All Four Phases of Project

Phase 1 — Data Exploration & Cleaning

1.1 EDA with Cross-tables & Pivot Tables

Theory (statistical intuition)

Cross-tabs help you estimate **conditional default rates**:

P(default=1|category)P(\text{default}=1 \mid \text{category})P(default=1|category)

This is the simplest "risk signal" check. If some category has a much higher default rate, it's likely predictive.

Business Insight

Banks do this early to:

- spot **high-risk segments** (e.g., certain grades/intents/home ownership),
- detect **policy issues** (e.g., high default rate in a product segment),
- support **risk appetite** decisions ("we avoid segment X or price it higher").

Critique & Improve

- Add **Population Stability Index (PSI)** for drift monitoring.
- Add **IV (Information Value)** and **WoE analysis** to quantify predictive power.

1.2 Outliers (detect + visualize)

Theory

Outliers can:

- distort model coefficients (especially logistic regression),
- represent data errors (e.g., impossible ages/incomes),
- represent real but rare cases (high-income, high-loan).

Boxplots and **scatter** reveal heavy tails.

Business Insight

Outlier policies connect to **fraud**, **policy exceptions**, and **portfolio concentration risk**.

Critique & Improve

- Use **robust scaling** or **winsorization** (cap extreme values).
- Use **outlier flags** as features (sometimes predictive).

1.3 Missing Data (imputation vs removal)

Theory (intuition)

Missingness can be:

- **MCAR** (random) → simple imputation OK.
- **MAR** (depends on other variables) → model-based imputation better.
- **MNAR** (depends on missing value itself) → missingness indicator is crucial.

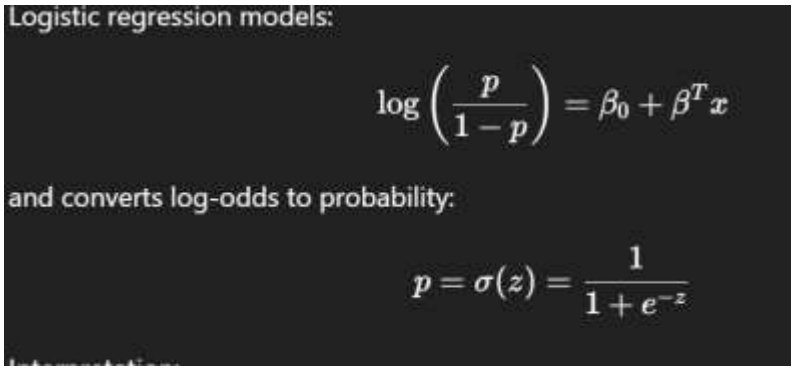
Imputation methods:

- **Mean/median**: fast baseline, but shrinks variance.
- **Most frequent**: common for categorical.
- **Model-based**: KNN / iterative, but can leak info if not done carefully.

Phase 2 — Logistic Regression (Industry Standard)

2.1 Theory: Sigmoid, log-odds, probability

Logistic regression models:



Interpretation:

- A coefficient ßj means a 1-unit increase in xj multiplies odds by e^ ßj

Business Insight

Logistic regression is popular because it's:

- interpretable (regulators like it),
- stable and easy to monitor,
- good baseline PD model.

2.2 Feature Engineering: One-Hot + WoE (optional but portfolio-strong)

Quick WoE Theory

For a bin/category g:

$$WoE(g) = \ln\left(\frac{\%\text{non-default in } g}{\%\text{default in } g}\right)$$

WoE often makes relationships more linear in log-odds space and helps stability.

We'll implement:

- **OneHotEncoding pipeline** (industry baseline)
    -
    - **optional WoE** demo for categoricals

2.4 Metrics Theory: Confusion Matrix, Precision, Recall trade-off

- **Recall (TPR)**: "catching defaults" → reduces credit losses, but may reject good customers.
- **Precision**: "when we flag default, we're correct" → avoids unnecessary rejections.

Business Insight (bank perspective)

Threshold is a business decision:

- conservative bank → prioritize recall (avoid defaults),
- growth-focused bank → prioritize precision / acceptance.

Critique & Improve

- Tune threshold to optimize **expected profit**, not just accuracy.
- Use **regularization strength (C)** tuning.
- Use **calibration** (important for PD).

Phase 3 — Advanced Modeling (XGBoost / Gradient Boosting)

3.1 Theory: Gradient Boosting intuition

Boosting builds trees sequentially; each new tree learns to correct errors (residuals / gradients) of the previous ensemble. This reduces bias and captures non-linearities + interactions.

Business Insight

Tree boosting often improves rank-ordering (AUC), which helps:

- better **risk-based pricing**,
- stronger underwriting,
- more profitability per risk unit.

3.2 Class Imbalance: SMOTE / undersampling + calibration impact

Theory

SMOTE changes the training distribution; it can improve classification but can harm probability calibration. So you often:

- train with imbalance technique,
- then **calibrate** probabilities (Platt/Isotonic).

3.3 XGBoost Implementation + Cross-Validation + Feature Importance

Phase 4 — Model Evaluation & Business Strategy

4.1 Compare ROC & AUC

Theory

ROC compares rank-ordering across thresholds.
AUC = probability a random defaulter is scored riskier than a random non-defaulter.

.2 Calibration Curves (do PDs match reality?)

Theory

A model can have good AUC but poor calibration. Banks care about calibration because PD is used in:

- Expected Loss,
- IFRS 9 / CECL style provisioning (depending on regime),
- pricing and limit setting.

4.3 Strategy Table: Expected Loss (EL = PD × LGD × EAD)

Theory (banking core)

- **PD** from model
- **LGD** = loss given default (e.g., 45% unsecured assumption if not available)
- **EAD** = exposure at default (loan amount here)

EL=PD·LGD·EADEL = PD \cdot LGD \cdot EADEL=PD·LGD·EAD

We'll build a strategy table by sorting applicants by PD, then checking:

- acceptance rate,
- bad rate among accepted,
- total expected loss among accepted,

- expected portfolio value proxy.

Business Insight

This is exactly how a bank uses a PD model:

- pick a cutoff that meets **risk appetite** ("bad rate must be < X%"),
- maximize portfolio value under constraints,
- justify strategy to **risk + compliance**.

Critique & Improve

- Replace value proxy with real economics:
  - revenue (APR, fees), funding cost, expected prepayment, operational cost, capital charges.
- Add **reject inference** discussion (real-world acceptance bias).
- Add monitoring: drift, stability, backtesting, challenger models.

Summary (Interpretation):

This project builds a bank-style **Credit Risk Modeling** pipeline to predict **Probability of Default (PD)** from a consumer loan dataset and convert predictions into underwriting decisions. In Phase 1, the data is profiled using cross-tabs/pivots to identify high-risk segments, outliers are checked with boxplots/scatter plots, and missing values are treated to ensure model-ready inputs. Phase 2 trains an interpretable **Logistic Regression** model with leakage-safe preprocessing (imputation + one-hot encoding) and evaluates classification trade-offs using confusion matrix and ROC/AUC. Phase 3 benchmarks a higher-capacity **XGBoost** model with SMOTE and cross-validation, and explains risk drivers using permutation importance. Phase 4 compares models using ROC/AUC and **calibration curves**, then builds a **Strategy Table** using **Expected Loss (EL = PD×LGD×EAD)** to show how acceptance-rate cutoffs change portfolio bad rate, loss, and value.