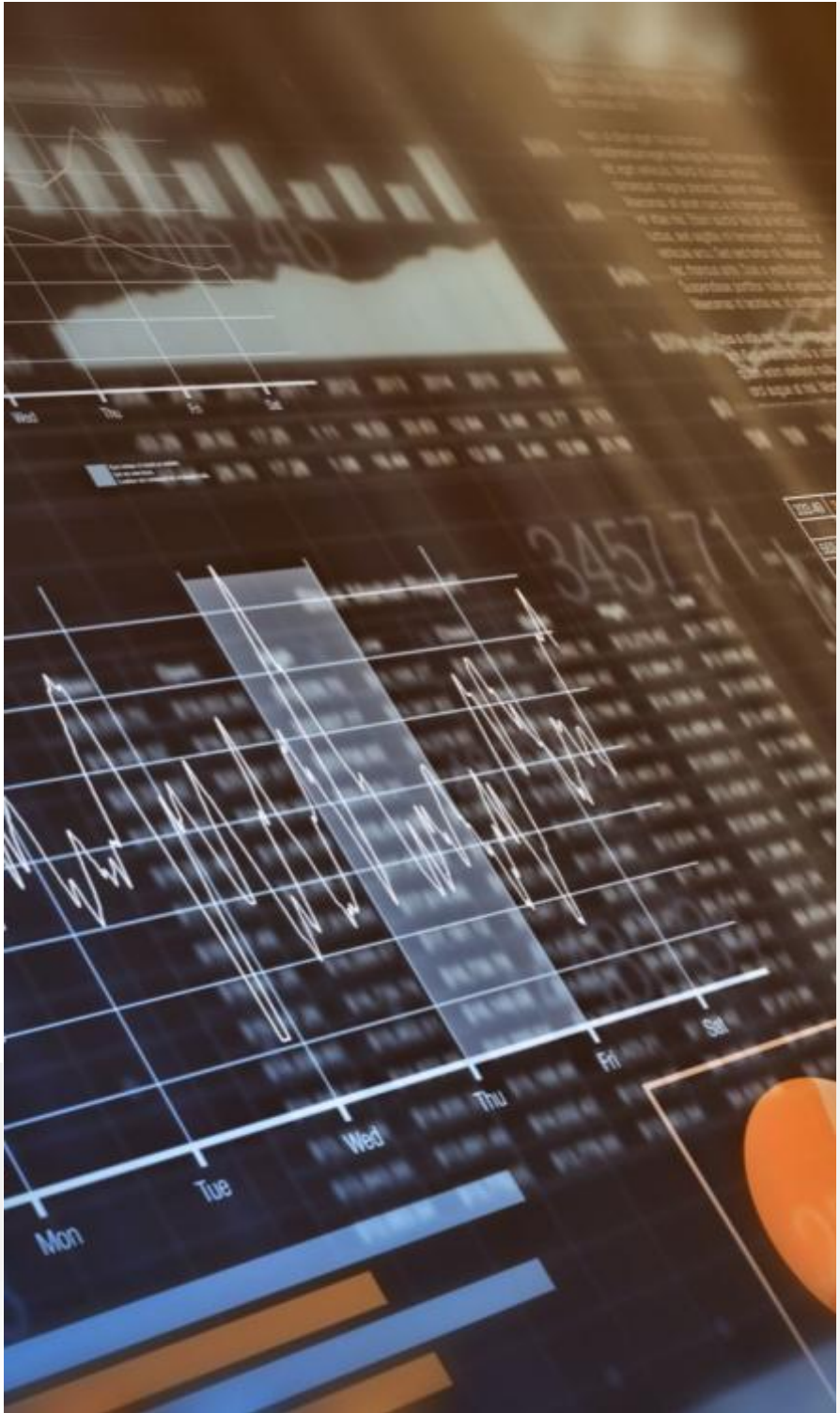
Business Data Analytics – Lecture 3: Data Quality & Feature Engineering

Assoc. Prof. Nguyen Binh Minh, Ph.D.



Learning Objectives (CLO)

- Understand data quality dimensions and measurement.
- Diagnose and handle missingness.
- Detect and avoid data leakage.
- Encode categorical variables and scale numeric features.
- Engineer features and perform feature selection.
- Build reproducible, leakage-safe pipelines.

Agenda & Timing

1) Data quality (40’)
2) Missing data (35’)
3) Data leakage (25’)
4) Categorical encoding (35’)
5) Scaling & outliers (25’)
6) Feature engineering & selection (40’)
7) Pipeline & mini-case (30’)
8) Short quiz & wrap-up (10’)

Running Case: Telco Churn Prediction

Dataset: 10k customers, 35 variables (demographics, plan, usage, label).

Goal: predict churn in the next 30 days.

Challenges: missing income/tenure, leakage (post-event features),

High cardinality plan_id, mixed scales.

What is Data Quality?

Accuracy, Completeness, Consistency.

Validity, Timeliness, Provenance.

Direct impact on model performance and ops cost.

Data Quality Scorecard

Example KPIs: % missing < 2%/feature; duplicate rate < 0.5%; out-of-range < 0.1%.

Data SLA: daily H+1 before 08:00; stable schema outside maintenance windows.

Data Quality Lifecycle

Ingest → Profile → Clean → Validate → Monitor → Feedback.

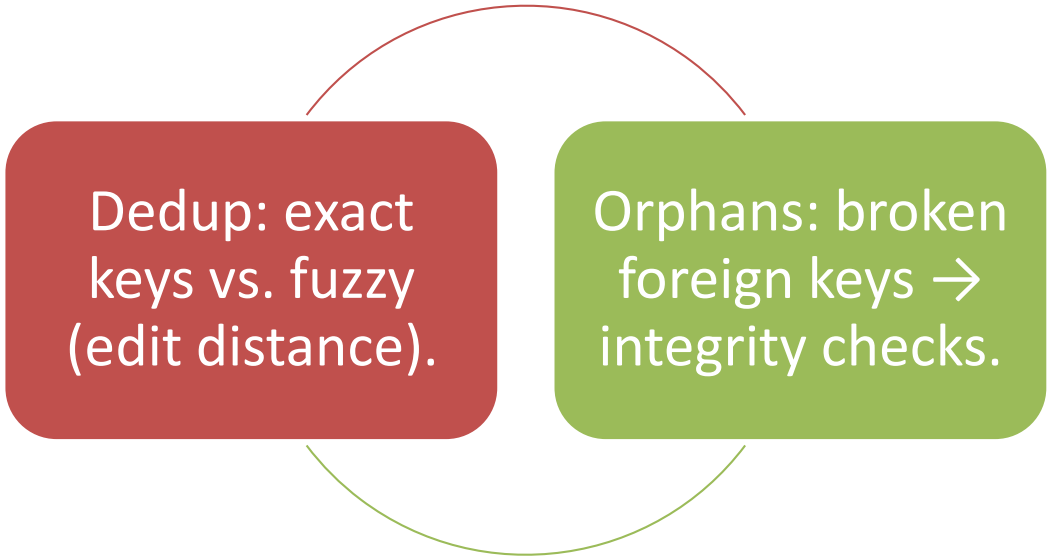
Combine rule-based checks with descriptive statistics.

Quick Data Profiling

Distributions, min/max, outliers, cardinality, missing map.

Visuals: histogram, boxplot, correlation heatmap, category bars.

Duplicates & Orphan Records

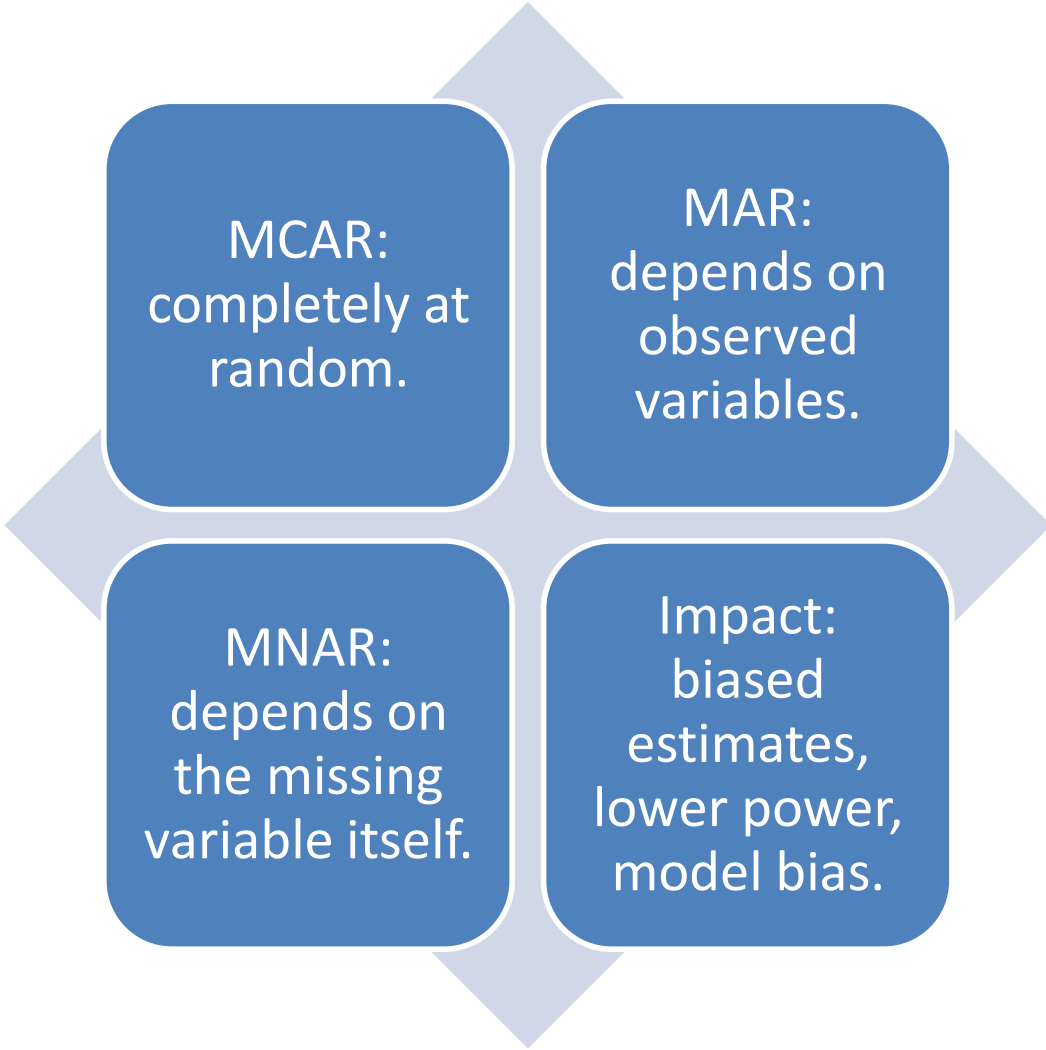


Units & Value Domains

Normalize currencies, percentages, and units (km $\leftrightarrow$ m).

Validity rules (e.g., age $\in$ [0, 120]).

Missingness Mechanisms



Diagnosing Missingness

Column/row missing maps; relation between missingness and label.

Little's MCAR idea; compare distributions in missing vs. non-missing groups.

Example: Missing by Segment

Income missing more in Prepaid; Tenure missing in new customers.

Likely MAR by plan_type; possible MNAR for income.

Missingness Strategies — Overview

Row/column drop if excessive missingness or low-value feature.

Imputation: mean/median/mode, group-wise, kNN, MICE, model-based, business rules.

Add an is_missing indicator.

Simple & Group-wise Imputation

Median/Mode robust to outliers.

Group-wise medians by plan_type.

kNN Imputation

- Find k nearest neighbors → average/majority.
- Pros: preserves multivariate structure; Cons: compute-heavy, scale-sensitive.
- Apply scaling inside a pipeline to avoid leakage.

MICE (Multiple Imputation by Chained Equations)

- Model each incomplete feature iteratively until convergence.
- Produce multiple datasets and pool estimates.
- Best for MAR; preserves correlation structure.

Business- and Time-aware Imputation

- Time series: forward/backward fill, interpolation, seasonal Kalman.
- Domain rules: income = base salary × factor; apply caps.

Assessing Imputation Impact

- Compare model metrics (AUC/RMSE) across strategies.
- Check distribution drift after imputation; keep predictive signal.

Common Pitfalls in Missingness

- Global-mean imputation using full data → leakage.
- Forgetting the is_missing flag.
- Imputing before train/test split → leakage!

Data Leakage — Definition & Impact

- Information from future/test creeps into training.
- Over-optimistic training scores, sharp drop in production.

Leakage Types

- Target leakage: features contain the label or its consequences.
- Train–test contamination: preprocessing fitted on full data.
- Temporal leakage: using post-hoc features for forecasting.

Example of Target Leakage (Churn)

- Feature 'refund_in_30d' appears in training but occurs after churn.
- Mitigation: drop or time-shift to before the prediction point.

Avoiding Temporal Leakage

- Time-based splits (TimeSeriesSplit/rolling origin).
- Freeze feature windows to pre-T0 data only.

Anti-Leakage Pipeline Principles

- Fit preprocessing only on training folds.
- Use Pipeline/ColumnTransformer.
- Perform cross-validation on the full pipeline.

Data Splitting Strategies

- Holdout, KFold, StratifiedKFold.
- TimeSeriesSplit for temporal data.
- Group-aware split when multiple rows per customer.

Categorical Encoding — Overview

- One-Hot, Ordinal, Target/Mean, WoE, Hashing, Frequency/Count.
- Issues: high cardinality, rare categories, target-encoding leakage.

One-Hot Encoding (OHE)

- Pros: simple, no order assumption.
- Cons: dimensionality explosion with high cardinality.
- Tips: group rare levels to 'Other'; use hashing if needed.

Ordinal Encoding

Map categories to integers with true ordering (S < M < L < XL).

Avoid for unordered categories.

Target/Mean Encoding (with CV)

- Replace category with target mean (with smoothing).
- Compute inside CV loops to prevent leakage.
- Add noise/smoothing to reduce overfit.

Weight of Evidence (WoE)

- WoE = ln((Good_i/Bad_i)/(Good/Bad)) — widely used in credit risk.
- Pros: monotonicity, interpretability; Cons: requires binning and enough samples.

Hashing Trick

- Map categories to a fixed space via hashing.
- Pros: handles high cardinality and streaming; Cons: collisions, less interpretable.

High-Cardinality Tactics

Collapse rare levels; frequency/count encoding.

CV-safe target encoding; or hashing + embeddings.

Text Features (quick)

Bag-of-Words/TF-IDF; n-grams; remove stopwords.

Pair with linear/logistic models or tree-based learners.

Entity Embeddings (idea)

Learn dense representations of categories with neural nets.

Great for high-cardinality interactions.

Why & When to Scale

Scale-sensitive algorithms: kNN, k-means, SVM, PCA, L1/L2 regression.

Tree-based models less sensitive but scaling helps when mixing models.

Scaling Techniques

- StandardScaler: (x−μ)/σ.
- MinMaxScaler: [0,1].
- RobustScaler: median/IQR for outlier robustness.
- Always fit on train, transform train/test.

Power / Variance-Stabilizing Transforms	Log, Box-Cox ($x>0$), Yeo-Johnson ($x \in \mathbb{R}$).	
	Make distributions closer to normal; reduce skew.	

Quantile / Rank-Gauss Transform	Map to uniform/normal distributions.
	Useful for linear models and normality assumptions.

Outliers — Detection

Z-scores, IQR (Q1−1.5*IQR, Q3+1.5*IQR).

Model-based: Isolation Forest, LOF.

Note: outliers may carry business signal.

Outliers

—

Handling

Capping/Winsorization; log transforms; robust loss.

Conditional removal with domain investigation.

Feature Transformations Toolkit

Binning (equal-width/frequency), interactions, polynomials.

Ratios/Rates (e.g., ARPU = revenue/usage).

Time-normalized rates.

Date–Time Features

- Calendar: day/week/month/quarter/year; holiday flags.
- Seasonality with sine–cosine for cycles (hour/day).
- Recency/freshness: days since last activity.

Geo Features (if coordinates)	Haversine distances, regional clustering, point density.
	Grid encoding (geohash) for regional aggregation.

Aggregation / Group-by Features	Counts/frequencies per user/plan.
	Group means/medians/std; shares/ratios.
	Leave-one-out strategies to reduce leakage.

Time-Series Features	Lags (t−1, t−7, t−30), rolling mean/std/max/min.
	Exponential moving features; differencing.

Text/NLP Features (extended)	Sentiment scores, keyword flags (complaints).	
	TF-IDF + SVD (LSA) for dimension reduction.	

Feature Selection — Overview

Filter: Chi-square, ANOVA, Mutual Information.

Wrapper: RFE, SFS/SBS.

Embedded: L1/L2, tree-based importance, SHAP.

Filter Methods

- Chi-square for categorical vs. target.
- ANOVA F-test for numeric vs. class.
- Mutual Information for non-linear dependencies.

Wrapper Methods

RFE with estimator; choose k via CV.

Trade-off: accuracy vs. compute time.

Embedded & Regularization	Lasso (L1) for sparsity; Ridge (L2) for variance reduction.
	Elastic Net balances L1/L2.
	Tree-based importance: Gini/Permutation.

Dimensionality Reduction

PCA preserves variance in orthogonal components.

t-SNE/UMAP for visualization; avoid direct forecasting use.

Golden Rules for Pipelines

Fit	Never fit preprocessing on the full dataset.
Use	Use ColumnTransformer for per-type pipelines.
Wrap	Wrap hyperparameter search around the pipeline.

Pipeline Diagram (concept)

- Raw → Split → (Impute → Encode → Scale → FeatureSelect) → Model → Eval.
- For temporal data: enforce pre-T0 feature windows.

Sklearn Pipeline (leakage-safe) — Pseudocode

```
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder, RobustScaler
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import StratifiedKFold, cross_val_score
from category_encoders.target_encoder import TargetEncoder

num_pipe = Pipeline([
    ("imputer", SimpleImputer(strategy="median")),
    ("scaler", RobustScaler())
])

cat_high_pipe = Pipeline([
    ("target_enc", TargetEncoder())
])

cat_low_pipe = Pipeline([
    ("onehot", OneHotEncoder(handle_unknown="ignore"))
])

pre = ColumnTransformer([
    ("num", num_pipe, num_cols),
    ("cat_high", cat_high_pipe, high_card_cols),
    ("cat_low", cat_low_pipe, low_card_cols)
])

clf = Pipeline([
    ("preprocess", pre),
    ("model", LogisticRegression(max_iter=200))
])

cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
score = cross_val_score(clf, X, y, cv=cv, scoring="roc_auc")
```

Model Evaluation after Preprocessing

- Metrics: ROC-AUC/PR-AUC (class imbalance), Brier score, KS.
- Compare impute/encode/scale strategies via consistent CV.

Data Validation & Monitoring

- Schema contracts (types, ranges, constraints).
- Pre-deploy checks: null %, ranges, cardinality, drift.
- Production monitoring with alerts and thresholds.

Data Contracts & Schema Evolution

- Agreements between data producers and consumers: fields, definitions, SLAs.
- Schema versioning and backfill strategies.

Reproducibility & Experiment Tracking

- Random seeds, environment files.
- Track params, metrics, artifacts, and data snapshots.

Common Anti-patterns

- Future-derived features; preprocessing on full data.
- Unbounded one-hot; ignoring rare categories.
- No is_missing flags; ignoring group-aware splits.

Mini-case: Churn — Part 1 (Profiling)

- Task: profile and define a data quality checklist.
- Deliverable: DQ metrics table for top 10 columns.

Mini-case: Churn — Part 2 (Missing & Encoding)

- Choose strategies for income and tenure imputation.
- Encode plan_id (high cardinality) with CV-safe target encoding.

Mini-case: Churn — Part 3 (Scaling & Model)

- Compare Logistic (scaled) vs. XGBoost (no scaling needed).
- Report ROC-AUC/PR-AUC and calibration plot.

In-class Hands-on

- 1) Build the leakage-safe pipeline as shown.
- 2) Run 5-fold CV and log results.
- 3) Write five lines summarizing trade-offs.

Summary — Key Takeaways

- Data quality sets the ceiling for model performance.
- Leakage-safe pipelines are mandatory.
- Encoding/Scaling must match the model context.
- Feature engineering blends domain knowledge and controlled experiments.

Appendix — Formulas & Notes

Standardize: $z = (x - \mu) / \sigma$.

IQR = Q3 – Q1; capping at [Q1 – 1.5*IQR, Q3 + 1.5*IQR].

WoE/IV basics; time cyclic encoding: $\sin(2\pi t/T)$, $\cos(2\pi t/T)$.

Appendix — Group-wise Imputation (pandas)

```
X['income'] = X.groupby('plan_type')['income'].transform(
    lambda s: s.fillna(s.median())
)
X['income_is_missing'] = X['income'].isna().astype(int)
```

Appendix — CV-safe Target Encoding

```
from category_encoders.target_encoder import TargetEncoder
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.linear_model import LogisticRegression


te = TargetEncoder(cols=['plan_id'], smoothing=0.2)
model = Pipeline([
   ('pre', ColumnTransformer([('plan', te, ['plan_id'])], remainder='drop')),
   ('clf', LogisticRegression(max_iter=200))
])
# Use within cross_val_score/cross_val_predict to avoid leakage
```

Recommended References

- Kuhn & Johnson (2019) — Feature Engineering and Selection.
- Pang-Ning Tan et al. — Introduction to Data Mining.
- scikit-learn docs: Pipeline, ColumnTransformer, Imputation, Encoding, Scaling.
- Credit scoring resources: monotone binning, WoE/IV.

Visual Example — Missingness Map

Synthetic telco-like dataset (1 = missing)

Visual Example — Outliers

Boxplot of Monthly Charges