UNIVERSITÀ
DI PARMA

# Impact of Data Preprocessing on Neural Network Performance: A Comparative Analysis

Saverio Mattia **Merenda**
saveriomattia.merenda@studenti.unipr.it

Fondamenti di Intelligenza Artificiale
[2024-2025]

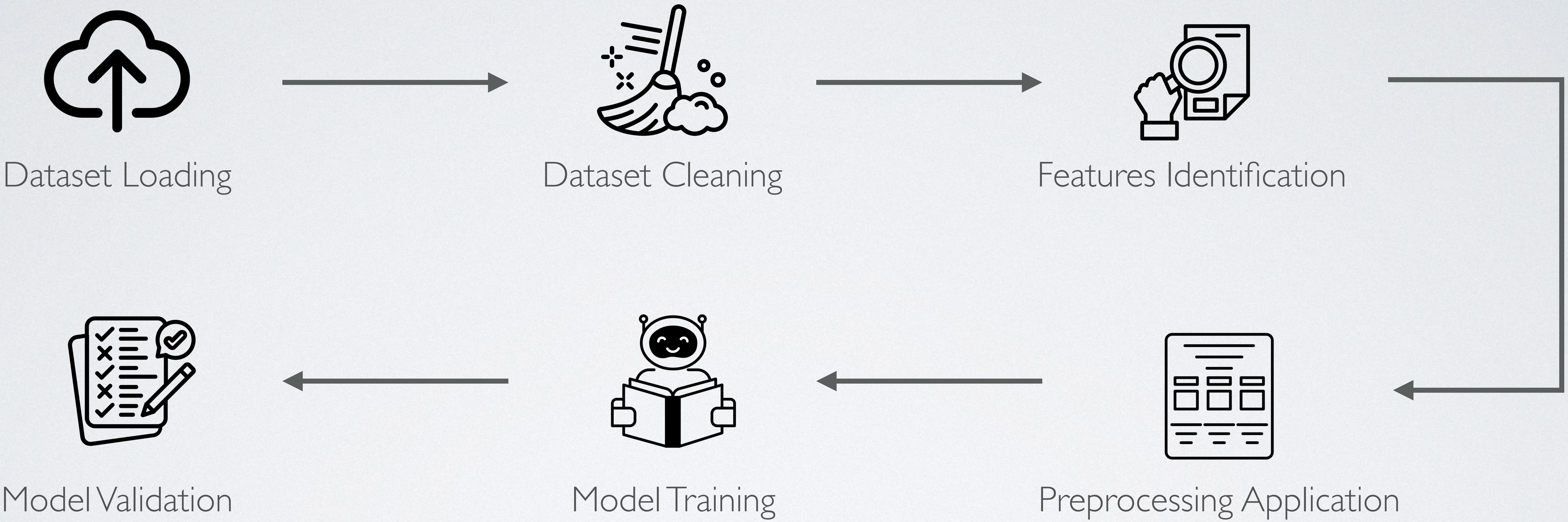
# Overview

- **Objective:** Study how neural network effectiveness varies when using clean datasets and unfiltered datasets
- **Presentation Overview:**
  - Context and motivations for data preprocessing
  - Preprocessing pipeline architecture
  - Different preprocessing scenarios
  - Comparative results

# Why Data Quality Matters?

- **Data preprocessing** is a crucial step that directly impacts neural network performance.
- **Poor quality data can lead to:**
    - Reduced Accuracy: Inconsistent and noisy data leads to poor model predictions
    - Training Instability: Missing values and outliers cause convergence issues
    - Bias Introduction: Improper handling of data can create systematic biases
- **Common Dataset Problems:**
    - Missing Values: "?", "nan", "NaN", empty cells
    - Outliers: Extreme values that skew distributions
    - Inconsistent Scaling: Features with different ranges

# Pipeline Workflow



# Scenario: NaN Values Removal

- **Advantages:**
  - Simple and straightforward approach
  - No assumptions about missing data patterns
  - Clean dataset with complete information
- **Disadvantages:**
  - Potential significant data loss
  - May introduce bias if missingness is not random
  - Reduced statistical power

# Scenarios: NaN Imputation Strategies

- **Three Imputation Methods:**
  - Mean Imputation
  - Mode Imputation
  - Median Imputation
- **Advantages:**
  - Maintains dataset size with multiple imputation options
- **Disadvantages:**
  - Can introduce bias and affect variance

# Scenario: Outlier Removal

- Isolation Forest method with various thresholds [1%, 3%, 5%]
- **Advantages:**
  - Improves model stability by removing extreme outliers
- **Disadvantages:**
  - May remove valuable data if the method is too aggressive

# Scenarios: Normalization and Transformation

- Descriptions of **z-score** normalization and **quantile** transformation
- **Advantages:**
    - Normalizes and transforms data for optimal training performance
- **Disadvantages:**
    - Could alter inherent characteristics of the original data

# Neural Network Architecture

- The neural network used is a **feed-forward network** configured with **3 hidden layers**:
  - First hidden layer: `size = max(64, input_size * 2)`
  - Second hidden layer: `size = max(32, input_size)`
  - Third hidden layer: `size = max(16, input_size // 2)`
- Each layer incorporates **Batch Normalization**, **ReLU activation**, and **Dropout**
- **Advantages:**
  - Adapts to the number of features and enhances training stability
- **Disadvantages:**
  - Fixed architecture that might not be optimal for every dataset

# Used Datasets

- Trained for 100 epochs
- **Classification:**
  - Census Income (48k instances): [archive.ics.uci.edu]
  - Bank Marketing (45k instances): [archive.ics.uci.edu]
- **Regression:**
  - Bike Sharing (17k instances): [archive.ics.uci.edu]
  - House Pricing (168k instances): [kaggle.com]

# Classification Metrics

- **Accuracy:** Proportion of correct predictions
- **Precision:** Ratio of true positives to all positive predictions
- **Recall (Sensitivity):** Ability to identify all actual positive cases
- **F1 Score:** Harmonic mean of precision and recall

# Regression Metrics

- **Mean Absolute Error (MAE):** The average absolute difference between actual and predicted values
- **Mean Squared Error (MSE):** The average of squared differences between actual and predictions
- **R Squared ($R^2$):** Represents the proportion of variance explained by the model

# Census Income Results

| Method | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| 01_without_NaN | 0.8513 | 0.8454 | 0.8513 | 0.8465 |
| 02_imputed_mean | 0.8530 | 0.8476 | 0.8530 | 0.8492 |
| 03_imputed_mode | 0.8552 | 0.8499 | 0.8552 | 0.8514 |
| 04_imputed_median | 0.8550 | 0.8496 | 0.8550 | 0.8510 |
| 05_no_outliers_0.01 | 0.8535 | 0.8489 | 0.8535 | 0.8504 |
| 05_no_outliers_0.03 | 0.8525 | 0.8489 | 0.8525 | 0.8503 |
| 05_no_outliers_0.05 | 0.8507 | 0.8466 | 0.8507 | 0.8481 |
| 06_normalized | 0.8511 | 0.8474 | 0.8511 | 0.8488 |
| 07_transformed | 0.8462 | 0.8430 | 0.8462 | 0.8443 |
| 08_normalized_transformed | 0.8452 | 0.8417 | 0.8452 | 0.8431 |
| (AutoML) Light Gradient Boosting Machine | **0.8737** | **0.8737** | **0.8694** | **0.8700** |

# Bank Marketing Results

| Method | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| 01_without_NaN | 0.7461 | 0.7255 | 0.7461 | 0.6929 |
| 05_no_outliers_0.01 | 0.7519 | 0.7313 | 0.7519 | 0.7086 |
| 05_no_outliers_0.03 | 0.7524 | 0.7304 | 0.7524 | 0.7111 |
| 05_no_outliers_0.05 | 0.7542 | 0.7325 | 0.7542 | 0.7106 |
| 06_normalized | 0.7526 | 0.7327 | 0.7526 | 0.7096 |
| 07_transformed | 0.7497 | 0.7289 | 0.7497 | 0.7033 |
| 08_normalized_transformed | 0.7530 | 0.7325 | 0.7530 | 0.7096 |
| (AutoML) Gradient Boosting Classifier | **0.9451** | **0.9451** | **0.9361** | **0.9384** |

# Bike Sharing Results

| Method | MSE | MAE | $R^2$ |
|---|---|---|---|
| 01_without_NaN | 732.0428 | 17.9583 | 0.9849 |
| 05_no_outliers_0.01 | 1262.5328 | 25.7145 | 0.9740 |
| 05_no_outliers_0.03 | 1323.4792 | 24.9745 | 0.9727 |
| 05_no_outliers_0.05 | 1737.3779 | 28.8209 | 0.9640 |
| 06_normalized | 11559.1484 | 83.1329 | 0.7615 |
| 07_transformed | 12692.0869 | 85.6862 | 0.7381 |
| 08_normalized_transformed | 15285.6006 | 95.9158 | 0.6843 |
| (AutoML) Linear Regression | **0.0** | **0.0** | **1.0** |

# House Pricing Results

| Method | MSE | MAE | $R^2$ |
|---|---|---|---|
| 01_without_NaN | 9.893339e+14 | 1.444670e+07 | -0.2592 |
| 05_no_outliers_0.01 | 7.572025e+14 | 1.376704e+07 | -0.3129 |
| 05_no_outliers_0.03 | 6.610446e+14 | 1.334692e+07 | -0.3506 |
| 05_no_outliers_0.05 | 6.028811e+14 | 1.296907e+07 | -0.3639 |
| 06_normalized | 4.179000e-01 | 2.709000e-01 | 0.5519 |
| 07_transformed | **1.296000e-01** | **2.722000e-01** | **0.8705** |
| 08_normalized_transformed | 1.501000e-01 | 2.942000e-01 | 0.8509 |
| (AutoML) Light Gradient Boosting Machine | 1.143050e+14 | 3.591056e+06 | 0.8545 |

# Key Findings

- **Most Effective Techniques**
  - Mode/Median Imputation: Better than mean imputation
  - Moderate Outlier Removal: 3-5% thresholds optimal
  - Quantile Transformations: Essential for skewed data
- **Less Effective Techniques**
  - Simple Normalization: Often degraded performance
  - Complex Combinations: No guaranteed improvements
  - Aggressive Preprocessing: Can remove valuable patterns

# AutoML & Future Research

- **AutoML Growing Adoption**
  - Industry-wide adoption: Major cloud providers (AWS, Google, Azure) integrate AutoML
  - Democratization of ML: Non-experts can now build effective models
  - Preprocessing automation: Growing focus on automated data preparation pipelines
- **Future Research Directions**
  - Adaptive preprocessing: Automatic technique selection based on data characteristics
  - Intelligent quality assessment: Automated metrics for preprocessing necessity
  - Context-aware pipelines: Real-time adaptation to dataset patterns

UNIVERSITÀ DI PARMA

# Impact of Data Preprocessing on Neural Network Performance: A Comparative Analysis

Saverio Mattia **Merenda**
saveriomattia.merenda@studenti.unipr.it

Fondamenti di Intelligenza Artificiale
[2024-2025]