0.2934**
- weighted test ROC-AUC: **0.6790**
- weighted test PR-AUC: **0.2664**

The validation confusion matrix at the calibrated threshold was:

$$\begin{bmatrix} 32196 & 8048 \\ 13555 & 11210 \end{bmatrix}$$

That translates to a class-1 precision of about **0.58** and recall of about **0.45** on validation.

4.5 Calibration

I tested a calibrated version of the PD model as well. In practice, calibration did not change much. The weighted Brier score on validation was:

- uncalibrated: **0.12244**

The calibrated and uncalibrated reliability plots were very similar, so the conclusion was that calibration did not particularly improve probability quality in this setup, even though it remained available as an option.

4.6 Key risk drivers

Permutation importance on the validation sample showed a very clear ranking. The strongest driver by a large margin was `sub_grade` in its ordinal form. After that, the most useful variables were `loan_to_income`, `int_rate`, `loan_amnt` and `home_ownership`. Secondary contributors included `dti`, `open_acc` and `verification_status`.

This makes sense for unsecured consumer credit as internal-style grade, affordability, pricing and leverage are one of the most important factors that determine defaults.

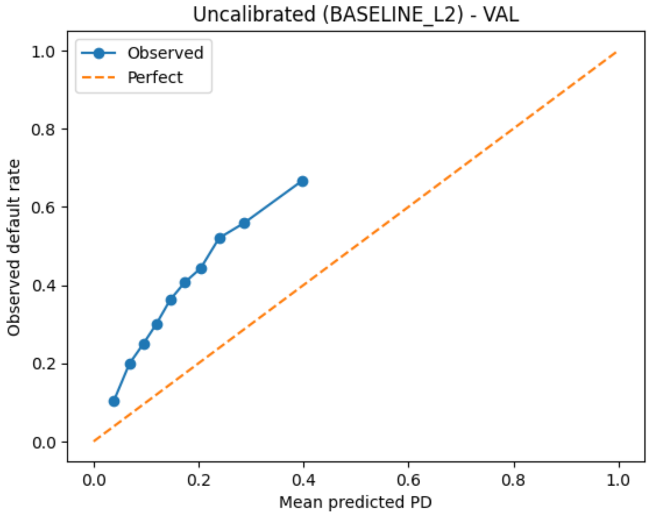


Figure 1: Uncalibrated validation calibration plot.

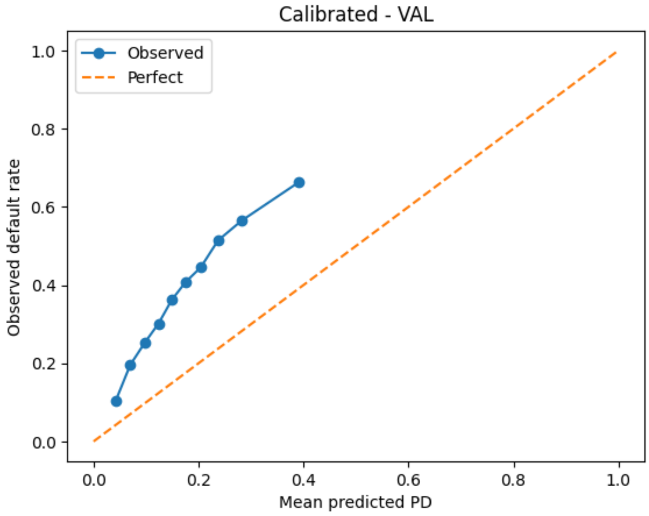


Figure 2: Calibrated validation calibration plot.

5 Stage 3: EAD and LGD assumptions

5.1 EAD

For EAD, I used a contractual amortisation approach. Rather than treating exposure as the original loan amount forever, I approximated the remaining balance after a given number of payments.

That produced:

- **EAD (6m)**
- **EAD (12m)**

These are scheduled balance proxies, not behavioural exposures. In other words, they do not capture prepayment, delinquency behaviour, or line utilisation dynamics. That is a simplification, but it is still much better than using original balance as a flat proxy.

5.2 LGD

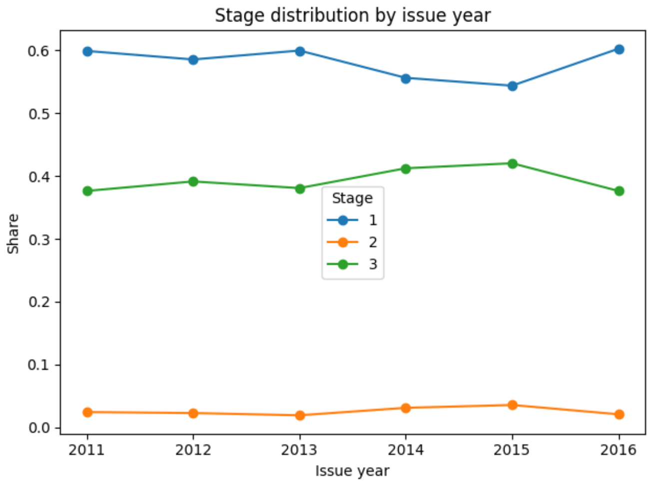
LGD was handled with transparent assumptions rather than estimated recoveries:

- base LGD: **0.40**
- downturn LGD: **0.55**

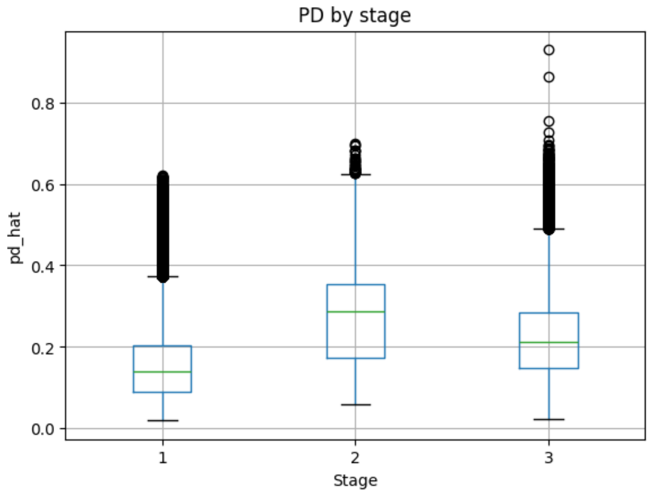
I also applied a simple grade-based scaling to make the LGD assumption more tailored and less uniform:

- A: 0.75
- B: 0.875
- C: 1.00
- D: 1.125
- E: 1.25
- F: 1.35
- G: 1.45

This was used as a straightforward segmentation device rather than a claim that recoveries were truly estimated.



(a) Stage distribution by issue year.



(b) Predicted PD by stage.

Figure 3: Comparison of stage distribution and predicted PD by stage.

6 Stage 4: IFRS 9 staging

6.1 Staging rule

The staging logic used a PD-based SICR proxy. First, I defined an origination PD benchmark as the median predicted PD inside each `sub_grade` bucket.

Then I used the following SICR rule:

$$\text{PD ratio} = \frac{PD_{\text{hat}}}{\max(PD_{\text{orig}}, \epsilon)} \geq 1.5 \quad \text{or} \quad \text{PD delta} = PD_{\text{hat}} - PD_{\text{orig}} \geq 0.07$$

with a small ϵ used only to avoid divide-by-zero issues.

Stages were assigned as:

- **Stage 1**: not SICR
- **Stage 2**: SICR flagged
- **Stage 3**: default proxy

6.2 Stage distribution

The final raw stage distribution was:

- Stage 1: **57.38%**
- Stage 2: **2.75%**
- Stage 3: **39.87%**

Once sample weights were applied, the portfolio-like stage mix shifted to:

- Stage 1: **79.59%**
- Stage 2: **3.82%**
- Stage 3: **16.59%**

The weighted matters most for ECL aggregation, because it corrects for the non-default downsampling used earlier.

7 Stage 5: Expected credit loss

7.1 Lifetime PD approximation

Stage 1 used 12-month PD. Stages 2 and 3 used a simple lifetime PD approximation based on the remaining term:

$$PD_{\text{LT}} = 1 - (1 - PD_{12m})^T$$

where T is the remaining term in years, derived from `term`.

7.2 ECL formulas

Using the chosen exposure and LGD assumptions:

$$ECL_{\text{base}} = PD \cdot LGD_{\text{base}} \cdot EAD$$

$$ECL_{\text{downturn}} = PD \cdot LGD_{\text{downturn}} \cdot EAD$$

The project then used a simple scenario blend:

$$ECL_{\text{weighted}} = 0.6 \cdot ECL_{\text{base}} + 0.4 \cdot ECL_{\text{downturn}}$$

I also ran a sensitivity where Stage 3 uses $PD = 1$, as a stricter credit-impaired assumption.

7.3 Portfolio results

The final weighted results were:

Table 2: Weighted portfolio ECL summary

Metric	Value
Weighted total EAD	14,764,123,459.99
Weighted ECL (base)	1,725,012,436.30
Weighted ECL (downturn)	2,371,892,099.92
Weighted ECL (scenario-weighted)	1,983,764,301.75
Weighted ECL (Stage 3 PD = 1 sensitivity)	2,436,120,045.14
ECL / EAD	0.1344

By stage, weighted average ECL behaved as expected:

- Stage 1: **887.22**
- Stage 2: **3,989.29**
- Stage 3: **4,010.09**

8 What worked well

A few parts of this project were especially useful:

- The maturity cutoff fixed the biggest evaluation issue by reducing label censoring.
- Carrying sample weights through the full pipeline made the final portfolio results much more honest.
- The time-based split made the validation setup more realistic than a random split.
- A plain logistic regression, with decent preprocessing, turned out to be a strong enough baseline for this scope.

9 Limitations of the current version

This is a good implementation, but it still has obvious limitations:

- The default flag is a **loan-status proxy**, not a true observed 12-month default timestamp.
- EAD is contractual, not behavioural.
- LGD is assumption-based, not estimated from realised recoveries.
- ECL is undiscounted; a full IFRS 9 implementation would usually include discounting using the effective interest rate.
- Lifetime PD is approximated with a constant-hazard formula rather than a proper transition or survival model.

10 Why this still matters

Even with those simplifications, the project demonstrates the full chain from raw loan data to a portfolio-level loss number. More importantly, it shows the practical work behind that chain:

- building a reproducible modelling pipeline,
- handling sampling bias with explicit weights,
- validating with a sensible temporal split,
- defining a staging rule instead of treating staging as a black box,
- and turning model outputs into something that can be aggregated and discussed.

That was the main value of the project: not claiming a production-ready bank model, but showing that the full workflow was designed, implemented, checked and documented properly.

References

[1] International Accounting Standards Board (IASB). Ifrs 9 financial instruments, 2014. Standard and subsequent amendments.

[2] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[3] wordsforthewise. Lending club accepted loans (2007–2018q4). Kaggle dataset, 2018. Dataset file used: `accepted_2007_to_2018Q4.csv.gz`.