Project: Exposure at Default (EAD), and Loss Given Default (LGD)

1. Goal

You are given a synthetic credit-card dataset (two CSV files).
Your job is to write a **single Jupyter notebook** that:

1. Loads the data.
2. Calculates **Exposure at Default (EAD)** for defaulted accounts.
3. Calculates **Loss Given Default (LGD)** for those same accounts.
4. Generates statical summaries of the above two metrics.

That's it. No modelling, no dashboards.

---

2. Data (what you need to know)

You will get two tables:

1. **lgd_accounts.parquet:** ***Account-level data*** (one row per account), with at least:
    - account_id
    - default_dt – the date the account went into default (empty if it never defaulted)
    - balance_at_default – the balance on the default date
    - write_off_amount – remaining balance after all recoveries (can be 0)
2. **lgd_transactions:** ***Transaction-level data*** (many rows per account), with at least:
    - account_id
    - trx_dt – transaction date
    - trx_amount – **signed** amount
        - positive: charges (purchases, fees, interest)
        - negative: payments
    - trx_type – e.g. PURCHASE, PAYMENT, FEE, INTEREST

You can assume the sign convention above is correct.

---

3. EAD (Exposure at Default) is "how much the customer owed the bank on the day they defaulted." In other words, their account balance at default.

---

4. LGD (Loss Given Default) is "what fraction of EAD the bank ultimately loses after trying to collect." Mathematically, for every defaulted account, i,

$$LGD_i = 1 - \frac{\min\left(recovery\ cash_i, EAD_i\right)}{EAD_i}$$

Where:

$$recover\ cash_i = \sum_{j:\ \forall\ PAYMENT\ trxs\ after\ default\ date} trx\ amount_j$$

---

5. Deliverables (what your notebook must do)

1. Output a table (e.g. a DataFrame) with at least:
    - account_id
    - EAD
    - recovery_amount
    - LGD
2. Output two tables that provide a statistical summaries of the EAD and LGD values:
    - # observations.
    - Min.
    - Max.
    - Mean.
    - STD.
    - Percentiles: 25%, 50%, 75%.

- Academic integrity criteria:
    - You are not allowed to collaborate with members or individuals outside of the group.
    - Any team member caught receiving external help outside of the instructors and fellow teammates will receive a zero grade for the whole group and potentially additional penalties including expulsion from the MFI programme.
    - Any member caught failing to report any suspicious activity of academic offence will be held liable and may receive equal punishment to the direct offenders.
    - Any team member who has any concerns about the work contributions of others is encouraged to reach out to the instructor before submission. This may result in the instructor and TA separately interviewing each member on the material and assessing if an equal grade should still be provided.

Some tips

- Start early.
- Review the lecture notes and the material covered and follow the same patterns and functions provided.
- Don’t complicate this: just meet the requirements.
- Make sure your delivery is clean and polished. We will judge on aesthetics, presentation quality, code and script cleanliness and efficiency, reproducibility (if I run the script I should see the same numbers as the notebook you gave me before running anything), not just output.
- Ask questions and engage the instructor/TA whenever unsure.
- Ensure you follow the practices preached and applied in the lecture notes and tutorial files.
- Leverage online sources such as Stack Overflow and ChatGPT for syntax.