Complete Guide to Data Pre-processing

1. What is Data and Why Pre-process?

Understanding Data

Data = Collection of data objects and their attributes

- **Object** = A record, instance, or sample (like a student record)
- **Attribute** = A property or characteristic (like age, grade, name)

Example:

```
Student ID | Name    | Age | Grade | Subject
3204395    | Alice   | 20  | B     | CSCI316
3194284    | Bob     | 21  | A     | CSCI316
3483509    | Carol   | 19  | C     | CSCI316
```

Why Pre-process?

Raw data is often:

- **Incomplete** (missing values)
- **Inconsistent** (different formats)
- **Inaccurate** (errors, outliers)
- **Imbalanced** (unequal class distribution)

The Goal: Transform raw data into a clean, suitable format for mining algorithms.

---

2. Data Exploration - Understanding Your Data

Key Questions to Ask:

1. What types of attributes do I have?
2. Are there missing values?
3. Are there outliers?
4. Is the data balanced?
5. How are attributes distributed?

Attribute Types:

- **Numeric:** Age (20), Grade (85.5)
- **Categorical:** Gender (Male/Female), Grade (A/B/C)
- **Binary:** Passed (Yes/No)
- **Ordinal:** Rating (Poor/Good/Excellent)

Essential Statistics:

- **Mean:** Average value
- **Median:** Middle value (better for skewed data)
- **Mode:** Most frequent value
- **Standard Deviation:** Measure of spread
- **Range:** Min to Max values

---

3. Data Quality Issues

3.1 Missing Values

Common causes:

- Data not collected
- Equipment failure
- Human error

Solutions:

1. **Delete records** (only if <5% missing)
2. **Fill with mean/median/mode**
3. **Use domain knowledge**
4. **Predict missing values** using other attributes

Example:

```
Age | Income | Status
25  | 50000  | Employed
30  | NULL   | Employed  ← Fill with mean income
NULL| 60000  | Student   ← Fill with mean age
```

3.2 Outliers

Definition: Data points significantly different from others

Detection Methods:

- **3σ Rule:** Values > 3 standard deviations from mean
- **IQR Method:** Values outside [Q1-1.5×IQR, Q3+1.5×IQR]

Handling Outliers:

1. **Keep them** (if they're valid)
2. **Remove them** (if they're errors)
3. **Transform them** (binning, capping)

3.3 Noise

Definition: Random errors in data

Examples:

- Typos in text fields
- Sensor measurement errors
- Age = 127 (clearly wrong)

Solutions:

- **Data cleaning:** Remove/correct obvious errors
- **Smoothing:** Apply filters to reduce noise

---

4. Data Transformation

4.1 Normalization

Purpose: Scale attributes to similar ranges

Min-Max Normalization (0 to 1):

```
v = (x - x_min) / (x_max - x_min)
```

Z-Score Normalization (mean=0, std=1):

```
v = (x - mean) / std_deviation
```

Example:

```
Original: [100, 200, 300, 400, 500]
Min-Max:  [0, 0.25, 0.5, 0.75, 1.0]
Z-Score:  [-1.41, -0.71, 0, 0.71, 1.41]
```

4.2 Discretization/Binning

Purpose: Convert continuous values to discrete categories

Example - Age Binning:

```
Age: [18, 25, 30, 45, 60, 70]
Bins: Young(18-30), Middle(31-50), Senior(51+)
Result: [Young, Young, Young, Middle, Senior, Senior]
```

4.3 Encoding Categorical Data

One-Hot Encoding:

```
Color: [Red, Blue, Green, Red]
→
Red  | Blue | Green
1    | 0    | 0
0    | 1    | 0
0    | 0    | 1
1    | 0    | 0
```

Ordinal Encoding:

```
Grade: [A, B, C, A, B]
→ [1, 2, 3, 1, 2]
```

---

5. Handling Imbalanced Data

The Problem:

When one class has much more samples than others

```
Class Distribution:
- Normal transactions: 9,900 (99%)
- Fraudulent transactions: 100 (1%)
```

Solutions:

1. Undersampling:

- Keep all minority class samples
- Randomly select equal number from majority class

2. Oversampling:

- Keep all samples
- Duplicate minority class samples

3. SMOTE (Synthetic Minority Over-sampling):

- Generate synthetic samples for minority class

---

6. Feature Engineering

6.1 Feature Selection

Goal: Remove irrelevant/redundant features

Methods:

1. **Correlation Analysis:** Remove highly correlated features
2. **Information Gain:** Keep features that best separate classes
3. **Domain Knowledge:** Use expert knowledge

6.2 Feature Creation

Goal: Create new meaningful features

Examples:

- BMI = Weight / $(Height)^2$
- Total_Score = Assignment + Lab + Exam
- Age_Group = Binned age values

---

7. Sampling Techniques

When to Sample:

- Dataset too large to process
- Need balanced training/test sets
- Want to reduce computational cost

Methods:

1. Simple Random Sampling:

- Every sample has equal chance of selection
- Risk: May miss rare classes

2. Stratified Sampling:

- Sample proportionally from each class
- Better representation of all classes

Example:

```
Original: 1000 samples (800 Class A, 200 Class B)
Stratified 10% sample: 100 samples (80 Class A, 20 Class B)
```

---

8. Data Integration

Challenges:

- Different data formats
- Attribute name mismatches
- Duplicate records
- Inconsistent values

Solutions:

1. **Schema Matching:** Align attribute names
2. **Data Cleaning:** Remove duplicates
3. **Format Standardization:** Convert to common format
4. **Entity Resolution:** Identify same entities

---

9. Practical Workflow

Step-by-Step Process:

1. **Understand the Problem**
   - What are you trying to predict/discover?
   - What domain knowledge applies?
2. **Explore the Data**
   - Check data types, distributions, missing values
   - Visualize with histograms, scatter plots
3. **Clean the Data**
   - Handle missing values
   - Remove/fix outliers and noise
   - Remove duplicates
4. **Transform the Data**
   - Normalize/scale numeric features
   - Encode categorical features
   - Create new features if needed
5. **Handle Imbalanced Data**
   - Use sampling techniques if needed
   - Consider cost-sensitive learning
6. **Select Features**
   - Remove irrelevant features
   - Reduce dimensionality if needed
7. **Validate**
   - Check if preprocessing improved data quality
   - Ensure no information loss

---

10. Common Pitfalls to Avoid

1. **Data Leakage:** Don't use future information to predict past
2. **Overfitting:** Don't overfit preprocessing to training data
3. **Information Loss:** Don't remove too much important information

4. **Inconsistent Processing:** Apply same preprocessing to train/test sets
5. **Ignoring Domain Knowledge:** Always consider what makes sense in your domain

---

Key Takeaways

✅ **Pre-processing is crucial** - Often 80% of data science work ✅ **No one-size-fits-all** - Choose techniques based on your data and problem ✅ **Domain knowledge matters** - Understand your data's context ✅ **Validate your choices** - Check if preprocessing improves results ✅ **Document everything** - Keep track of all transformations applied