Year-wise Grant to School Students of Class (6–10)

Dataset Source: *Kaggle – Year-wise Grant to School Students of Class (6–10)*

Project Overview

This project focuses on the preprocessing and preparation of educational grant data for students in classes 6–10. The dataset provides insights into year-wise grants distributed to school students, serving as a foundation for future data analysis and machine learning tasks.

In this phase, we have performed comprehensive data preprocessing to ensure data quality, integrity, and readiness for analytical modeling.

Major Tasks in Data Preprocessing

1. Data Cleaning

- We focused on improving data quality and consistency through:
- Filling in missing values using appropriate imputation techniques.
- Smoothing noisy data to remove irregularities and inconsistencies.
- Identifying and removing outliers that distort the dataset.
- Resolving inconsistencies in naming, formatting, and categorical labels.

2. Data Reduction

- To optimize performance and storage, we applied several data reduction techniques:
- Dimensionality Reduction: Removed redundant or less significant attributes.
- Numerosity Reduction: Aggregated and summarized data to reduce record count while maintaining essential information.
- Data Compression: Utilized encoding and compact formats to minimize data size.

3. Data Transformation & Discretization

- We transformed and standardized data to make it more suitable for analysis:
- Normalization: Scaled numeric features to a common range for better comparison.
- Concept Hierarchy Generation: Grouped attributes into meaningful hierarchies (e.g., year ranges, class categories).
- Data Discretization: Converted continuous data into categorical bins to simplify analysis.

4. Data Aggregation

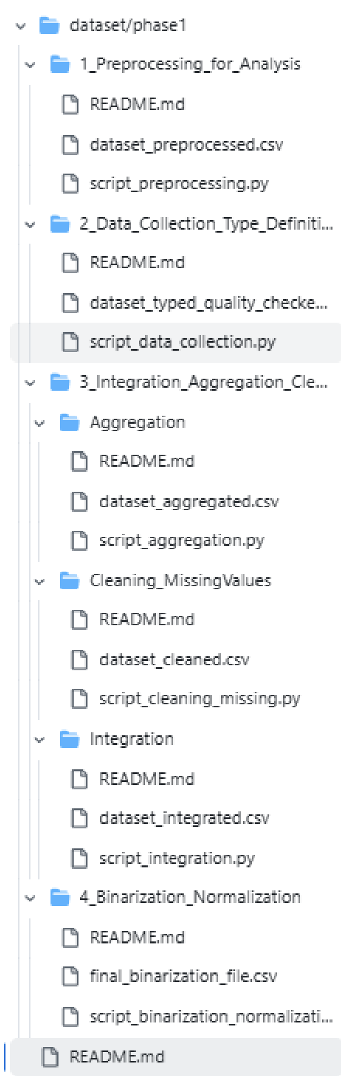
After preprocessing, data aggregation was performed to derive summarized views — combining records by year, class, and gender to identify overall trends and funding patterns.

Objectives

- Clean, structure, and prepare the dataset for future analysis.
- Enable efficient visualization and trend identification.
- Build a solid base for predictive modeling and grant distribution optimization.

Repository Structure

This repository is organized into multiple folders, each representing a specific phase of data preprocessing. Every phase contains a README.md for documentation, a .csv dataset output, and a .py script used for that stage.



Phase Descriptions

This section describes the main objectives and methods for each phase of the data processing pipeline.

1. Preprocessing for Analysis

This folder contains the first stage of the pipeline.
It ensures data consistency, removes errors, and prepares the dataset for later stages.

Includes:

- script_preprocessing.py – performs initial cleaning, encoding, and column formatting.
- dataset_preprocessed.csv – cleaned dataset ready for data typing.
- README.md – describes preprocessing methods and reasoning.

Goals:

- Remove duplicates and nulls;
- Standardize text fields;
- Ensure consistent column names and formats.

2. Data Collection & Type Definition

This step defines and validates the data schema and types.

Includes:

- script_data_collection.py – ensures every column has the correct type (numeric, categorical, etc.);
- dataset_typed_quality_checked.csv – dataset after type correction and validation;
- README.md – documents data typing and validation rules.

Focus:

- Type conversions (string → int/float);
- Detection of invalid values;
- Structural validation of columns.

3. Integration, Aggregation & Cleaning

This is a multi-part phase responsible for unifying, cleaning, and summarizing the dataset.

Integration

Merges multiple sources or datasets into a single consistent dataset.

- Aligns schemas;
- Removes redundancy;
- Produces dataset_integrated.csv.

Cleaning_MissingValues

Handles missing and incomplete data using:

- Mean/median imputation;
- Mode replacement for categories;
- Removal of records with excessive missing values.
  Produces dataset_cleaned.csv.

Aggregation

Performs grouped aggregations by year, class, or gender.
Summarizes total and average grant distributions.
Produces dataset_aggregated.csv.

4. Binarization & Normalization

Final step that prepares data for modeling or analysis.

Includes:

- script_binarization_normalization.py – converts categorical data to binary (0/1) and normalizes numeric features;
- final_binarization_file.csv – standardized dataset ready for analysis;
- README.md – explains binarization and normalization methods.

Techniques applied:

- Min–Max normalization;
- One-hot encoding;
- Feature scaling for comparability.

Summary Table

Phase	**Folder**	**Task**	**Output**
1	1_Preprocessing_for_Analysis	Cleaning, formatting	dataset_preprocessed.csv
2	2_Data_Collection_Type_Definition	Type validation	dataset_typed_quality_checked.csv
3	3_Integration_Aggregation_Cleaning	Integration, cleaning, aggregation	dataset_integrated.csv, dataset_cleaned.csv, dataset_aggregated.csv
4	4_Binarization_Normalization	Normalization and Binarization	final_binarization_file.csv

Results & Findings

- Data Quality Improvement: All missing, inconsistent, and duplicated records were resolved, ensuring a clean and coherent dataset.
- Data Integrity and Structure: Each column now has a clear data type and valid range of values, improving reliability for statistical analysis.
- Unified and Enriched Dataset: The integration phase produced a comprehensive view linking demographic, regional, and educational dimensions.
- Analytical Readiness: After aggregation and normalization, the dataset is ready for visualization and model-based evaluation of fairness and equity.
- Fairness Exploration: The final normalized dataset enables comparisons between groups—by gender, social category, or region—to assess whether scholarships were distributed evenly or unequally.

Interpretation

- Gender-Based Patterns: Comparing totals for male and female students helps detect if one gender systematically receives more or fewer scholarships.
- Social Category Inequality: Analyzing totals by category (Gen, OBC, SC, ST) can expose overrepresentation or underrepresentation of specific social groups.
- Regional and Developmental Gaps: Evaluating aspirational versus non-aspirational districts provides insight into whether underdeveloped regions receive equitable support.
- Class-Level Differences: Observing funding distribution across class levels can reveal if higher or lower grades receive disproportionate amounts of funding.

Key Takeaways

- A clear four-phase pipeline for data preparation.
- Clean and modular Python scripts for each transformation step.
- Progressive improvement in data quality and usability.
- Ready-to-analyze dataset suitable for statistical or ML-based exploration.

Conclusion

The project successfully transformed raw, unstructured educational data into a fully processed, analysis-ready dataset. It enables fair, transparent, and data-driven evaluation of government scholarship distribution. Each phase contributed to improving data accuracy, consistency, and analytical depth, establishing a solid foundation for future equity and policy analysis in the educational domain.