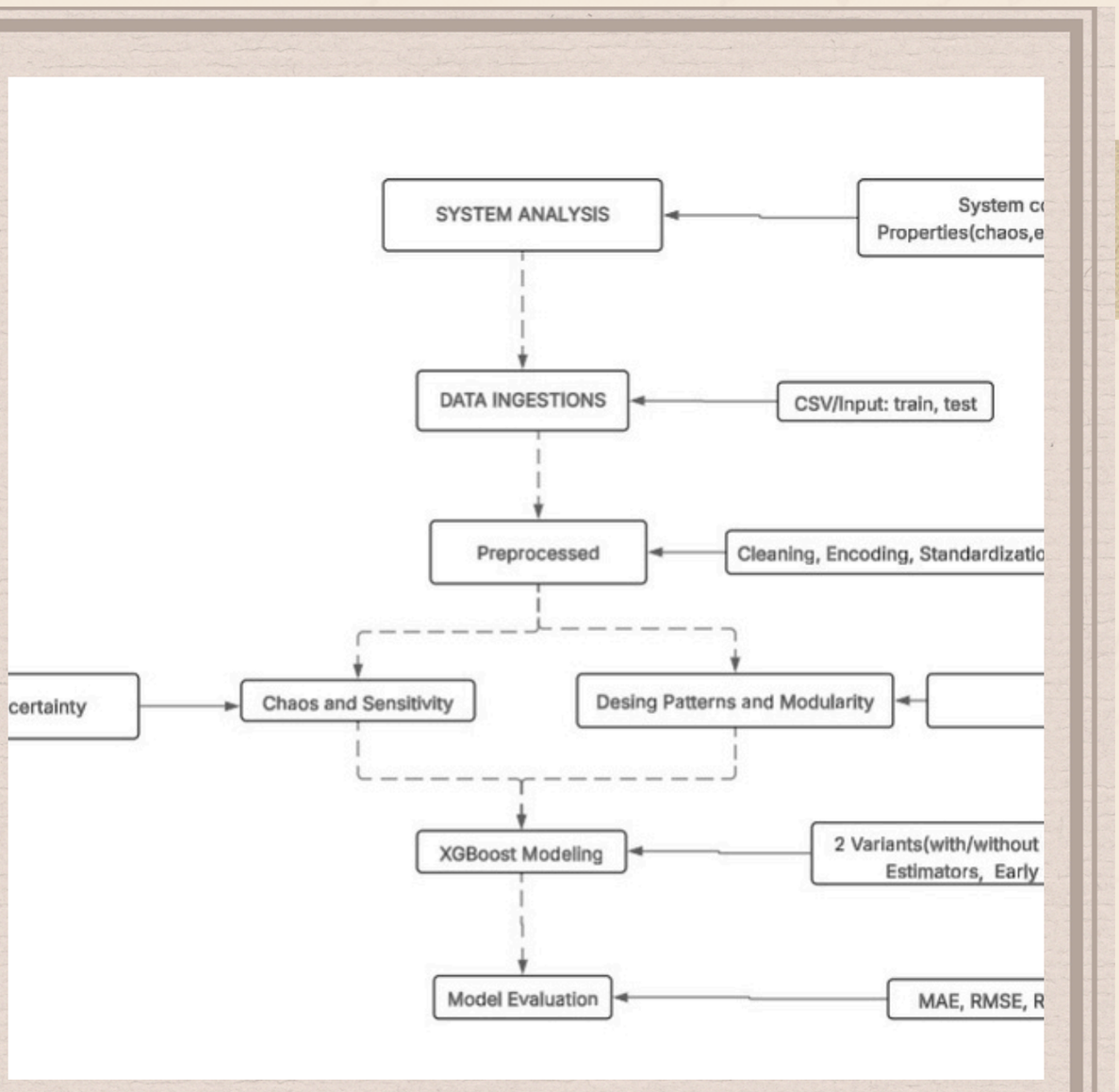
FINAL PROJECT: KAGGLE COMPETITION



INTRODUCTION AND GOALS

- This project addresses the Kaggle competition , where the goal is to predict backpack prices based on product features such as brand, material, size, and weight capacity.
- We aim to build a robust system capable of handling market unpredictability, feature noise, and chaotic behavior using machine learning, systems thinking

ARCHITECTURE

The system design is based on a modular architecture divided into independent stages: data ingestion, robust preprocessing, and dual-model training. It uses the Strategy Pattern to switch flexibly between including or excluding the “Material” feature. This modular approach improves maintainability and allows systematic sensitivity analysis to handle complex market dynamics.
Each module works autonomously to ensure data quality, optimize features and prevent overfitting. The architecture supports future scalability and extension for new variables or improved training strategies.

PROPOSED SOLUTION

The solution adopts an XGBoost regression model to predict backpack prices, addressing nonlinear relationships and chaotic behavior in the dataset. It implements a dual-model strategy to compare performance with and without the “Material” feature, helping to detect redundant information.

A robust preprocessing pipeline removes incomplete records, encodes categorical variables appropriately (ordinal for Size, one-hot for Brand, Material, and Style), and optimizes hyperparameters with early stopping. This ensures generalization, stability, and resilience against data noise.

RESULTS

The final models were tested on a validation set, achieving an RMSE of 38.35 with the “Material” feature and 38.41 without it. This small difference confirms the redundancy of the Material variable and validates the dual-model approach.

The system demonstrates resilience to outliers, adaptability to new market data, and emergent behavior by capturing complex interactions between brand, material, and size. The solution provides a solid baseline for future improvements and integration with external market variables.

```
ss_data(df, test):
dle missing values and encode categorical variables
'Size'] = test['Size'].fillna(test['Size'].mode()[0]
opna(inplace=True)
order = ['Small', 'Medium', 'Large']
er = OrdinalEncoder(categories=[size_order])
ize'] = encoder.fit_transform(df[['Size']])
'Size'] = encoder.transform(test[['Size']])
orical_cols = ['Brand', 'Laptop Compartment', 'Water
le', 'Color']
od.get_dummies(df, columns=categorical_cols)
= pd.get_dummies(test, columns=categorical_cols)
n df, test
```