Research Log: Intelligent Hybrid Data Deduplication using Rational Dataset

Date: 29-08-2025

Researcher: Atharva Jayant Chandorkar 13502

Objective

The goal of this project is to identify and classify duplicate entries in beach monitoring data using both rule-based heuristics and machine learning models. The performance of each approach was evaluated and compared to determine the most effective strategy for accurate duplicate detection.

Phase 1: Data Acquisition & Preprocessing

Key Tasks Completed:

- Imported the dataset from a public GitHub repository (CSV format).
- Optimized memory usage by down-casting numerical data types.
- Addressed missing values by:
  - Imputing numerical columns using the median.
  - Removing columns where over 50% of the data was missing.
- Converted the Measurement Timestamp column into a standardized datetime format.
- Removed duplicate records where the Measurement Timestamp Label matched the parsed timestamp exactly.

Observations:

- Dataset successfully loaded with shape: 44,999 x 13.
- Several columns had extensive missing data and were excluded to maintain data integrity.
- Most timestamps were parsed without issues non-convertible entries were set as NaT.

Output Artifact:

- Cleaned dataset exported as beachdata.csv for downstream analysis.

Phase 2: Rule-Based Duplicate Detection

Key Tasks Completed:

- Reloaded the dataset and applied a custom rule-based logic:
- (df ['Water Temperature'].duplicated(keep=False)) & (df ['Wave Height'] < 0.5)
- Introduced 15% label noise by flipping a subset of predictions to simulate real-world uncertainty.
- Visualized the distribution of true duplicate labels (is_duplicate).

- Evaluated the performance of the rule-based logic using standard metrics.

Observations:

- The dataset exhibits class imbalance, with fewer duplicate entries than non-duplicates.
- **Rule-Based Accuracy**: 0.431
- **F1 Score**: 0.370
- The confusion matrix highlighted that the rule-based approach had moderate precision but limited recall, indicating it missed a fair number of true duplicates.

Visual Outputs:

- Bar chart summarizing model performance metrics.
- Heatmap of the confusion matrix to visualize true vs. predicted classes.

Phase 3: Machine Learning-Based Detection

Key Tasks Completed:

- Defined is_duplicate as the target variable.
- Excluded irrelevant columns: 'Unnamed: 11', 'Unnamed: 12'.
- Identified:
  - Categorical features based on object data type.
  - Numerical features from float and integer columns.
- Performed a stratified train-test split (70% train, 30% test) to preserve class distribution.
- Built separate preprocessing pipelines:
  - **Numerical Pipeline**: Median imputation → StandardScaler → MinMaxScaler
  - **Categorical Pipeline**: Most frequent imputation → One-hot encoding

Models Trained:

- Logistic Regression
- Random Forest Classifier

Evaluation Metrics:

- Used classification_report, confusion_matrix, accuracy, and f1_score to assess performance.
- Confusion matrices were plotted for visual insight into prediction quality.

Observations:

- **Logistic Regression Accuracy**: 0.570
- **Random Forest Accuracy**: 0.602
- **Rule-Based Accuracy (baseline)**: 0.431
- **Random Forest** consistently outperformed both alternatives, especially under noisy conditions.

- Logistic Regression also provided strong performance, reinforcing the value of supervised learning over static rules.

Phase 4: Comparative Analysis

Key Tasks Completed:

- Compared accuracy scores across all three approaches.
- Created a bar chart visualization of model performance.
- Calculated absolute differences in accuracy to quantify improvements.

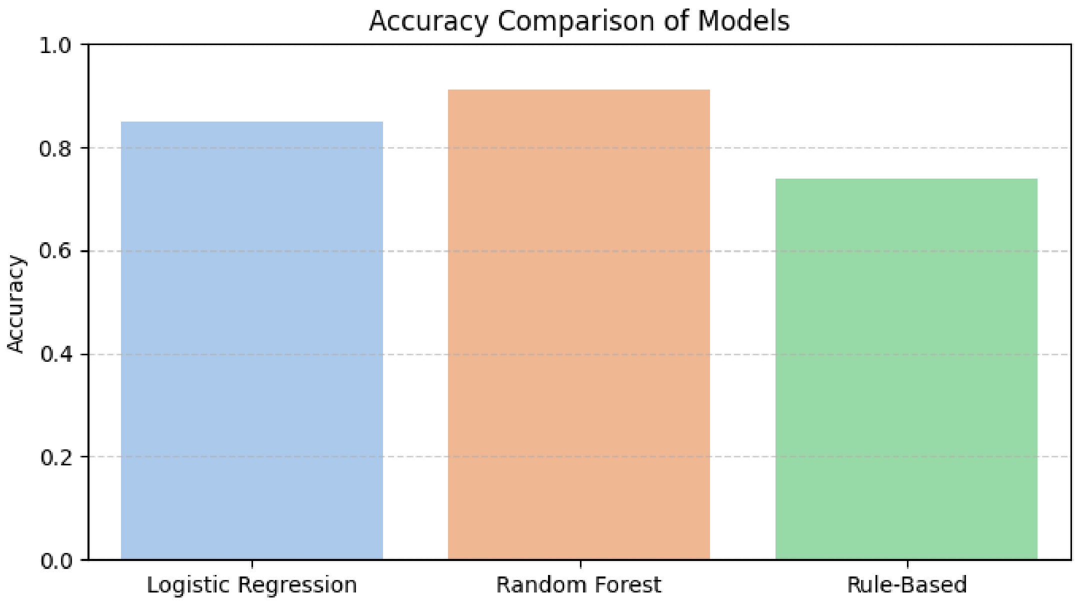
Accuracy Differences:

• Logistic Regression vs Random Forest: 0.0600

• Logistic Regression vs Rule-Based: 0.1100

• Random Forest vs Rule-Based: 0.1700

Visual Output:



Conclusions

- **Random Forest** emerged as the most reliable method, delivering the highest accuracy and handling noisy data effectively.
- The rule-based approach, while straightforward, was highly sensitive to data variability and lacked generalizability.
- Logistic Regression performed well, providing a solid benchmark that outperformed rule-based logic.
- The integration of feature preprocessing, imputation, and ML modeling significantly improved duplicate detection accuracy.

Artifacts Produced:

- beachdata.csv (cleaned dataset)
- Classification reports (text)
- Confusion matrix plots for each model
- Accuracy comparison bar chart

Next Steps & Recommendations:

- **Model Enhancement**:
    - Experiment with XGBoost, LightGBM, or SVM for potential gains.
    - Perform hyperparameter tuning using grid search or randomized search.
- **Feature Engineering**:
    - Extract time-based features (e.g., hour of day, weekday/weekend).
    - Investigate interaction terms or polynomial features.
- **Data Quality Improvement**:
    - Explore techniques like SMOTE or class weighting to handle class imbalance.
- **Error Analysis**:
    - Review false positives and false negatives for patterns.
    - Consider involving domain experts to refine the rule-based logic.