DES646 – Project Report

AI-Driven Sustainable Material Decision Support System Integrating Cost Prediction, User Feedback Analysis, and Multi-Criteria Design Optimization

1. Team Details

Name	Roll No.	Department
Aayush Shailesh Jhawar	230028	Mechanical Engineering
Sanchit Garg	230908	Mechanical Engineering
Gauri Gupta	230415	Earth Science
Aryan Raj	230219	Earth Science

2. Abstract

This project investigates the construction of a comprehensive materials database and the development of a quantitative material scoring framework combining price, embodied carbon, and engineering performance. The final model integrates category-weighted imputation, robust scaling, PCA, clustering, and TOPSIS ranking. The report summarizes data collection, challenges in data cleaning, machine learning methods, design implications, and reflections on scientific decision-making.

3. Introduction

This project aimed to build a unified, searchable, and analysable dataset of engineering materials, and then generate an interpretable composite score to support material selection decisions. Because material data online is fragmented, inconsistent, and often ambiguous, the project focused equally on data engineering and machine learning analysis. The outcome is a structured dataset with 2400+ materials enriched with cost, embodied carbon (CO_2), engineering properties, PCA clusters, and TOPSIS-based ranking.

4. Data Collection

A. Scraping Strategy

- The raw data originated from scraped pages of MatWeb-like sources using Python (BeautifulSoup + Requests).
- The scraper extracted property names, values, units, and metadata per material.
- Multiple issues appeared due to inconsistent formatting, commas inside values, multi-language entries (Metric/English), and overlapping property blocks.

B. Challenges Encountered

Several persistent issues emerged:

1. **Comma-induced column shifts**: Some properties contained commas inside descriptions (e.g. "2.23g/cc @ Temperature 25°C"), causing CSV rows to break.
2. **Mixed Metric/English duplicates**: Many rows contained paired "Metric, English" entries.
3. **Irregular value formats**: Values included ranges, units, additional text, or comments.
4. **Category ambiguity**: Materials belonged to multiple overlapping categories, making imputation difficult.
5. **Missing values**: No material contained every property, leading to extremely sparse columns.

C. Mitigation Strategies

The final dataset required several layers of cleaning:

- Manual validation of column headers, removing descriptive/comment columns that contained only text.
- Regex-based extraction of the *first numeric value* per property.
- Parsing categories into lists for multi-category weighting.
- Removing disruptive fields such as GUID after initial validation.
- Converting all string numerics using controlled coercion to `float`.
- Verifying each cleaning step through intermediate CSV exports.

5. Data Cleaning and Imputation

A. Category-Weighted Imputation

Missing values were imputed using a weighted combination of category medians:

$$v_{\text{imputed}} = w_1 m_1 + w_2 m_2 + w_3 m_3,$$

with weights $(0.6, 0.3, 0.1)$ assigned to the first three categories of each material. If no category value existed, the global median was used as the default.

B. Scaling

A RobustScaler (median/IQR) was selected because:

- Properties have heavy-tailed distributions.
- Some outliers represent real materials (e.g., aerogels vs. tungsten).
- MinMaxScaler compressed values too tightly and distorted the variance.

6. Dimensionality Reduction

A. Principal Component Analysis

PCA was performed on engineering properties only (excluding cost and CO_2). Outputs include:

- PC1 (stiffness–density trade-off)
- PC2 (thermal–electrical behaviour)
- Clear clustering separation between ceramics, metals, and polymers

PCA scatter plots were generated for both 2D projection and silhouette-scored cluster validation.

B. KMeans Clustering

K-Means was used to identify material families with unsupervised patterns.

- k chosen by silhouette score (typically 4–6)
- Clusters aligned well with expected material classes
- Outliers (e.g., foams, superalloys) were identifiable

7. Material Scoring (TOPSIS)

Feature Selection

TOPSIS used:

- UTS
- Elastic Modulus
- Strength-to-Weight Ratio
- Specific Stiffness
- Thermal Conductivity
- Density (cost)
- Cost (cost)
- CO_2 footprint (cost)

Weighting Scheme

- Engineering performance: weight = 1.0
- Thermal properties: 0.9
- Cost + CO_2: 0.8
- Others: 0.6

Results Summary

- Distribution is centred without collapse to 1.0 (previous issue fixed).
- Top-ranked materials tend to have high modulus–density ratios and low embodied carbon.
- Bottom-ranked materials are either extremely dense or high-cost/high-CO_2.

8. Visual Analysis

Plots included:

- PCA projections (PC1 vs PC2)
- Correlation matrix
- TOPSIS score histogram
- Ashby charts (E–Density, UTS–Density)
- Cluster distributions

All plots are embedded into the analysis notebook and exported separately.

9. Design Implications

- High-ranking materials are ideal for lightweight structural applications.
- Cost/CO_2 inclusion shifts preference away from traditional metals.
- Ceramics dominate thermal performance clusters.
- Polymers become competitive only after cost/CO_2 normalization.

10. Limitations

- Scraped data may include residual formatting noise.
- Imputation introduces smoothing that may obscure true extremes.
- TOPSIS weights are subjective—different stakeholders could choose different priorities.
- Cost data scraped from online stores is approximate.

11. Reflection

This project required an extensive data cleaning effort, surpassing the ML work in difficulty. The most time-consuming task was diagnosing corrupted CSV rows and fixing structural inconsistencies. The experience underscored the importance of data quality in data science and the necessity of creating transparent data pipelines.

Key lessons:

- Automated scrapers must be designed to tolerate inconsistent HTML.
- Always validate CSVs after every major processing step.
- Real-world datasets rarely align with textbook examples.
- Weighted imputation using domain structure (material families) produces more realistic variability than global or per-column medians.

12. Conclusion

The project successfully delivers a unified materials database, a robust imputation pipeline, interpretable dimensionality reduction, and a final ranking framework integrating cost and environmental impact. The workflow can be extended to other databases, additional properties, or real industrial datasets.

Appendix A: Project File Structure

The project folder contains all intermediate and final artefacts generated during scraping, cleaning, integration, modelling, and analysis. The structure is preserved here for transparency and reproducibility.

A.1 High-Level Overview

The directory is organised into five workflow stages:

- Data scraping and raw collection
- Data cleaning and preprocessing
- Imputation and enrichment (cost & CO_2)
- Machine learning pipeline (PCA, clustering, TOPSIS)
- Analysis outputs and visualisations

- Project Report
- Group logs
- Presentation PPT
- IPDF

Each stage corresponds to specific folders and files listed below.

A.2 Detailed Folder and File Descriptions

1. Machine Learning Pipeline and Analysis

- `ML_pipeline.ipynb` — Full end-to-end modelling notebook including numeric selection, scaling, PCA, clustering, scoring, and TOPSIS.
- `Analysis.ipynb` — Visual analysis of PCA, correlations, distributions, and results interpretation.

2. Final Ranked Outputs

- `materials_ranked.xlsx` — Final TOPSIS-ranked materials with integrated engineering, cost, and CO_2 metrics.
- `materials_ranked_final.xlsx` — Fully cleaned and presentation-ready final ranking.

3. Final Cleaned Datasets

- `final_clean.py` — Script used to generate the final cleaned dataset prior to imputation.
- `materials_final_with_price.xlsx` — Dataset with integrated price and environmental (CO_2) values.
- `dataset_final_imputed.xlsx` — Dataset after weighted category-based imputation.

4. Analysis Exports

- `analysis_outputs/` — Folder containing all exported plots:
  - PCA projections
  - TOPSIS distributions
  - Correlation heatmaps
  - KMeans cluster plots
  - Ashby-style charts

5. Data Cleaning Pipelines

- `Database_cleaning/` — Scripts and interim files used during the multi-stage repair of the scraped dataset. Includes:
  - column alignment checks
  - comma correction utilities
  - regex-based numeric extractors
  - manual anomaly logs

6. Cost and Environmental Integration

- `Cost and Environment data integration/` — Contains scraped and manually verified price and CO_2 lookup tables.

7. Original Raw Data (Scraped)

- `comprehensive_matweb_data.xlsx` — The original raw dataset extracted from web sources, including misaligned and inconsistent rows.
- `scrape.py` — Primary Python scraper used for initial MatWeb-like extraction.
- `matweb_guids_checkpoint.xlsx` — GUID-level checkpoint used to track scrape progress and re-run missing pages.
- `guids.py` — Secondary utility script for GUID parsing.

8. Additional Data Sources (Not Used)

- `Online_databases_not_used/` — External datasets explored but ultimately not included due to incompatible formats or missing metadata.

9. Failed and Deprecated Files

- `Failed/` — Folder containing datasets or scripts that were generated during failed preprocessing attempts (e.g., unfixable delimiter corruption).

A.3 Notes on Reproducibility

Each stage of the workflow outputs a CSV or XLSX checkpoint to ensure traceability. The project is fully reproducible end-to-end by running:

1. `scrape.py`
2. cleaning scripts inside `Database_cleaning/`
3. `final_clean.py`
4. `ML_pipeline.ipynb`
5. `Analysis.ipynb`

This ensures transparency and validates the robustness of the pipeline across multiple iterations.