Data Preprocessing and Feature Engineering

Before model training, extensive preprocessing and feature engineering steps were conducted to prepare the dataset for analysis. The original dataset comprised a mix of textual, numerical, and boolean features.

**3.1 Data Cleaning**

Initial exploratory data analysis (EDA) was performed to understand the structure and quality of the data. Redundant columns that did not contribute meaningful information were removed. Additionally, all rows containing missing values were dropped to ensure the dataset remained clean and consistent for downstream modeling.

**3.2 Target Variable Transformation**

A histogram of the target variable, price, revealed a right-skewed distribution, indicating a concentration of low-price items and a long tail of high-price items. To address this skewness and stabilize variance, a log-transformation was applied to the price column. This transformation enhances model performance by bringing the distribution closer to normality.

**3.3 Feature Engineering**

To improve the predictive capacity of the models, several new features were engineered:

**Color Features**: The color column, which contained textual descriptions, was used to derive three binary indicators:

* + plain\_color: equal to 1 if the item was a solid color.
  + pattern: equal to 1 if the item featured any kind of pattern.
  + print: equal to 1 if the item had printed designs.

These distinctions were made using regular expressions to parse the text content from mainCatCode column and assign the appropriate flags.

**Category-Based Features**: The mainCatCode column was parsed using regular expressions to extract multiple layers of categorical information:

* 1. **Premium Indicator**: A binary premium flag was created, with value 1 for items belonging to premium product lines.
  2. **Gender Tags**: Three binary gender flags—men, ladies, and neutral—were generated to classify the target audience.
  3. **Item Category Flags**: Additional binary features were created for specific product categories such as shoes, bottoms, jeans, tops, socks, outerwear, accessories, underwear, dresses, sportswear, and loungewear, which were further divided into even smaller categories. These categories were inferred from textual patterns.

**Material Tier Features**: The materials column, containing composition text, was parsed to extract dominant materials using regex, filtering only those with ≥50% presence. Each item’s key materials were then matched against three predefined quality tiers:

* **High-tier** (e.g., leather, silk, cashmere)
* **Mid-tier** (e.g., cotton, linen, modal)
* **Low-tier** (e.g., polyester, acrylic, plastic)

Binary flags (high\_tier, mid\_tier, low\_tier) were assigned based on material presence, allowing multiple tiers per item. For instance, a product made of 70% of leather, 60% of cotton and 10% of polyester would have high\_tier = 1, mid\_tier = 1, low\_tier = 0 for the threshold set at 0.5. This structured encoding captures the value impact of materials for price prediction.

**Textual Embedding**: To utilize unstructured textual information, the details column underwent TF-IDF (Term Frequency–Inverse Document Frequency) vectorization. This technique transforms textual content into numerical features based on the relevance and uniqueness of words, improving the model's ability to capture semantic cues from product descriptions.

**Feature Utility**

All engineered features were developed with the goal of enhancing the model’s ability to accurately predict item price. By isolating key product attributes—such as category, composition, visual design, and description—this enriched feature space provides meaningful signals that support more robust learning and generalization.

Used methods

## Models

6 distinct models were initialized and their hyperparameters were tuned:

**Support vector regression (SVR)** – in SVR, dependent variables are continuous and the goal is to find the hyperplane which can predict the distribution of observations. Kernel function (e.g., radial basis function) allows to find non-linear relationships [5]. Grid search contained tuning error/margin trade-off parameter C and the insensitivity parameter epsilon and choosing the optimal kernel function.

**Multilayered Perceptron (MLP)** – is a neural network that consists of fully connected neurons (perceptrons) with one or more hidden layers which operates using feedforward and backpropagation mechanism [6] [7]. MLPs excel at high dimentional regression with non-linear relationship between variables [4]. Grid search provided optimal hyperparameters which included activation function, hidden layer size, initial learning rate, optimizer, regularization strength alpha, maximum number of epochs.

**Decision Tree (DT)** is a supervised learning model used both for regression and classification. It consists of a root node containing the entire sample, which then gets divided further into interior nodes. The last nodes, called leaf nodes, represent the outcome of the regression. The splitting criterion is derived from Gini index [8]. Grid search provided optimal max\_depth (maximum depth of the tree), min\_samples\_split hyperparameters (the minimum number of samples required to split an internal node), max\_features (the number of features to consider when looking for the best split).

**Random Forest** is an ensemble machine learning method that can be used both for regression and classification. Random Forest combines results of multiple decision trees which are constructed using bootstrap aggregation. For regression problems the result of Random Forest is equal to average value of the decesion tree predictions [9]. Hyperparameters include n\_estimators, which is the number of decision trees used in Random Forest.

**k-Nearest Neighbor (kNN)** is a self-learning non-parametric regression model. It is based on measuring distance between the input point and the training sample points and selecting predefined *k* nearest sample points. Average is then calculated as the mean of the target values of the *k*neighbors. kNN can discover non-linear relationships, but is sensitive to outliers [10]. Hyper-parameters tuned include weights, algorithm and *k* number of nearest neighbors.

**XGBoosting** is an ensemble method based on creating multiple decision trees in additive model, where each decision tree improves the result of the previous one. The goal is to minimize the gradient which consists of the regularized loss function. Unlike classical gradient boosting has non-zero regularization parameter [11]. Grid search provided fine-tuned hyperparameters which are the learning rate, number of trees, subsample (the fraction of samples to be used for fitting the individual base learners) and max\_depth (maximum depth of the individual regression estimators).

Dataset was split into train set (80%) and holdout test set (20%). Train test was used for models hyperparameters tuning and training. Holdout test was reserved for the final evaluation of the trained models.

## Hyperparameter tuning

For hyperparameter optimization, a 5-fold cross-validation (CV) was employed on the training set. This method splits the training data into subsets, ensuring that each fold was used both for training and validation across iterations. Grid search was applied to test combinations of prediction model hyperparameters, identifying the optimal configuration for each model. The metrics used during grid-search were negative mean squared error (NMSE) for kNN and XGBoost and R2 score for Decision Tree, Random Forest, MLP and SVR. Optimal hyperparameters were saved in json format and used later for initialization of the models.

## Validation

Training data was fit using optimal hyperparameters found using the grid-search. Evaluation was conducted on the holdout test split. To quantify regression accuracy, the following metrics were employed:

* **Root Mean Squared Error (RMSE)**: Measure of deviation between observed variable and predicted ones in original units.
* **Mean Absolute Error (MAE)**: Measures the average error magnitude in original units. Unlike RMSE is not influenced by large errors [1].
* **Coefficient of Determination (R²)**: Indicates how well the model explains the variance of the target variable. R2 score of 1 indicates that model account all the variability in target variable, while score of 0 indicates that no variance is explained [2] [3].

The individuals metrics barcharts are shown in the figures \[1] , \[2], \[3].

Citations

[1] Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research*, *30*(1), 79-82.

[2] Hughes, A. J., & Grawoig, D. E. (1971). Statistics, a foundation for analysis. *(No Title)*.

[3] Remegio, F. M. C. (2024, March). Predicting Student Performance to Boost Educational Outcomes: The Efficacy of a Random Forest Approach. In *2024 13th International Conference on Educational and Information Technology (ICEIT)* (pp. 253-260). IEEE.

[4] Hernández, N., Talavera, I., Dago, A., Biscay, R. J., Ferreira, M. M. C., & Porro, D. (2008). Relevance vector machines for multivariate calibration purposes. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *22*(11‐12), 686-694.

[5] Shen, X., Gong, X., Cai, Y., Guo, Y., Tu, J., Li, H., ... & Zhu, Z. J. (2016). Normalization and integration of large-scale metabolomics data using support vector regression. *Metabolomics*, *12*, 1-12.

[6] Sagiroglu, S., & Yildiz, C. (2002). A multilayered perceptron neural network for a micro-coplanar strip line. *Electromagnetics*, *22*(7), 553-563.

[7] Wright, L. G., Onodera, T., Stein, M. M., Wang, T., Schachter, D. T., Hu, Z., & McMahon, P. L. (2022). Deep physical neural networks trained with backpropagation. *Nature*, *601*(7894), 549-555.

[8] Kassim, N. M., Santhiran, S., Alkahtani, A. A., Islam, M. A., Tiong, S. K., Mohd Yusof, M. Y., & Amin, N. (2023). An Adaptive Decision Tree Regression Modeling for the Output Power of Large-Scale Solar (LSS) Farm Forecasting. *Sustainability*, *15*(18), 13521.

[9] Zhang, J., Fan, H., He, D., & Chen, J. (2019). Integrating precipitation zoning with random forest regression for the spatial downscaling of satellite‐based precipitation: A case study of the Lancang–Mekong River basin. *International Journal of Climatology*, *39*(10), 3947-3961.

[10] Cai, L., Yu, Y., Zhang, S., Song, Y., Xiong, Z., & Zhou, T. (2020). A sample-rebalanced outlier-rejected $ k $-nearest neighbor regression model for short-term traffic flow forecasting. *IEEE access*, *8*, 22686-22696.

[11] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).