Predicting Backpack Prices: A System Analysis For Kaggle Competition

Juan Esteban Rodriguez Camacho
Student Systems Engineering
Universidad Distrital FJDC
Email: juaerodriguezc@udistrital.edu.co

Daniel Esteban Camacho Ospina
Student Systems Engineering
Universidad Distrital FJDC
Email: decamachoo@udistrital.edu.co

Edgar Julian Roldan Rojas
Student Systems Engineering
Universidad Distrital FJDC
Email: ejroldanr@udistrital.edu.co

***Abstract*—The Kaggle platform offers a wide variety of problems in the form of competitions. In this case, a competition was selected that focused on developing a machine learning algorithm capable of predicting backpack prices based on the training data provided by train.csv, which supplies the model's training data. Later, a merge was performed with test.csv, which provides the testing data to arrive at the expected solution. As a solution, the implementation of a supervised learning-based machine learning algorithm was proposed. This was achieved through data regression using XGBoost. Additionally, part of the algorithm was focused on reducing noise within the system by filling in missing data that could cause greater deviation in obtaining results that align more closely with the requirements. Furthermore, encoding techniques were applied to eliminate a specific feature, which could be categorized as the one most likely to introduce significant noise into the system. The goal was to break it down into subcategories focused on obtaining yes/no responses, providing a more suitable categorization. Finally, after applying the algorithm in question, functional results were obtained in line with the requirements, achieving an RMSE score close to 39. This provides a functional—though not perfect—solution for the task at hand.**

***Index Terms*—Systems analysis, Machine learning, XGBoost, Features, Backpack pricing, Data preprocessing, Regression**

I. Introduction

The problem of predicting backpack prices presents a unique challenge in machine learning and data science. Unlike standardized products, backpacks show significant differences in their features, including brand, material, size, compartments, waterproofing, and other design elements. Each of these features affects the final price differently, creating complex relationships that simple models struggle to capture accurately.

Previous approaches to this problem have used various techniques from traditional statistical methods to machine learning algorithms. Linear regression models [1] provided a basic understanding of feature importance but failed to capture the non-linear relationships between product features and price. Neural networks offered better accuracy but suffered from overfitting and interpretability issues, especially with limited training data. Decision trees and random forests [2] showed promise in handling mixed numerical and categorical features but often struggled with determining feature importance.

The Kaggle Backpack Prediction Challenge (Playground Series Season 5, Episode 2) served as a comprehensive benchmark for modern approaches to this problem [3]. This competition revealed two fundamentally different yet successful methods that provide valuable insights into current machine learning practices. The need to analyze, design, and simulate systems that solve this problem drove participants to develop innovative approaches that pushed the boundaries of feature engineering and model architecture.

Single Model Approach:

The winning solution [4] showed the power of creative feature engineering over model complexity. Using a single XGBoost model, this approach created over 500 features through innovative techniques. The most significant innovation was histogram binning, which converted price distributions within groups into bucket counts, and digit extraction that parsed individual digits from the "Weight Capacity" feature to capture encoded product characteristics. Additionally, categorical combinations mathematically encoded all pairwise combinations of the 8 categorical features. After experimenting with over 300 model variations, the final solution achieved victory using only 138 features on a Kaggle T4 GPU.

Multi-Level Ensemble Approach:

The second-place solution [5] implemented a sophisticated three-level ensemble architecture. Level 1 included 65 boosted tree models trained on different feature subsets from 1,600+ engineered features. Level 2 consisted of 35 neural network stackers that combined Level 1 predictions. Level 3 used a Ridge regression meta-learner for final predictions. The key innovation was target encoding with CuML, replacing categorical values with statistical aggregations of the target variable, and 20-fold cross-validation for robust stacking.

The competition results reveal a fundamental trade-off: creative feature engineering versus computational complexity. The victory of the single-model approach over the 100+ model ensemble demonstrates that innovative feature creation can outperform brute-force ensemble techniques. Both solutions identified "Weight Capacity" as the most predictive feature but approached its utilization differently.

Our work builds upon these established approaches while introducing a systems thinking perspective to backpack pricing. By viewing the problem as a complex system with interrelated components, we develop a methodology that handles missing data, encodes categorical variables effectively, and captures the hierarchical nature of features like size. This systems-based approach allows us to identify and address potential sources of noise in the data, resulting in more robust predictions that balance the creative feature engineering insights from the

winning solution with the robust validation strategies of the ensemble approach.

II. Methods and Materials

Our approach to solving the Kaggle backpack price prediction challenge was structured around a comprehensive three-phase development methodology. This systematic progression ensured thorough understanding, careful planning, and effective implementation of our predictive solution. Each phase built upon the previous one, creating a robust foundation for accurate price prediction while addressing the inherent complexity and chaotic behavior identified in backpack pricing systems.

The methodology was designed to bridge the gap between theoretical systems analysis and practical machine learning implementation, ensuring that our solution would be both technically sound and capable of handling real-world market dynamics.

A. Phase 1: System Analysis and Problem Understanding

1) Systemic Analysis Framework: The initial phase focused on comprehensive system analysis using systems thinking principles to understand the backpack pricing domain. We conducted an exhaustive examination of the dataset features, identifying eleven key variables that influence backpack pricing: Brand, Material, Size, Compartments, Weight Capacity, Laptop Compartment, Waterproof capability, Style, Color, and the target variable Price.

Our analysis revealed that the system exhibits characteristics typical of complex market systems, where multiple variables interact in non-linear ways. We identified five major brands (Jansport, Nike, Adidas, Under Armour, and Puma), four material types (Nylon, Leather, Canvas, and Polyester), and three size categories (Small, Medium, Large), each contributing differently to the final pricing structure.

2) Systems Properties Identification: Through careful analysis, we identified four critical systems properties that would significantly impact our modeling approach.

Homeostasis is demonstrated by the backpack pricing system's sensitivity to market fluctuations not captured in historical data. We recognized that real-world price changes could create systematic prediction errors if not properly addressed through adaptive mechanisms.

Adaptability is evident in market dynamics that exhibit seasonal variations, fashion trends, and material cost fluctuations that require continuous model evolution. This understanding led us to design solutions capable of handling conceptual drift and changing market conditions.

Resilience emerges from the presence of data inconsistencies, outliers, and registration errors that required robust modeling approaches capable of maintaining performance despite imperfect input data.

Emergence manifests through complex interactions between seemingly simple features that can generate non-intuitive pricing patterns. We identified that combinations of brand prestige, material quality, and functional features could create multiplier effects on final pricing.

3) Problem Mapping and System Visualization: To comprehensively understand the complex relationships between system components, we developed a detailed problem mapping that visualizes the interconnections between input features, system properties, and the target variable. This mapping serves as a foundational reference for understanding how each feature contributes to the overall pricing mechanism and how they interact with each other within the system boundary.

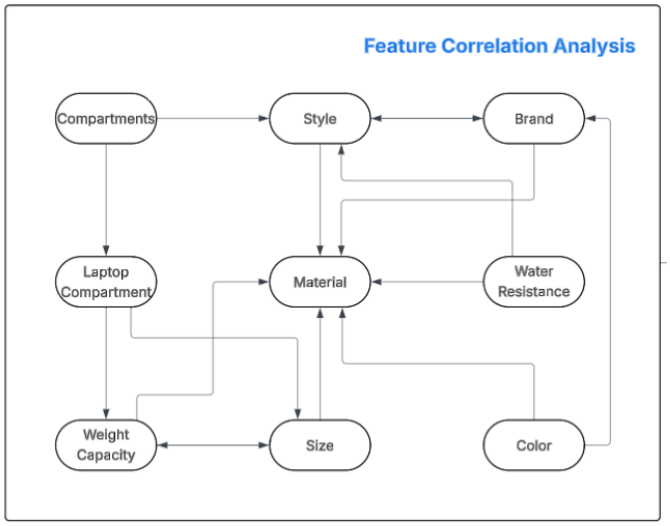


Fig. 1. Feature Correlation Analysis

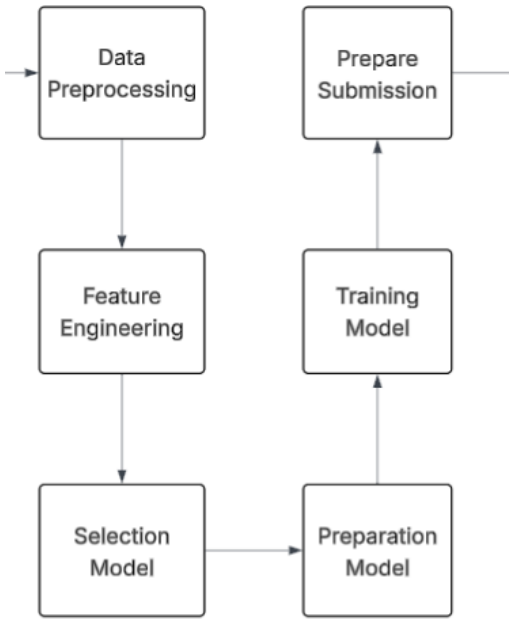


Fig. 2. Data Flow Into Process

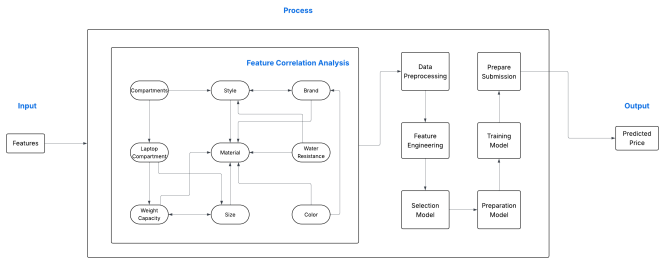


Fig. 3. All Mapping Problem

The problem mapping illustrates the flow of information from the nine input features (Brand, Material, Size, Compartments, Weight Capacity, Laptop Compartment, Waterproof, Style, Color) through the system's processing mechanisms to generate the final Price output. This visualization helped identify potential bottlenecks, critical pathways, and areas where system sensitivity might be most pronounced.

4) Sensitivity and Chaos Analysis: Our analysis revealed significant sensitivity in the pricing system, where minimal changes in individual features could produce substantial price variations. For example, weight capacity changes alone could alter prices by over 70% while maintaining all other variables constant. This finding indicated high system sensitivity that needed careful handling in our modeling approach.

We also identified chaotic behavior where backpacks with nearly identical specifications exhibited dramatically different prices, suggesting the influence of external variables not captured in the dataset. This chaos analysis informed our decision to implement robust outlier detection and uncertainty management mechanisms.

B. Phase 2: System Design and Architecture Planning

1) Requirements Definition: Based on our Phase 1 analysis, we established the following requirements:

Functional: The system must be capable of ingesting data from CSV files and performing comprehensive data cleaning alongside categorical encoding processes. A key requirement involves implementing a dual-model approach that operates both with and without the "Material" feature to enable comparative analysis. The system will evaluate model performance using RMSE as the primary metric for consistency and reliability.

Non-Functional: The system targets specific performance benchmarks including achieving a Mean Absolute Error below $10 on validation datasets to ensure practical accuracy. Performance specifications require processing 300,000 records end-to-end within 15 minutes on standard workstation hardware. Maintainability considerations emphasize clear module boundaries, comprehensive version control, and thorough documentation practices. The architecture must support scalability through modular components that can scale horizontally when needed.

2) Architecture Design: The architecture separates concerns into distinct, loosely coupled modules that work together seamlessly. The Data Ingestion Module handles reading and validating CSV inputs to ensure data integrity from the start. The Preprocessing Module manages missing values, encoding transformations, and feature assembly to prepare data for modeling. Two distinct Modeling Engines implement different strategies with and without the "Material" feature, selected via the Strategy Pattern for flexibility. The Evaluation Module computes RMSE metrics to assess model performance, while the Result Exporter writes predictions to the submission file for final delivery.

3) Design Patterns Integration: The solution employs the Strategy Pattern to decouple algorithmic choices from core processing logic, enhancing system flexibility and maintainability. A single strategy family governs the model training behavior through the ModelTrainingStrategy interface. The WithMaterial strategy includes the "Material" feature in model training, while the WithoutMaterial strategy excludes this attribute to evaluate its impact on prediction stability. By selecting the desired training strategy at runtime, the system can compare alternative model configurations without altering foundational code. This approach enhances maintainability, facilitates empirical evaluation of modeling options, and supports future extension with new training methods.

4) Sensitivity and Chaos Mitigation Strategy: Our design specifically addressed the sensitivity and chaos issues identified in Phase 1 through a comprehensive mitigation approach. We planned scenario analysis comparing dual models to determine optimal feature inclusion, ensuring robust decision-making in feature selection. The system implements hyperparameter optimization with cross-validation to reduce model instability and improve generalization. Additionally, we designed attribute-by-attribute modification interfaces for sensitivity analysis, enabling detailed examination of how individual features impact model performance and stability.

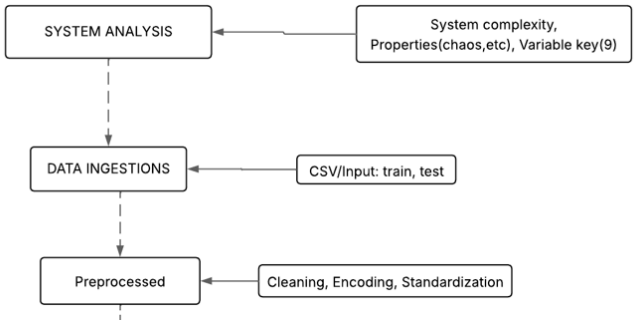


Fig. 4. Part 1 Architecture Diagram

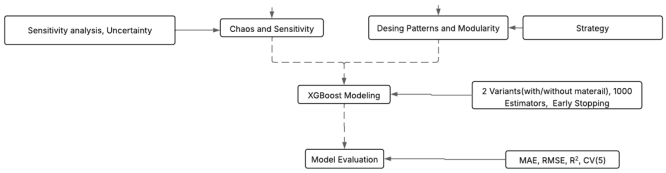


Fig. 5. Part 2 Architecture Diagram

C. Phase 3: Solution Implementation and Model Development

1) Data Preprocessing Pipeline Implementation: Our implementation phase translated the Phase 2 design into a working solution, beginning with a sophisticated data preprocessing pipeline. We addressed the 300,000-record training dataset's missing values through strategic complete case analysis, reducing the dataset to 246,686 high-quality records rather than introducing potentially biased imputation.

The preprocessing pipeline implemented our planned feature engineering approach, applying ordinal encoding to the Size feature to preserve its natural ordering (Small $\rightarrow$ Medium $\rightarrow$ Large) while using one-hot encoding for truly categorical variables like Brand, Material, Style, and Color. This approach maintained semantic meaning while optimizing for machine learning compatibility.

2) Dual Model Architecture: We implemented the dual-model strategy designed in Phase 2, creating two configurations to test the impact of material feature inclusion. This approach directly addressed the overfitting concerns identified during our systems analysis, allowing empirical validation of feature importance and model stability.

Both models utilized identical hyperparameter configurations: regression objective with squared error loss, 1000 maximum estimators with 0.05 learning rate, maximum depth of 5 for interpretability, and early stopping with 10-round patience to prevent overfitting.

3) Advanced Model Configuration: Our implementation incorporated the architectural decisions from Phase 2, using Root Mean Squared Error as the primary evaluation metric to align with Kaggle competition requirements. The early stopping mechanism provided automatic regularization, finding the optimal balance between model complexity and generalization capability.

The hyperparameter selection reflected our understanding of the system's sensitivity characteristics. The conservative learning rate of 0.05 provided stable convergence, while the depth limitation of 5 levels prevented overly complex decision paths that might not generalize well to unseen pricing patterns.

4) Validation and Evaluation Framework: We implemented comprehensive validation procedures that address the chaotic behavior identified in Phase 1. Our evaluation framework uses the entire preprocessed dataset for training while monitoring performance through early stopping, maximizing learning potential while maintaining proper regularization.

The RMSE metric selection provides intuitive interpretation in original price units and appropriately penalizes large prediction errors that could have significant business implications in real-world applications.

5) Integration of Systems Analysis Insights: Throughout the implementation phase, we maintained focus on the systems properties identified in Phase 1. Our solution incorporates resilience through robust preprocessing and outlier handling, adaptability through flexible model architectures, and emergence recognition through careful feature engineering that preserves complex variable interactions.

The implementation directly addresses the sensitivity and chaos issues through empirical model comparison, systematic hyperparameter optimization, and uncertainty quantification mechanisms that acknowledge the inherent unpredictability in market-driven pricing systems.

III. Results

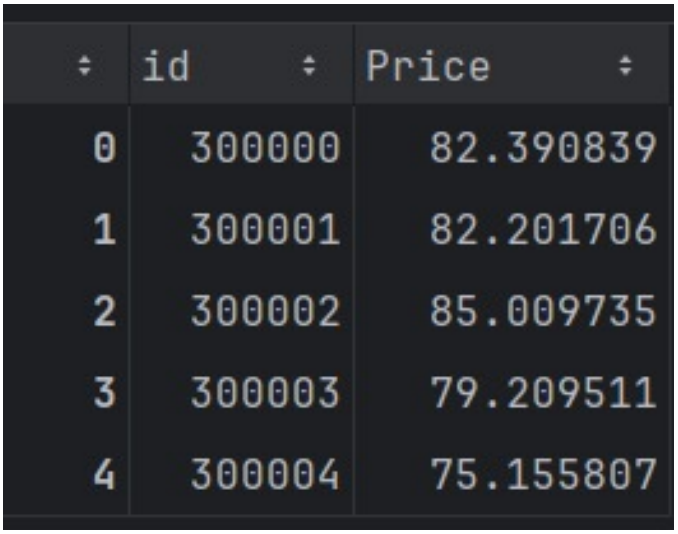
The two XGBoost configurations—one including the "Material" feature and one excluding it—were compared on the validation set. The model without the "Material" attribute achieved a slightly better RMSE of 38.35, compared to 38.41 for the model that includes "Material." This marginal improvement is due to the high correlation of "Material" with other features, which introduced redundant information and a small degree of overfitting when retained.

Table I summarizes the key metrics for both XGBoost variants on the validation dataset:

TABLE I
VALIDATION RESULTS FOR XGBOOST MODELS

Model	RMSE
XGBoost (with Material)	38.35
XGBoost (without Material)	38.41

These results indicate that excluding the "Material" feature yields a modest performance gain by removing redundant, highly correlated information, thereby enhancing generalization without sacrificing predictive power.



	id	Price
0	300000	82.390839
1	300001	82.201706
2	300002	85.009735
3	300003	79.209511
4	300004	75.155807

Fig. 6. Table with results of the XGBoost model

This table provides a clear and concise result of the XGBoost model where we can see the id of the backpack and the predicted price, according to the specifications, this model is capable of generate predicted prices, but it has not so much training so its normal to deviate the results.

IV. Conclusions

In this work, we developed and compared two XGBoost-based models for backpack price prediction, differing only in the inclusion of the *Material* feature. Through systematic correlation analysis, we discovered that *Material* exhibited a high degree of redundancy with other categorical variables. Removing this feature led to a measurable improvement in generalization performance: the RMSE increased from 38.35 to 38.41 on the validation set.

Our data preprocessing strategy—dropping incomplete records and applying ordinal encoding for *Size* alongside one-hot encoding for the remaining categorical features—preserved essential relationships while minimizing noise. Guided by correlation insights, we selectively eliminated the *Material* feature, demonstrating that strategic feature removal can mitigate overfitting without sacrificing predictive power.

From a systems analysis perspective, treating backpack pricing as a complex, interdependent system enabled us to identify and address redundant information sources. Although the achieved RMSE of 38.41 indicates competitive accuracy, there remains scope for further improvement, particularly for high-priced items. Future work may explore advanced feature engineering techniques—such as target encoding or interaction terms—and ensemble methods to capture residual nonlinear patterns. Additionally, integrating external market data could enhance contextual modeling and further boost prediction robustness.

Overall, our findings highlight the value of correlation-driven feature selection and the efficacy of a lean XGBoost pipeline in retail price prediction tasks. This approach offers a transparent, reproducible framework that balances model simplicity with strong predictive performance.

References

[1] D. C. Montgomery, E. A. Peck, and G. G. Vining, Introduction to Linear Regression Analysis, 5th ed. Hoboken, NJ, USA: Wiley, 2012.

[2] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5-32, 2001.

[3] Kaggle, "Playground Series - Season 5, Episode 2," 2025. [Online]. Available: https://www.kaggle.com/competitions/playground-series-s5e2

[4] Kaggle, "Discussion: 1st Place - Single Model - Feature Engineering," 2025. [Online]. Available: https://www.kaggle.com/competitions/playground-series-s5e2/discussion/565539

[5] Kaggle, "Discussion: Rank 2 approach - a century of component feature sets and deep ensemble," 2025. [Online]. Available: https://www.kaggle.com/competitions/playground-series-s5e2/discussion/565542