Week-2 Planning Report: Data Cleaning & Preprocessing Framework for Food Quality Data

Data Analytics Task

December 2025

1 Introduction: Why Data Cleaning is Critical

Food processing industry data usually comes from multiple heterogeneous sources such as sensors (temperature, humidity, pH, flow rate), laboratory measurements (moisture content, microbial counts), production logs, and inspection records. In raw form, this data is often:

- **Incomplete**: missing readings due to sensor downtime or manual entry errors.
- **Noisy**: spikes and faulty values caused by sensor malfunction or human mistakes.
- **Inconsistent**: mixed units (e.g. °F and °C), different date formats, and spelling variations in categorical fields.
- **Redundant**: duplicate rows from repeated logging or system retries.

Agar aisa raw data directly analysis mein use kiya jaata hai, to:

- Averages, trends, correlations aur models biased ho sakte hain.
- Outliers ki wajah se thresholds galat set ho sakte hain (e.g. ek faulty sensor value poori control-limit design ko distort kar sakti hai).
- Regulatory ya business decisions galat interpretation par based ho sakte hain.

Is liye Week-2 ka focus pure planning par hai: **data profiling, cleaning strategy, preprocessing (model-ready data) ka plan, aur documentation ka framework** define karna, taaki Week-3+ mein safe and reliable analysis ki groundwork ready ho.

2 Data Profiling Plan

Data profiling ka main goal hai raw dataset ko systematically samajhna before any cleaning or modeling. Week-2 mein sirf yeh plan document kiya jaayega ki profiling practically kaise ki jaayegi.

2.1 Column Structure and Naming

- Saare column names ki list banayi jaayegi (e.g. `batch_id`, `temp_c`, `moisture_pct`, `ph_value`, `defect_flag`).
- Inconsistent ya unclear names ko rename karne ka plan banega (e.g. `Temp`, `temperature_C`, `T_C` ko standard `temp_c` par laana).
- Har column ke liye short description likha jaayega (data dictionary draft).

2.2 Data Types Check

- Har column ka expected type define kiya jaayega:
  - Numeric (e.g. temperature, moisture, pH, counts).
  - Categorical (e.g. product type, line ID, shift, defect type).
  - DateTime (e.g. batch start time, sensor timestamp).
- Plan: profiling ke time par data-type mismatches detect karna (e.g. numeric field stored as string) aur unke liye conversion rules likhna.

2.3 Missing Values Analysis

- Har column ke liye:
  - Missing count (absolute).
  - Missing percentage (relative).
- Columns ko categories mei divide karne ka plan:
  - Low missingness ($<5\%$).
  - Moderate (5–20%).
  - High ($>20\%$).
- Results ke basis par missing-value handling strategy (Section 3) apply hogi.

2.4 Duplicate Rows Check

- Exact row-level duplicates identify karne ka plan (same batch, same timestamp, same values).
- Timestamp-based duplicates: e.g. same sensor, same timestamp ke multiple records ko flag karna.
- Key-based duplicate rules define karna (e.g. same `batch_id` + `line_id` + `timestamp` considered duplicate).

2.5 Outlier Scanning Plan

- Numeric columns ke liye:
  - Boxplot/IQR-based outlier detection.
  - Z-score calculation for approximately normal variables.
- Domain-based checks: e.g. temperature $-5°C$ in a process that should always be between $0°C$ and $120°C$ ko immediately suspect mark karna.

2.6 Basic Descriptive Statistics

- Har important numeric column ke liye:
  - Mean, median, min, max, standard deviation.
- Categorical columns ke liye:
  - Unique values ka count.
  - Top categories aur unki frequency.

- Ye stats profiling summary report mein capture kiye jaayenge taaki cleaning decisions justify ho saken.

3 Data Cleaning Strategy

Is section mein Week-2 ke liye conceptual strategy define hogi: kaun se issues kaise handle honge. Actual implementation later weeks mein hogi.

3.1 Missing Values Handling

Numeric features (e.g. temperature, moisture, pH):

- **Mean imputation**: Use jab distribution reasonably symmetric ho aur outliers limited hon (e.g. temperature sensor data in controlled process).
- **Median imputation**: Use jab distribution skewed ho ya strong outliers hon (e.g. microbial counts, defect counts).
- **Forward/Backward fill**: Time-series data (continuous sensor logs) ke liye:
  - Forward fill (last valid value carry forward) jab missing gap short ho.
  - Backward fill agar start me missing ho ya reverse direction lebih suitable ho.

Categorical features (e.g. product type, shift, line ID):

- **Mode imputation**: Most frequent category se fill jab missing proportion low ho aur pattern random ho.
- **Separate "Unknown" category**: Jab missingness informative ho (e.g. certain supplier often omits data) ya high percents hon.

High-missing features:

- Agar kisi column mein bahut zyada missing values hon (e.g. >40–50%), to:
  - Option 1: Feature ko drop karne ka plan (agar domain-wise critical nahi hai).
  - Option 2: Rows drop karna (agar feature critical hai lekin sirf kuch specific rows mein missing hai).

3.2 Outlier Detection & Treatment

Detection methods:

- **IQR method**: Q1 aur Q3 compute karke Inter-Quartile Range (IQR = Q3–Q1) nikala jaayega. Jo values $[Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR]$ ke bahar hain, unko outlier candidate maanenge.
- **Z-score method**: Approximately normal variables ke liye $|z| > 3$ ko extreme outlier candidate consider karna.
- **Domain rules**:
  - Temperature below plausible lower limit (e.g. $-5°C$ in a process that never goes below $0°C$).
  - Moisture >100% ya pH range ke bahar (e.g. pH < 0 or > 14).

Treatment plan:

- Obvious sensor errors (impossible values) ko:
  - Either missing treat karke imputation se handle karenge, ya
  - Row drop karenge agar record ka core information unreliable ho.
- Genuine but extreme values (possible rare events) ke liye:
  - Winsorization (cap at certain percentile) ya
  - Separate analysis track (flagged but retained) plan kiya jaayega.

3.3 Duplicate Handling

- **Exact duplicates**: Puri row identical hone par unko safely remove karne ka plan (unique measurement ko preserve karte hue).
- **Timestamp-based duplicates**: Same sensor + same timestamp + similar values ke multiple records ke liye:
  - Latest record keep karna, ya
  - Average value lena (agar domain-wise acceptable ho).

3.4 Inconsistency Resolution

- **Unit conversion**:
  - All temperatures ko °F se °C me convert karna:

$$T_{\circ C} = (T_{\circ F} - 32) \times \frac{5}{9}$$

  - Moisture, concentration, etc. ke units standardize karna (e.g. % w/w).
- **Spelling / category consistency**:
  - "Line-1", "line1", "LINE_1" ko ek standardized category banaana.
  - Product names ke liye mapping table maintain karna.
- **Date format standardization**:
  - Jo dates "DD-MM-YYYY" ya "MM/DD/YYYY" me hain, unko standard ISO format "YYYY-MM-DD" me convert karne ka rule define karna.

4 Preprocessing (Model-Ready Data) Plan

Ye section is baare mein hai ki cleaned data ko future modeling/analysis ke liye kaise prepare kiya jaayega. Week-2 me sirf planning document hogi.

4.1 Normalization & Scaling

Min–Max Scaling:

- Use jab features ko $[0, 1]$ range me laana ho, especially distance-based algorithms (e.g. k-NN, clustering) ke liye.
- Example: temperature, moisture, pH ko comparable scale par lana for certain models.

Standard Scaling (Z-score):

- Use jab algorithms mean 0 aur variance 1 wali distribution assume karte hain (e.g. linear regression, logistic regression, PCA).
- Plan: numeric features ko μ and σ ke basis par standardize karna, outlier treatment ke baad.

4.2 Categorical Encoding

One-hot encoding:

- Low-cardinality categorical features (e.g. shift = {Morning, Evening, Night}; line_id = L1, L2, L3) ke liye.
- Har category ke liye separate binary column banane ka plan.

Label encoding:

- Jab algorithm simple tree-based ho (e.g. decision tree, random forest) ya ordinal relationship ho (e.g. freshness level = {Low, Medium, High}).
- Categories ko integer codes assign karna, with clear mapping table.

4.3 Feature Selection

- **Correlation-based**:
  - Highly correlated features ko identify karna (multicollinearity), unnecessary duplicates drop karne ka plan.
- **Variance-based**:
  - Near-constant ya zero-variance features (e.g. koi column jisme saare values almost same) ko drop karna, kyunki wo model ko information nahi dete.
- **Domain knowledge**:
  - Food processing context mein: pH, temperature profile, holding time, moisture content jaisi variables ko zyada priority; purely identifiers (product ID, internal code) ko modeling se exclude karna (sirf joins/lookup ke liye).

5 Documentation & Tracking Plan

Week-2 se hi har transformation ko traceable rakhne ke liye documentation plan define kiya jaayega.

5.1 File Naming Convention

- Raw files: `food_quality_raw_YYYYMMDD.{csv/xls}`.
- Cleaned files: `food_quality_cleaned_v1.{csv/xls}`.
- Processed/model-ready files: `food_quality_processed_v1.{csv/xls}`.

5.2 Versioning Strategy

- Har major change ke baad version increment:
  - v1, v2, v3 ... based on cleaning/preprocessing iterations.

- Raw data ko **read-only** treat karna; kabhi overwrite nahi karna.

5.3 Change Log

- Ek simple change-log document maintain karna jisme:
  - Date/time.
  - Responsible person.
  - Applied step (e.g. "mean imputation on temp_c", "dropped column X due to 70% missing").
  - Affected file name & version.
- Ye log later audit aur debugging ke liye reference ka kaam karega.

5.4 Before–After Snapshots

- Important cleaning steps (e.g. missing-value handling, outlier treatment) se pehle/baad main summary stats store karne ka plan:
  - Row count, missing % per column, basic mean/median/min/max.
- Isse yeh trace kiya ja sakega ki har step ka quantitative impact kya tha.

Optional Conclusion

Week-2 planning ka main result ek clear, written framework hai jo batata hai:

- Data profiling kaise kiya jaayega.
- Missing values, outliers, duplicates aur inconsistencies ko systematically kaise handle kiya jaayega.
- Cleaned data ko model-ready banane ke liye scaling, encoding aur feature selection ka kya approach hoga.
- Har change ko traceable rakhne ke liye documentation aur versioning ka kya system hoga.

Actual coding aur dataset cleaning next week mein is plan ke according implement ki jaayegi.