Data Pre-processing

Classification Methods

Daniel E. Acuna
Associate Professor, University of Colorado Boulder

Contents of This Video

In this video, we will cover:

- Why preprocessing is essential for classification
- Scaling numeric features for fair comparison
- Encoding categorical variables
- Proper train/validation/test splits
- Building preprocessing pipelines
- Avoiding data leakage

The Student Success Dataset Challenge

Raw Student Data Issues:

- **Mixed feature scales:** Study hours (0-40), GPA (0-4), Attendance (0-1)
- **Categorical variables:** Major, study time preference, study space
- **Missing values:** Some students don't report all features
- **Different units:** Hours vs. percentages vs. counts

Without Preprocessing: Models may focus on wrong features or fail entirely

Example Scale Differences: - Study Hours: 5-40 (mean ≈ 22) - GPA: 2.0-4.0 (mean ≈ 3.0) - Attendance: 0.4-1.0 (mean ≈ 0.7) - Assignments: 0-10 (mean ≈ 5)

Feature Scaling: The Foundation

Standardization (Z-score):

$$x_{scaled} = \frac{x - \mu}{\sigma}$$

Min-Max Scaling:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Why Scale? - KNN: Distance calculations dominated by large features - Logistic Regression: Faster convergence - All methods: Fair feature comparison

Scaling Effects: - **Original Data:** Study hours (10-40) vs Attendance (0.5-1.0) - **Standardized:** Both features centered around 0 with unit variance - **Min-Max Scaled:** Both features scaled to 0-1 range

Encoding Categorical Variables

One-Hot Encoding Process:

Original Categorical Data: - Major: Engineering, Business, Liberal Arts, Science - Study Time: Morning, Afternoon, Evening, Night
- Study Space: Library, Dorm, Coffee Shop, Home

One-Hot Encoded Result: - Major_Engineering: 0 or 1 - Major_Business: 0 or 1 - Major_Liberal_Arts: 0 or 1 - Major_Science: 0 or 1

Key Benefits: - No artificial ordering between categories - Each student gets exactly one “1” per categorical feature - Models can learn different weights for each category

Proper Data Splitting

Three-Way Split:

- **Training Set (60%):** Fit the model
- **Validation Set (20%):** Tune hyperparameters
- **Test Set (20%):** Final performance evaluation

Critical Rules:

- Split BEFORE any preprocessing
- Never let test data influence model decisions
- Stratify to maintain class balance
- Use same splits across all models for fair comparison

Workflow: Train → Validate → Test (Fit Model → Tune Parameters → Evaluate Performance)

Preprocessing Pipelines

Student Success Prediction Pipeline:

1. **Raw Student Data** → 2. **Train/Val/Test Split** → 3. **Preprocessing** → 4. **Model Training** → 5. **Clean Data Ready for ML**

Key Preprocessing Steps: - Scale numeric features - Encode categorical variables - Handle missing values - Feature selection

Pipeline Benefits: - **Fit Preprocessor (Training Only):** Learn scaling parameters from training data - **Transform All Splits:** Apply same transformations to validation and test sets - **Consistent Processing:** Same steps applied in training and deployment - **Prevents Data Leakage:** Test data never influences preprocessing decisions

Avoiding Data Leakage

Data Leakage Examples:

- **Scaling using all data:** Test statistics influence preprocessing
- **Feature selection on full dataset:** Choosing features based on test performance
- **Target encoding with all data:** Using test outcomes for encoding

Consequences: - Overly optimistic performance estimates - Models that fail in real deployment - Invalid scientific conclusions

Performance Impact: - **Proper Validation:** Realistic accuracy around 75% - **Data Leakage:** Falsely inflated accuracy around 85% - **Reality Check:** Leakage gives falsely high performance!

What We've Covered

In this video, we've explored:

- The importance of preprocessing for fair feature comparison
- Feature scaling techniques: standardization and min-max scaling
- One-hot encoding for categorical variables
- Proper data splitting strategies
- Building robust preprocessing pipelines
- Avoiding data leakage pitfalls