Smartphone Price Prediction via Enhanced Interaction Features

Luoqi Li

{2451340@tongji.edu.cn}

School of Design and Creativity, Tongji University,1239, Siping Road, Shanghai 200092, China

Abstract. This study fixes the problems of old feature methods for smartphone price prediction. It uses a two-step feature engineering method. This method mixes market knowledge with statistical checks. It builds interactive features from market logic. Then, it picks five key features using significance tests and overall evaluation. Tests with four models including linear regression, support vector regression, random forest, and XGBoost show the better feature set improves predictions for all models. The R^2 score goes up by 59.9% for linear regression and 4.9% for random forest. More tests show this method can improve results by up to 32.2% more than old feature selection ways. This work gives a useful feature engineering method for smartphone price prediction. It shows why feature quality matters and gives a guide for building features in tough market prediction jobs.

Keywords: Feature engineering; Smartphone price prediction; Interactive feature construction; Machine learning

1 Introduction

The smartphone market is very competitive. So, predicting phone prices well is very useful for buyers and companies [1]. Using machine learning here has created a research plan focused on improving algorithms, engineering features, and evaluating models.

Studies show that comparing and improving supervised learning algorithms can make predictions better. Old classifiers like support vector machines and random forests work well after tuning. Also, ensemble learning methods can make predictions more stable [2]. Feature engineering, by using statistical tests to find key features or by reducing dimensions, is a main way to make models better [3].

For model design and tuning, recent studies have made good progress. Güven çet al. compared K-Nearest Neighbors (KNN) and deep neural networks for phone price prediction. They found deep neural networks are much better at handling complex feature links [4]. Chang et al. mixed support vector machines with Optuna and Hyperopt tuning tools. They made a good system for tuning models that greatly improved prediction performance [5]. Also, Saeed et al. looked at the harder problem of pricing used phones. They said people must think about things like device condition and market changes [6]. And, the evaluation system from Hamid et al.

ICIAAI 2026, May 22-24, Singapore, Singapore

DOI 10.4108/eai.22-5-2026.2365096

helped the buyer choice process [7]. These studies helped use deep learning for price prediction and gave new ideas for tuning models in complex cases.

But there are still big problems to solve. First, for making features, most studies just use basic hardware specs. They do not add market ideas like brand value or tech combinations. This limits how well people can explain the models in economic terms and use them in real life [8]. Second, for models, complex models like deep neural networks predict accurately but are hard to explain. This is a big problem for business uses where people need clear decisions [9]. Finally, for evaluation, different studies use different tests, data splits, and metrics. This makes it hard to compare studies and check progress. Also, researchers need more tests to see if models work in real markets [10].

To fix these problems, this study makes a smart pricing system. It mixes feature engineering using market knowledge with advanced model tuning. By building features from market logic and tuning models automatically, this system aims to make predictions both more accurate and easier to explain. Through careful tests, the new method gives a fresh way to predict smartphone prices.

2 Methodology

2.1 Feature Construction and Selection

Dataset Used. This study uses the public Kaggle dataset called "Real World Smartphone's Dataset." It has 21 basic features like brand, model, price, and key hardware specs. From this start, interactive features were made. Then, a two-step filter was used to pick the final features for modeling. The dataset has a CC0 license, so it is good for research.

Feature Construction Based on Domain Knowledge. This paper shows a feature engineering method using market knowledge. It turns basic specs into features that make sense for the market to better predict phone prices. Numbers like price and storage are changed into business-related groups to be stronger and clearer. Four kinds of interactive features are built. The first kind shows how brand and features work together. The second finds links between different technologies. The third shows what specs are liked in different price groups. The fourth judge's total device performance using ratios and scores. This plan creates a clear set of possible features to help with later feature selection and model training.

Feature Selection via Statistical and Quality Assessment. To find strong and reliable features from many options, this study makes a two-step feature selection plan. This plan first uses statistical significance tests, then an overall quality check. This slowly narrows down from statistically related features to good predictors.

Step one is statistical significance filtering. It removes features with no real statistical link to the target, price. For each possible feature and price, the test is picked based on feature type. Number features use the Pearson correlation test. The p-value is found with the related formula, shown in Equation (1).

$$p = P(|r| \geq r_{obs} \mid H_0: \rho = 0) \tag{1}$$

For category or yes/no features, one-way analysis of variance (ANOVA) is used. The p-value is found with the related measure, shown in Equation (2).

$$p = P(F \geq F_{obs} \mid H_0: \mu_1 = \mu_2 = \cdots = \mu_k) \tag{2}$$

The significance level is $\alpha = 0.05$. Only features with a p-value below α move to step two. This makes sure picked features have a clear statistical link to price.

Step two is an overall quality check. Features that passed step one get more checking based on two things, which are data quality and information amount. Specifically, a total quality score Q is made to measure each feature's strengths and weaknesses. This score comes from three parts combined with weights. The first part is coverage rate C. It measures how complete the feature data is. It is the share of good data, meaning the ratio of usable data points to all samples N. The second part is uniformity U. It checks how balanced and spread out the feature values are. This is usually found from information entropy or the Gini coefficient. It shows how spread the values are and avoids problems if values are too similar or skewed. The third part is mutual information score MI. It measures the nonlinear statistical link between the feature and price. This can find complex links beyond simple correlation. It adds to the linear testing in step one, as shown in Equation (3).

$$MI = I(X_i; y) \tag{3}$$

Finally, the total quality score Q is found by a weighted sum, as in Equation (4).

$$Q = 0.3 \times C + 0.4 \times \frac{U}{100} + 0.3 \times MI \times 10 \tag{4}$$

In this step, the limits are $C \geq 0.5$ and $U \geq 3$. All features that pass are sorted by Q score from high to low. The top 5 features are picked for the final set. This two-step method mixes statistical care with business sense. It removes useless features and makes sure the model is strong and works well in many ways, like data quality, stable spread, and prediction information.

2.2 Experiment Validation Design

Performance Across Different Models. To carefully check if the better interactive features work for different models, this study picks four machine learning models with different designs for comparison. Linear regression is simple and explainable. It clearly shows the overall straight-line link between features and the target, giving a baseline for more complex models. Support vector regression handles nonlinear patterns using kernel tricks. Its focus on reducing structural risk keeps performance steady with small samples or many features. Random forest is an ensemble method. It models complex feature interactions well by building many decision trees and combining their predictions. XGBoost uses a gradient boosting framework. It makes predictions better step-by-step by improving groups of decision trees. These four models cover the range from simple linear to complex nonlinear. This allows a full check of how well the feature engineering method works with different modeling ideas.

Each model is tested with two feature sets. The first set picks the five best original features using correlation. The second set has five interactive features from the two-step feature engineering. Model strength is checked with five-fold cross-validation. Three metrics, the coefficient of determination(R^2), root mean square error (RMSE), and mean absolute error (MAE), are calculated on a separate test set. The study looks at how each model performs with the different feature sets. The goal is to see if the better features improve all models.

Robustness Validation. To make sure the two-step feature engineering method is strong and can generalize, this study makes a careful robustness test plan. It compares the two-step method with three old feature selection ways, which are correlation analysis, F-test, and random forest importance ranking. This comparison aims to fully check how feature quality affects prediction performance. Correlation analysis picks feature most linked to price using the Pearson correlation coefficient. The F-test finds feature groups that help explain price differences using analysis of variance. Random forest importance ranking finds nonlinear links and feature interactions using the model's own workings. By comparing these four methods, this research wants to prove the two-step method works well and is better across different evaluation angles and technical ideas.

3 Results

3.1 Feature Selection Results from Two-Stage Engineering

Following the method in Section 2.1, this study builds interactive features from market knowledge starting from 21 basic features. This makes 54 possible features. Then, a two-step filter is used. First, an initial filter uses statistical significance with a p-value below 0.05. Next, a finer filter uses a quality score from several measures. Finally, the five features with the best prediction power are picked, as shown in Table 1.

Table 1. Top 10 Features Selected by the Two-Stage Feature Engineering Process

Feature Name	Feature Description	Feature Quality Score and Key Metrics
calc_price_per_gb	Price per GB storage ratio	10.779 (C: 1.00, U: 98.0, MI: 3.362)
price_5g_premium	5G capability premium	3.346 (C: 1.00, U: 490.0, MI: 0.362)
perf_high_speed	High performance combination	3.146 (C: 1.00, U: 490.0, MI: 0.296)
tech_high_ram_high_storage	High RAM + High Storage combination	3.088 (C: 1.00, U: 490.0, MI: 0.276)
price_large_screen_premium	Large screen size premium	3.083 (C: 1.00, U: 490.0, MI: 0.274)

Note: C = Coverage, U = Uniformity, MI = Mutual Information with target (price).

As Table 1 shows, the top feature, calc_price_per_gb (price per GB of storage), has a score of 10.779, much higher than others. This shows how important storage cost efficiency is for pricing. Also, premium and combination features like price_5g_premium and tech_high_ram_high_storage are included. This shows the market values specific functions and their combined effects. All picked features have full coverage, meaning the data is complete and modeling is strong. The final feature set went from 54 to just 5 features. This finds a good balance between prediction power, statistical strength, and business sense, giving a solid base for later modeling.

3.2 Model Performance

Table 2. Model Performance Across Engineered and Original Feature Sets

Model	BASE R^2	TOP5 R^2	BASE RMSE	TOP5 RMSE	BASE MAE	TOP5 MAE
Linear Regression	0.455	0.727 (+59.9%)	24185.95	17118.359 (-29.2%)	11871.25	8855.099 (-25.4%)
SVR	0.633	0.725 (+14.6%)	19843.17	17175.001 (-13.4%)	10107.24	8499.103 (-15.9%)
Random Forest	0.705	0.739 (+4.9%)	17789.19	16715.883 (-6.0%)	9691.79	8483.912 (-12.5%)
XGBoost	0.710	0.733 (+3.1%)	17625.36	16932.388 (-3.9%)	9658.32	8636.518 (-10.6%)

Following the method in Section 2.2, the test compares two feature sets across four models. The results are in Table 2.

Table 2 shows the performance of four models with two feature setups. Overall, the better feature set improves all models. Linear regression's R^2 goes up by 59.9%, and its RMSE and MAE go down by 29.2% and 25.4%. This means better features make the explainable straight-line link stronger and predictions more accurate. Support vector regression also improves R^2 by 14.6%, with lower errors. Random forest and XGBoost start with higher baselines but still improve R^2 by 4.9% and 3.1%. This shows interactive features also help nonlinear models.

Also, the drop in RMSE is bigger than the rise in R^2 for all models. This shows the better features are especially good at reducing large prediction mistakes. This overall improvement proves that interactive features built from business logic help all kinds of models. They give real gains in prediction power for machine learning models of any complexity.

3.3 Robustness Validation

To check the strength of feature selection methods, Figure 1 compares four different feature selection ways on the random forest model.

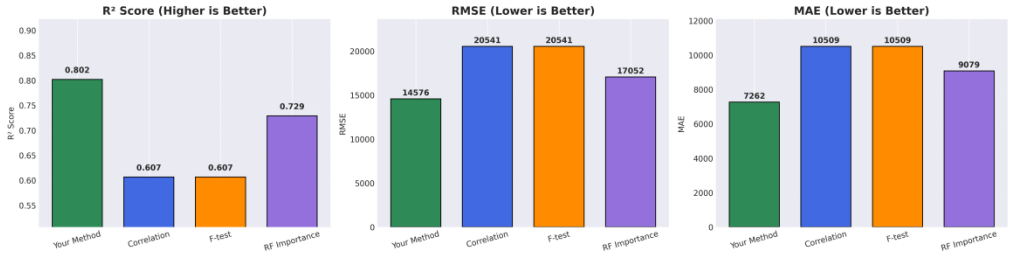

Fig. 1. Performance Comparison of Four Feature Selection Methods (left: Comparison of R^2, mid:Comparison of RMSE, right: Comparison of MAE) (Picture credit: Original)

The two-step method beats old statistical ways and standard importance ranking on all three metrics (R^2, RMSE, MAE). It gets an R^2 of 0.802. This is 9.9 percentage points higher than random forest importance and over 32 points higher than correlation and F-test ways. RMSE is 14.5% and 29.1% lower, and MAE is also the lowest seen. Correlation and F-test results are similar, showing a shared limit of old statistical feature selection. The random forest importance method works better than pure statistical tests but is still below the two-step strategy using market knowledge. This confirms that adding business logic gives improvements that pure statistical methods cannot. It also shows random forest is a good evaluation baseline, as it finds nonlinear interactions well while keeping feature representation stable.

4 Conclusion

This study makes and tests a two-step feature engineering method. It mixes market knowledge with statistical filtering. It builds interactive features from market logic and uses significance tests with checks across many measures to pick five strong core features from the options. The test uses four models including linear regression, support vector regression, random forest, and

XGBoost to show the method works for different models. It shows R^2 for linear regression and random forest improves by 59.9% and 4.9%. Strength tests show a steady advantage of 32.2%. The new idea here is mixing business logic with data-guided steps. It gives an efficient answer for smartphone price prediction and makes a flexible feature engineering framework. But the current method still has some limits. These include not adapting well to changing markets, not showing new tech features enough, and not being tested for different markets and product types. Future work will focus on making dynamic feature update systems, adding deep learning for full feature learning, using the method for other consumer electronics, and explaining better why features work in economic terms. This will help mix feature engineering and market knowledge more.

References

[1] Parth Bhatnagar, Gururaj H L, Shreyas J, Francesco Flammini, Disha Panwar, & Shadeeksha Shree. Prediction of mobile phone prices using machine learning. In Proceedings of the 2024 9th International Conference on Machine Learning Technologies (ICMLT), pp. 6 – 10, 2024.

[2] Çetin, M., & Koç Y. Mobile phone price class prediction using different classification algorithms with feature selection and parameter optimization. In 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 483 – 487, IEEE, 2021.

[3] Hu, N. Classification of mobile phone price dataset using machine learning algorithms. In 2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML), pp. 438 – 443, IEEE, 2022.

[4] Güvenç E., Çetin, G., & Koçak, H. Comparison of KNN and DNN classifiers performance in predicting mobile phone price ranges. Advances in Artificial Intelligence Research, vol. 1, no. 1, pp. 19 – 28, 2021.

[5] Chang, Y. J., Lin, Y. L., & Pai, P. F. Support vector machines with hyperparameter optimization frameworks for classifying mobile phone prices in multi-class. Electronics, vol. 14, no. 11, p. 2173, 2025.

[6] Saeed, A., Mukhtar, A., Arafat, Y., Abbas, M., & Saeed, A. Intelligent assessment of secondhand mobile phone prices by machine learning techniques. Journal of Computing & Biomedical Informatics, 2024.

[7] Hamid, M. T., & Abid, M. Decision support system for mobile phone selection utilizing fuzzy hypersoft sets and machine learning. Journal of Intelligent Management Decision, 3(2), 104 – 115, 2024.

[8] Sunariya, N., Singh, A., Alam, M., & Gaur, V. Classification of mobile price using machine learning. In SCI, pp. 55 – 66, 2024.

[9] Bhargav, R., Ujwal, H. U., Adithya, M., Pruthvi, M. H., Grishma, N., & Devamane, S. B. Mobile price tagger: A machine learning-driven neural network for mobile phone price classification. In 2024 International Conference on Computational Intelligence for Security, Communication and Sustainable Development (CISCSD), pp. 130 – 133, IEEE, 2024.

[10] Kumar, M., Pilania, U., & Varshney, C. Predicting mobile phone prices with machine learning. In 2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE), pp. 181 – 185, IEEE, 2023.