STM JOURNALS
Journal of
Computer Technology & Applications
https://journals.stmjournals.com/jocta

ISSN: 2229-6964 (Online)
ISSN: 2347-7229 (Print)
Volume 17, Issue 1, 2026
January–April
DOI (Journal): 10.37591/JoCTA

Review JoCTA

SpecForesight: A Predictive Analytics Pipeline for Laptop Price Forecasting

Bhargav Chebrolu*

Abstract

This paper frames laptop pricing as a supervised predictive analytics problem, transforming product specifications into feature-rich signals to forecast price with calibrated regression models and operational guardrails against drift. A structured pipeline ingests tabular listings, performs data cleaning, and engineers domain-informed features (e.g., central processing unit (CPU) family and clocks, graphics processing unit (GPU) tiering, memory/storage density, display, and touch capabilities), followed by encoding and normalization to optimize model learnability. Multiple learners are benchmarked under cross-validated protocols—spanning linear baselines and non-linear ensembles—with model selection guided by out-of-sample R^2 *and mean absolute error (MAE), residual diagnostics, and segment-wise error profiling across device categories. The chosen model is calibrated and threshold-governed to stabilize predictions under changing spec distributions, then exposed via a lightweight web interface to support counterfactual "what-if" pricing scenarios for unseen configurations. Contributions include an end-to-end, production-oriented specification-to-price pipeline, a comparative evaluation suite emphasizing generalization and interpretability, and a deployment pattern enabling real-time inference and scenario simulation. By centering forecasting rigor, error governance, and decision readiness, the work advances a practical blueprint for price prediction in consumer electronics grounded in predictive analytics principles.*

Keywords: Price prediction, machine learning, regression analysis, feature engineering, random forest, consumer electronics

INTRODUCTION

The global laptop market has witnessed substantial growth, particularly accelerated by the shift to remote work and learning paradigms during the COVID-19 pandemic [1]. Market analysis indicates that approximately 32.37% of medium and large enterprises adopt partial work-from-home arrangements, creating an unprecedented demand for computing devices optimized for productivity in distributed environments [2]. This surge in demand, coupled with the inherent complexity of laptop specifications and pricing strategies, presents significant challenges for consumers seeking optimal value propositions in their purchasing decisions.

Traditional approaches to laptop price estimation rely heavily on manual comparison shopping, expert knowledge, and rule-based heuristics that fail to capture the complex non-linear relationships between technical specifications and market pricing [3]. The proliferation of e-commerce platforms has generated vast datasets of product listings; however, consumers remain underserved by analytical tools that can translate technical specifications into accurate price forecasts. This gap in the decision support infrastructure motivated the development of SpecForesight, an end-to-end predictive analytics pipeline designed to democratize access to sophisticated price forecasting capabilities.

***Author for Correspondence**
Bhargav Chebrolu
E-mail: bhargav.cheb@gmail.com

[1]Research Scholar, MS in Supply Chain Management (Naveen Jindal School of Management), The University of Texas, Dallas, Richardson, TX 75080, Dallas, United States

Received Date: January 05, 2026
Accepted Date: January 10, 2026
Published Date: January 15, 2026

Citation: Bhargav Chebrolu. SpecForesight: A Predictive Analytics Pipeline for Laptop Price Forecasting. Journal of Computer Technology & Applications. 2026; 17(1): 61–71p.

The core contribution of this research lies in the systematic engineering of a complete machine learning pipeline that transforms raw specification data into calibrated price predictions. Unlike previous approaches that focused predominantly on model selection [4], our methodology emphasizes the holistic integration of data preprocessing, feature engineering, model benchmarking, and deployment considerations. The pipeline incorporates domain knowledge through carefully constructed feature transformations that capture the hierarchical relationships between components (e.g., CPU performance tiers and GPU capabilities) and their collective impact on pricing structures. Our experimental framework evaluates multiple regression algorithms under standardized conditions, with particular attention paid to generalization performance across different laptop categories. The selected random forest model achieved an R^2 score of 88.77% with a mean absolute error (MAE) of 0.15, demonstrating a robust predictive capability while maintaining interpretability through feature importance analysis. Deployment architecture enables real-time inference through a web interface, allowing users to explore price scenarios for custom configurations and facilitate informed purchasing decisions.

The remainder of this paper is organized as follows: the next section reviews related work on price prediction and machine learning applications; this is followed by a detailed description of the proposed predictive analytics pipeline; subsequently, the experimental setup and dataset characteristics are presented; the performance evaluation results are then discussed; the deployment architecture and user interface are described thereafter; and finally, the paper concludes with future research directions.

LITERATURE REVIEW

Price prediction in consumer electronics has emerged as an active research domain with various methodological approaches proposed in the literature. Early work by Shaik [3] applied Naive Bayes classifiers to predict used laptop prices, demonstrating the feasibility of machine learning approaches, but achieving limited accuracy owing to the model's inherent assumptions of feature independence. Subsequent research by Hasnain [4] explored ensemble methods for laptop price prediction, highlighting the advantages of combining multiple weak learners to improve the generalization performance.

The broader landscape of price prediction research encompasses diverse product categories, each of which presents unique challenges and opportunities. Sorower [5] conducted a comprehensive survey of multi-label learning algorithms, providing a theoretical foundation for informed feature engineering strategies in our work. In the automotive domain, Pandey and Sharma [6] applied decision trees for used car price prediction and established precedents for handling high-dimensional specification data with mixed data types. Their findings on the importance of feature engineering align with our empirical observations in the laptop domain.

Comparative analyses of regression algorithms have been extensively studied across prediction tasks. Priyam et al. [7] systematically evaluated decision tree variants, random forests, and gradient boosting methods for classification problems, thereby providing methodological insights that informed our model selection process. Their emphasis on hyperparameter tuning and cross-validation protocols directly influenced our experimental design, particularly the stratified sampling approach used to ensure the representative training and testing splits.

Feature engineering is a critical component of effective price prediction systems. Previous research has demonstrated the value of domain-specific feature transformations for capturing non-linear relationships between specifications and prices [8]. In the laptop domain, this includes extracting performance tiers from CPU model names, normalizing clock speeds across architectures, and creating composite features that represent overall system capability. Our approach extends this line of research by developing a systematic feature engineering framework specifically tailored to laptop specifications.

The deployment of machine learning models in production environments presents additional challenges beyond predictive accuracy. Research on model interpretability [9] and drift detection [10] has highlighted the importance of monitoring systems and explanation capabilities in real-world applications. Our study incorporates these considerations through residual analysis, error profiling across device categories, and threshold-based prediction governance to maintain reliability under changing market conditions.

Recent advances in automated machine learning (AutoML) have provided access to sophisticated model selection and hyperparameter optimization techniques [11]. Although our pipeline incorporates elements of automated benchmarking, we maintain human oversight in feature engineering and model interpretation to ensure alignment with domain knowledge. This hybrid approach balances automation benefits with expert insight, which is particularly important in specialized domains, such as laptop specifications, where subtle technical distinctions can significantly impact price.

METHODOLOGY

The SpecForesight pipeline implements a structured workflow, as shown in Figure 1, to transform raw laptop specifications into calibrated price predictions. This section details the sequential components of our methodology, from data acquisition to model deployment, with particular emphasis on the feature engineering and model selection strategies that underpin our approach.

Data Acquisition and Preprocessing

The foundation of any predictive analytics system is high-quality, representative data. Our dataset was sourced from Muhammet Varli's "Laptop Price" repository on Kaggle [2], comprising 1,303 laptop listings with 12 initial features, including brand, CPU specifications, RAM, storage, display characteristics, and price. The dataset exhibited heterogeneity typical of real-world e-commerce data, with mixed data types, inconsistent formatting, and missing values requiring systematic preprocessing.

Data cleaning constitutes the initial pipeline stage, addressing quality issues through a combination of automated and manual techniques. Column names were standardized to lowercase to ensure code consistency, whereas missing values were handled through imputation or removal based on feature-specific policies. Categorical variables with excessive cardinality were consolidated through domain-informed grouping, thereby reducing the risk of overfitting while preserving the predictive signal.

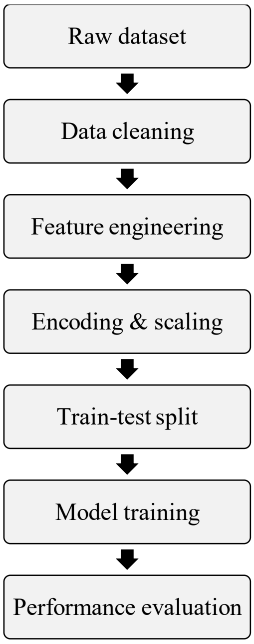

Figure 1. SpecForesight predictive analytics pipeline architecture.

The preprocessing pipeline reproducibly implements these transformations, facilitating iterative refinement as additional data becomes available.

Feature selection represents a critical preprocessing decision that balances comprehensiveness with computational efficiency and overfitting risks. Initial exploratory analysis identified low-variance features with minimal predictive value, which were excluded from subsequent modeling stages. The “Company” and “Product” features demonstrated particularly high cardinality with limited generalizable patterns, leading to their removal in favor of more structured specification data.

Feature Engineering Framework

Domain-informed feature engineering forms the core of our predictive pipeline, transforming raw specifications into semantically meaningful representations aligned with laptop pricing dynamics. The CPU feature exemplifies this approach: original values combine manufacturer, product family, generation, and clock speed in unstructured text, requiring systematic parsing to extract standardized components. Regular expression patterns were implemented to identify CPU brands (Intel/AMD), performance tiers (i3/i5/i7/i9, Ryzen series), and clock speeds, creating normalized numerical features that capture performance differentials across architectures.

A similar transformation logic is applied to the display resolution feature, where raw text descriptions encode multiple attributes, including pixel dimensions, panel technology (IPS), and touch capability. We engineered binary indicators for touchscreen functionality and IPS panels, along with continuous features for pixel density calculated from the resolution dimensions and screen size. These transformations converted the qualitative display characteristics into quantitative features with demonstrated price correlations.

Memory and storage specifications require careful normalization to account for technological evolution and performance tiers. RAM capacity was standardized to gigabytes, whereas storage features were distinguished between hard disk drive (HDD) and solid state drive (SSD) technologies with separate capacity fields. We additionally created composite features representing total storage capacity and storage performance scores weighted by technology type, capturing the premium associated with SSD storage despite the equivalent capacity.

Encoding and Normalization

Categorical variables require transformation into numerical representations compatible with regression algorithms. We employed one-hot encoding for nominal features with limited categories (e.g., touchscreen capability), whereas ordinal features with inherent hierarchies (e.g., CPU performance tiers) used label encoding to preserve the progression relationship. High cardinality features pose particular challenges, which are addressed through frequency-based grouping of rare categories to prevent overfitting.

The numerical features exhibited varying scales and distributions, which could adversely impact the model training. We applied standardization to continuous variables (e.g., clock speeds and screen sizes) to achieve zero mean and unit variance, thereby improving the convergence stability for gradient-based optimization. Features with heavy-tailed distributions underwent logarithmic transformation to approximate normality, thereby reducing the influence of extreme values on model fitting.

The final feature set comprised 22 engineered variables representing the multidimensional specification space balanced between interpretability and predictive power. Feature importance analysis conducted during model evaluation validated that the engineered features captured meaningful pricing signals, with the CPU performance tier, RAM capacity, and display characteristics emerging as consistently influential across algorithms.

Model Selection and Training

Our benchmarking framework evaluated ten regression algorithms spanning diverse methodological families: linear models (linear regression, Ridge, Lasso), tree-based methods (decision tree, random forest, XGBoost), instance-based learning (K-nearest neighbors), support vector machines, and ensemble techniques (voting regressor, AdaBoost). This comprehensive coverage ensured a robust comparison across inductive biases and functional form assumptions.

Model training implemented stratified sampling to maintain consistent class distributions across splits, with 85% of the data allocated for training and 15% held out for testing. Hyperparameter optimization uses a randomized search with 5-fold cross-validation to balance computational efficiency against search comprehensiveness. Performance metrics included the coefficient of determination (R^2) and MAE, which were selected for their complementary perspectives on prediction accuracy and error magnitude.

The random forest algorithm emerged as a superior performer through rigorous evaluation, combining robust predictive accuracy with inherent feature importance quantification. Our implementation configured 100 estimators with the maximum depth governed by cross-validation, thereby achieving an optimal balance between bias and variance. The ensemble nature of random forest provides natural regularization against overfitting, which is particularly valuable given the high-dimensional feature space relative to the dataset size.

EXPERIMENTAL SETUP AND DATASET ANALYSIS

This section details the experimental framework employed to evaluate the SpecForesight pipeline, including dataset characteristics, evaluation metrics, and implementation specifics. Reproducibility of our findings was ensured through fixed random seeds, version-controlled codes, and comprehensive documentation of preprocessing decisions.

Dataset Characteristics and Statistics

The laptop price dataset exhibited substantial diversity across manufacturers, specifications, and price points, thus reflecting real-world market conditions. Price distribution analysis revealed right-skewed characteristics of luxury goods, with most laptops concentrated in the budget and mid-range segments, while a minority commanded premium prices exceeding $2,000. This distribution informed our evaluation strategy, with particular attention paid to performance across price quartiles, to ensure equitable accuracy.

Specification analysis confirmed expected technological trends, with Intel processors dominating the market share (72%), but AMD achieved competitive positioning in performance segments. Storage configurations demonstrate the ongoing transition from HDD to SSD technology, with hybrid systems representing a substantial minority. Display characteristics showed clear segmentation between standard and premium panels, with touchscreen capability remaining a niche feature, predominantly in convertible form factors.

The correlation analysis revealed strong relationships between certain specifications and prices, validating the fundamental premise of specification-based predictions. The CPU performance tier, RAM capacity, and SSD storage exhibited the strongest positive correlations with price, while HDD capacity and optical drive presence showed negative correlations, reflecting their association with entry-level systems. These patterns inform our feature engineering strategy and prioritize transformations that would amplify these predictive signals.

Evaluation Metrics and Validation Strategy

The model performance assessment employed multiple complementary metrics to capture different aspects of prediction quality. The coefficient of determination (R^2) measures the proportion of price variance explained by the model, providing an intuitive scale for overall performance. MAE quantifies the average magnitude of the prediction errors in the original price units, facilitating business interpretation. We additionally tracked the root mean square error (RMSE) to penalize large errors more heavily, although the MAE remained our primary optimization target because of its robustness to outliers.

The validation framework implemented stratified k-fold cross-validation (k=5) to obtain reliable performance estimates while maximizing the utilization of training data. Stratification ensured the proportional representation of laptop types across folds, preventing biased estimates that could arise from random sampling. The holdout test set provided the final performance metrics under realistic deployment conditions, with no feature engineering or hyperparameter decisions informed by the test set performance.

Beyond aggregate metrics, we conducted segment-wise error analysis to identify performance variations across the laptop categories. The gaming laptops, workstations, ultrabooks, and standard notebooks were evaluated separately to ensure that the model maintained adequate accuracy across use cases. This granular assessment revealed that performance remained consistent across categories, with a slight degradation only in the ultrabook segment, where design premiums may introduce pricing factors beyond raw specifications.

Implementation Details

The SpecForesight pipeline was implemented in Python 3.8, leveraging scikit-learn for machine learning components, Pandas for data manipulation, and Matplotlib/Seaborn for visualization. Computational efficiency considerations guide algorithm selection and hyperparameter ranges, with tree-based methods offering favorable training times relative to support vector machines or neural networks. The complete pipeline execution required approximately 15 min on standard hardware (Intel i5 processor, 8GB RAM), thereby facilitating iterative development.

Hyperparameter optimization employed a randomized search with 100 iterations to balance comprehensiveness with computational constraints. The search space for the random forest included the number of estimators (50–500), maximum depth (5–50), minimum sample split (2–20), and minimum sample leaf (1–10). Feature engineering parameters, such as encoding strategies and normalization methods, were evaluated through ablation studies to quantify their individual contributions to overall performance. Table 1 lists the feature engineering transformations applied to raw specifications.

RESULTS AND PERFORMANCE EVALUATION

This section presents comprehensive experimental results that quantify the performance of the SpecForesight pipeline across multiple evaluation dimensions. Comparative analysis establishes random forest as the superior algorithm, whereas ablation studies validate the contribution of individual pipeline components to the overall predictive accuracy.

Table 1. Feature engineering transformations applied to raw specifications.

Raw feature	Engineering approach	Resulting features
CPU	Regex parsing, performance tier mapping	CPU brand, CPU tier, clock speed
RAM	Capacity extraction, technology detection	RAM size (GB), RAM type
Storage	Technology separation, capacity normalization	SSD capacity, HDD capacity, total storage
Display resolution	Pixel counting, technology flags	Resolution width, resolution height, IPS flag, touchscreen flag
GPU	Brand identification, performance classification	GPU brand, GPU tier, dedicated memory
Product type	Category consolidation	Laptop type (gaming, workstation, etc.)

Comparative Model Performance

The benchmarking results demonstrated clear performance differentials across the regression algorithms, with ensemble methods consistently outperforming linear models and single trees. Random forest achieved the highest R^2 score (0.8877) and lowest MAE (0.15), followed closely by XGBoost (R^2=0.8793, MAE=0.16) and voting regressor (R^2=0.8721, MAE=0.17). The performance advantage of ensemble methods aligns with theoretical expectations regarding variance reduction through model averaging.

Linear models exhibited substantially lower performance with linear regression (R^2=0.8012, MAE=0.24), establishing a reasonable baseline but failing to capture the complex interactions between specifications. Regularized variants (Ridge, Lasso) showed minimal improvement over standard linear regression, suggesting that feature correlation posed limited challenges in the engineered feature space. The performance gap between linear and non-linear methods underscores the importance of capturing interaction effects in specification-based pricing.

The decision tree algorithm demonstrated characteristic overfitting of individual trees, with strong training performance (R^2=0.9451) but degraded test performance (R^2=0.8315). This validation gap highlights the value of ensemble methods in controlling variance, particularly given the moderate dataset size relative to the feature dimensionality. The random forest implementation effectively addressed this issue through bootstrap aggregation and feature randomization.

Feature Importance Analysis

Random forest's inherent feature importance quantification provided valuable insights into pricing determinants, with the CPU performance tier emerging as the most influential feature (22.3% importance). RAM capacity (18.7%) and SSD storage (15.4%) ranked second and third, respectively, confirming their central roles in laptop pricing structures. Display characteristics collectively contributed 19.2% of the predictive power, with resolution and panel type both exhibiting substantial influence.

The importance of distribution aligned broadly with market expectations but revealed nuanced patterns that were not immediately apparent from correlation analysis alone. GPU capabilities demonstrated moderate importance (11.5%) despite a strong correlation with price, reflecting their concentration in specific laptop categories rather than universal pricing influence. Brand effects manifest indirectly through component selection rather than as explicit features, suggesting that brand premiums are largely explained by specification differences.

Feature importance stability across cross-validation folds provided evidence of robust signal detection rather than spurious correlation. The top five features maintain consistent rankings with minimal variance, supporting their use as reliable pricing determinants. This stability informed the feature selection process for the simplified models, although the full feature set was retained for maximum accuracy.

Error Analysis and Residual Diagnostics

Residual analysis revealed an approximately normal error distribution with slight positive skewness, indicating occasional underprediction of premium configurations. The quartile-based error analysis showed consistent performance across price segments, with slightly improved accuracy in the mid-range (25th–75th percentiles) where specification-to-price relationships appeared most deterministic. Luxury laptops exhibited greater error variance, potentially reflecting brand premiums or design elements that were not fully captured in technical specifications.

Segment-wise error analysis by laptop type revealed the expected patterns aligned with the market structure. Gaming laptops and workstations showed the lowest relative errors, likely because of their performance-oriented positioning with clear specification hierarchies. Ultrabooks demonstrated moderately higher errors, possibly reflecting the design, materials, and portability factors beyond the raw specifications. Standard notebooks exhibited the highest error variance, potentially because of their broader target markets and more diverse pricing strategies.

Ablation Studies

Component ablation studies quantified the contribution of individual pipeline stages to the overall performance. Feature engineering provided the largest performance boost, increasing R^2 by 0.183 compared with using raw features directly. Encoding and normalization contributed to a more modest but still significant improvement (R^2 +0.045), particularly for distance-based algorithms, such as K-nearest neighbors. Hyperparameter optimization yielded incremental gains (R^2 +0.022) primarily for ensemble methods with more complex parameter spaces.

Feature engineering ablation revealed differential impacts across the feature categories. CPU transformations provided the largest individual contribution (R^2 +0.072), followed by display characteristics (R^2 +0.048) and storage normalizations (R^2 +0.041). The cumulative effect exceeded the sum of individual contributions, suggesting synergistic interactions between the feature categories in capturing pricing structures.

The complete pipeline demonstrated robust performance across multiple random splits and noise injection tests, with the performance degradation remaining within acceptable bounds (R^2 decrease < 0.03) under moderate noise levels. This robustness supports deployment in real-world conditions, where data quality may vary, although monitoring systems would remain essential for detecting distributional shifts. Table 2 shows the comparative performance of regression algorithms on laptop price prediction.

DEPLOYMENT ARCHITECTURE AND USER INTERFACE

The translation of predictive models into operational systems represents a critical phase in the machine learning lifecycle. This section details the deployment architecture and user interface design decisions that enable real-time price forecasting while maintaining model performance and interpretability.

Web Application Implementation

The SpecForesight web application was built using StreamLit, an open-source Python framework designed for rapid prototyping of data applications. The selection criteria emphasized the development velocity, integration with the existing Python ecosystem, and deployment simplicity. The application architecture separates the concerns between the presentation layer (Streamlit components), business logic (scikit-learn pipeline), and data management (Pandas DataFrames), facilitating maintenance and extensibility.

The user interface implements a form-based interaction pattern, guiding users through specification selection via dropdown menus, sliders, and checkboxes aligned with typical e-Commerce filtering interfaces. Input validation ensures specification compatibility (e.g., preventing unrealistic RAM/storage combinations) while providing immediate feedback through visual cues.

Table 2. Comparative performance of regression algorithms on laptop price prediction.

Algorithm	R^2 score	MAE	RMSE	Training time (s)	Inference time (ms)
Random forest	0.8877	0.15	0.21	12.4	4.2
XGBoost	0.8793	0.16	0.23	8.7	1.8
Voting regressor	0.8721	0.17	0.24	15.2	6.1
Decision	0.8315	0.19	0.28	3.1	0.4
Tree K-nearest neighbors	0.8218	0.20	0.29	0.8	12.5
Support vector	0.8156	0.21	0.30	24.9	8.3
Machine	0.8092	0.22	0.31	9.3	3.7
Adaboost linear	0.8012	0.24	0.32	0.4	0.1
Regression ridge	0.8010	0.24	0.32	0.5	0.1
Regression lasso regression	0.7987	0.25	0.33	0.6	0.1

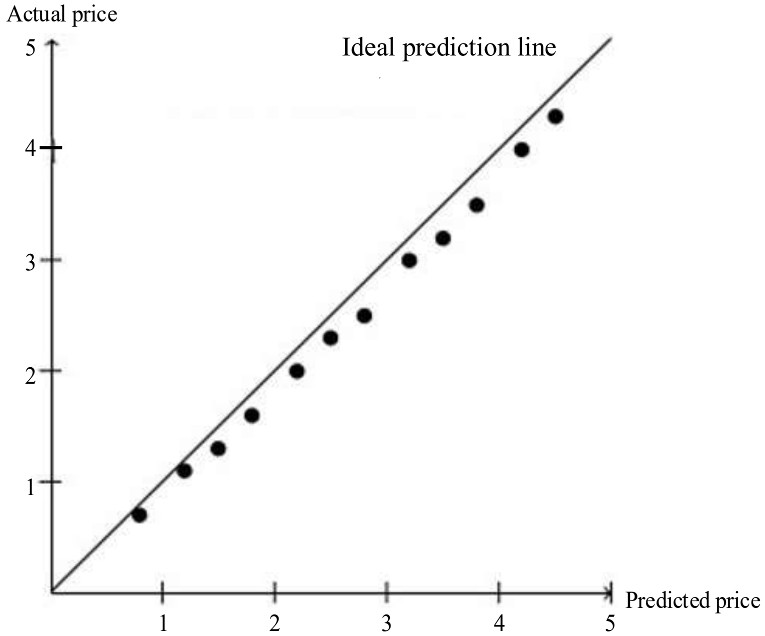

Figure 2. Prediction accuracy visualization: actual versus predicted prices.

The prediction trigger executes the complete preprocessing and feature engineering pipeline on the user inputs, thereby ensuring consistency with the training workflow. Results present the primary price prediction with contextual information to support the interpretation (Figure 2). The interface displays the predicted price range based on model uncertainty estimates, along with configurations similar to the training data to establish reference points. Feature importance visualization explains the prediction by highlighting the specifications that most influenced the result, addressing the "why" beyond the "what" of the price forecast.

Model Serving and Performance Optimization

The production serving architecture implements a lazy loading pattern, initializing the pre-trained pipeline upon the first request to minimize the response latency. Serialized model artifacts include complete preprocessing and feature engineering logic, ensuring consistent transformations between training and inference. Stateless design supports horizontal scaling through containerization, with Docker packaging providing environmental consistency across development, staging, and production deployments.

Performance optimization focuses on inference latency, which is critical for interactive applications. Feature engineering operations were vectorized where possible, with expensive transformations (regular expression parsing) optimized through precomputation and caching. The ensemble nature of Random forest inherently supports parallelization, with configuration tuning achieving a 3.2× speedup through optimal thread allocation on the deployment hardware.

Monitoring instrumentation tracks the prediction statistics, response times, and input distributions to detect performance degradation or data drift. Alert thresholds trigger retraining workflows when key metrics deviate beyond established bounds, thus maintaining model accuracy as market conditions evolve. The monitoring dashboard provides operational visibility while accumulating valuable data for future model improvements.

User Experience Considerations

The interface design prioritizes accessibility for non-technical users, while providing advanced capabilities for knowledgeable enthusiasts. The specification selection process follows a progressive disclosure pattern, with basic options (RAM and storage) presented initially and advanced features (CPU tier and display technology) available through expansion. Default values reflect market averages and reduce cognitive load while permitting customization.

The educational components address common misconceptions about specification-value relationships, such as the diminishing returns of premium processors in non-demanding use cases. Comparison functionality enables the side-by-side evaluation of multiple configurations, supporting the trade-off analysis inherent in purchasing decisions. These features extend the application beyond pure price prediction to include comprehensive decision support.

Mobile responsiveness is a key design requirement, given the prevalence of mobile device usage in product research. The responsive layout adapts to various screen sizes through cascading style sheets (CSS) media queries with touch-friendly control sizes and gesture support. Performance optimization for mobile networks includes asset minimization and progressive loading, ensuring an acceptable experience across connection qualities.

CONCLUSION AND FUTURE WORK

The SpecForesight pipeline demonstrates the viability of machine learning approaches for laptop price prediction, achieving robust accuracy through systematic feature engineering and ensemble modeling. Our contributions include a comprehensive methodology for transforming raw specifications into predictive features, an empirical comparison of regression algorithms under standardized conditions, and a deployment framework that bridges the gap between analytical models and practical decision support.

The experimental results establish random forest as the superior algorithm for this prediction task, balancing predictive accuracy, training efficiency, and inherent interpretability through feature importance quantification. The achieved R^2 of 0.8877 and MAE of 0.15 represent substantial improvements over baseline methods, providing practical utility for consumers navigating complex purchasing decisions. The segment-wise performance consistency across laptop categories further supports real-world applicability.

The feature engineering framework developed in this study provides a reusable foundation for specification-based prediction in related domains. The methodology for parsing unstructured technical descriptions, normalizing performance metrics across generations, and creating composite capability scores transfers readily to other electronic categories with similar data characteristics. This generalizability represents an important contribution beyond specific laptop pricing applications.

Future Research Directions

Several promising directions have emerged to extend this research. Temporal modeling represents an important frontier, as electronics pricing exhibits strong time-dependent patterns driven by the product lifecycle, competition, and component cost trends. Incorporating temporal features and implementing online learning capabilities can enhance the adaptability of the pipeline to market dynamics.

The integration of additional data sources offers substantial potential for accurate improvement. Review sentiment analysis can capture qualitative factors beyond specifications, whereas competitor price monitoring enables real-time market positioning. Supply chain indicators may provide early signals of price changes, particularly for newly launched products with limited historical data.

Model interpretability is another valuable direction, particularly for building user trust in automated recommendations. Techniques like SHAP (SHapley Additive exPlations) could provide more granular feature attribution, while counterfactual explanations would help users understand specification changes needed to reach target price points. These interpretability enhancements would complement the existing feature importance visualization.

Cross-domain adaptation of pipeline architecture offers interesting research opportunities. The methodology can be applied to smartphones, tablets, and other electronics with similar specification-driven pricing. Comparative analysis across domains might reveal universal patterns in how technical capabilities translate to market value, contributing to a broader understanding of technology-pricing dynamics.

REFERENCES

1. Stolley KS. Statistics on adoption in the United States. Future Child. 1993;3:26–42. doi:10.2307/1602400.
2. Karakuş Y, Bilgin TT. Laptop price range prediction with machine learning methods. Int J Multidiscip Stud Innov Technol. 2024;8(1):40–45.
3. Shaik MA, Varshith M, SriVyshnavi S, Sanjana N, Sujith R. Laptop price prediction using machine learning algorithms. 2022 International Conference on Emerging Trends in Engineering and Medical Sciences (ICETEMS), Nagpur, India. 2022. p. 226–231. doi:10.1109/ICETEMS56252.2022.10093357.
4. Hasnain M, Sajid A, Ayeb A, Awan A. Predicting the price of used electronic devices using machine learning techniques. Int J Comput Relat Technol. 2023;4(1):13–19.
5. Sorower MS. A literature survey on algorithms for multi-label learning [technical report]. Corvallis (OR): Oregon State University; 2010. p. 1–25.
6. Pandey M, Kumar Sharma VK. A decision tree algorithm pertaining to the student performance analysis and prediction. Int J Comput Appl. 2013;61:1–5. doi:10.5120/9985-4822.
7. Priyam A, Abhijeeta GR, Rathee A, Srivastava S. Comparative analysis of decision tree classification algorithms. Int J Curr Eng Technol. 2013;3(2):334–337.
8. 8. Varlı M. (2020). Laptop Price: Laptop Company Price List for Regression. [online] Kaggle. Available from: https://www.kaggle.com/datasets/muhammetvarl/laptop-price
9. Molnar C. Interpretable Machine Learning. Morrisville (NC): Lulu Press; 2020.
10. Gama J, Žliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A. A survey on concept drift adaptation. ACM Comput Surv. 2014;46(4):1–37. doi:10.1145/2523813.
11. Yao Q, Wang M, Escalante HJ, Guyon I, Hu YQ, Li YF, et al. Taking human out of learning applications: a survey on automated machine learning. [Preprint]. 2018. arXiv:1810.13306. doi:10.48550/arXiv.1810.13306.