Data-Driven Prediction of Game Frame Rates Using Linear Regression and Random Forest Models

Yubo Wang *

Cats Academy Boston, Massachusetts, United States

* Corresponding Author Email: dadouniubi@gmail.com

Abstract. Game frame rate (FPS) is a critical indicator of user experience and hardware performance in modern video games. However, players often lack accessible tools for predicting FPS under different configurations without running costly benchmarks. This study proposes a lightweight prediction framework that quantifies GPU performance, game workload, resolution scaling, and technologies such as DLSS and Frame Generation into measurable indicators. A synthetic dataset of 1296 configuration combinations was generated to ensure coverage of diverse scenarios. Linear Regression was implemented as a baseline model, and Random Forest was introduced as an improvement, demonstrating higher accuracy and robustness in both training and testing. An interactive command-line tool was developed, allowing users to input GPU, game, and resolution information to obtain FPS predictions. Although based on synthetic data, the framework illustrates the feasibility of FPS prediction via data-driven modeling and provides a foundation for future extensions with real-world benchmarks, broader hardware and game coverage, and advanced AI models for intelligent decision support.

Keywords: FPS prediction, linear regression, random forest, GPU performance.

1. Introduction

In the context of today's rapidly developing computer graphics and game industry, the prediction and optimization of Game Frames Per Second (FPS) have become an important issue of common concern for players and hardware manufacturers. With the continuous improvement of game image quality and the rapid iteration of hardware architecture, users face many problems when choosing graphics cards and adjusting image quality settings. For example, large-scale 3A games such as Cyberpunk 2077 or Wild Darts 2 have extremely high requirements for hardware, while lightweight games such as Minecraft and League of Legends rely less on performance. Therefore, how to reasonably predict the game running frame rate under different hardware configurations and image quality parameters can not only provide reference for players, but also bring value to hardware evaluation and market decision-making.

The traditional frame rate evaluation method mainly relies on actual testing, that is, by running games in different hardware environments and collecting frame rate data to obtain benchmark results. However, this method has obvious limitations: on the one hand, the testing process is time-consuming and requires a lot of hardware support; on the other hand, the results often lack universality and cannot cover all graphics cards, resolutions and function combinations. In addition, although the benchmark test results provided by manufacturers or media are of reference to a certain extent, it is difficult for users to directly apply them to their own systems due to differences in the device environment and personal usage scenarios. This real contradiction has promoted the research of FPS prediction based on data modeling.

In recent years, machine learning methods have shown great potential in the field of performance prediction and system modeling [1, 2]. As the most basic and interpretative algorithm, Linear Regression provides a lightweight and intuitive solution for game frame rate prediction [3]. This research combines GPU Score, Game Score, Resolution Factor, DLSS and Frame Generation switches, etc. key features, constructs a synthetic dataset, and trains the regression model based on this [2]. By quantifying these subjective and objective factors into calculable indicators, the system can output the expected frame rate prediction results after entering the game name, GPU model and graphic settings [1].

This research is not only meaningful at the methodological level, but also has certain application value in practice. First of all, it provides users with an interactive prediction tool to help players quickly evaluate the performance of games with different hardware and image quality configurations in the absence of actual test data. Secondly, it provides new ideas for researchers to explore the game performance prediction model, proving that even in the absence of real test data, the combination of synthetic data and machine learning can produce reasonable inferences. Finally, this project also lays the foundation for the introduction of more complex nonlinear models (such as random forests or neural networks) and real benchmark data in the future.

2. Methods

2.1. Data Collection and Quantification

The data source of this article is mainly synthetic data, and the existing public running scores, player experience and documentation are fully referred to in the construction process. The following key characteristics were defined:

- GPU Score: According to the public running score and hardware grade, ranging from 30 (integrated graphics card) to 180 (RTX 5090) [4];
- Game demand index: according to the high/medium/low load classification, combined with ChatGPT assistance and manual fine-tuning, assign a load score of 1–180 to 90 games;
- Resolution factor: define the magnification according to the effect of resolution on performance, 1080p = 1.0, 1440p = 1.5, 4K = 2.5;
- DLSS and FrameGen: modeled as binary variables, giving 1.3 times and 1.5 times performance bonuses respectively when data is generated.

2.2. Construction of Synthetic Data Sets

This study has generated 1296 samples by combining GPU × game × resolution × DLSS × FrameGen. This method avoids the problem of insufficient coverage in the real test, while ensuring the balance and diversity of the training set. The generated data set contains five input features and a target variable: FPS [3].

2.3. Model Selection and Implementation

The initial model adopts linear regression. Its advantage is that it is simple and explainable, and it can clearly show the impact of various characteristics on FPS. However, in the experimental process, it was found that there was a large deviation in the prediction of linear regression in high nonlinear scenarios.

In order to solve this problem, this study has introduced random forest regression as an improvement plan. Random forests can capture the nonlinear interaction between features, and the generalization performance is better. The experimental results show that the fitting effect of random forests on training sets and test sets is significantly better than linear regression.

3. Results and Discussion

In the method section, the data set has been built and two models have been trained: one is the basic linear regression, and the other is the improved random forest regression. This chapter mainly shows the results of these two models, including the visualization of predictions, error comparison and further analysis.

3.1. Performance of Training Set and Test Set

The data set is proportionally divided into training set (80%) and test set (20%). The training set is used for model learning, and the test set is used to verify the prediction ability on new data.

In linear regression, the fit of the training set is not bad, but the gap between the predicted value and the real value on the test set is obvious, especially in the combination of high resolution and large 3A games, the error is the largest.

The stochastic forest performs better on both the training set and the test set, and the distribution of the predicted results is closer to the ideal diagonal (y=x).

After drawing the scatter plot, it can be seen that the horizontal axis is the real FPS, the vertical axis is the predicted FPS, and the red dotted line is the ideal prediction. The distribution of linear regression points is relatively scattered, and the deviation from the dotted line is obvious; the random forest prediction point almost fits the dotted line, and the accuracy is higher. This shows that random forests are superior to linear regression in capturing nonlinear relationships.

3.2. Model Comparison: Indicator Analysis

In order to visually compare the performance of the model, the mean square error (MSE), the average absolute error (MAE) and the decision coefficient (R^2) are used as indicators. The results are shown in Table 1 [5].

Table 1. The performance of various models

Model	Assemble	MSE ↓	MAE ↓	R^2 ↑
LR	Training Set	14.8	2.8	0.90
LR	Testing Set	22.6	3.4	0.82
RF	Training Set	3.4	1.2	0.98
RF	Testing Set	4.9	1.5	0.95

The results show that the R^2 of linear regression on the test set decreased more, and the generalization ability was limited; the random forest maintained a high level in both training and testing, with R^2 more than 0.95 and low error.

3.3. Resolution and Performance under Game Load

Further group by resolution:

At 1080p, the gap between the two models is relatively small.

Under 4K, the linear regression error increases significantly, with an average relative error of more than 12%; the random forest is still maintained at about 6%.

This shows that the higher the resolution, the more FPS is affected by complex factors such as GPU, DLSS and FrameGen, and the more nonlinear models are advantageous.

3.4. Characteristic Importance Analysis

In order to identify the main characteristics, the ablation experiment was carried out:

After removing the DLSS feature, the accuracy of the model decreased significantly;

Remove the FrameGen feature, the accuracy is reduced but the amplitude is relatively small;

When only GPU and Game are used and resolution and function switches are ignored, the accuracy rate is significantly reduced.

The characteristic importance given by the random forest is sorted as follows: resolution > game load > GPU > DLSS > FrameGen. The result is consistent with intuitive cognition [1].

3.5. Case Analysis

Case 1: RTX 4070 + RDR2 + 1440p + DLSS open

The linear regression prediction is 88 FPS, and the real is 95 FPS, with a large error.

Random forest prediction 93 FPS, closer to the real result.

Case 2: GTX 1050 + Cyberpunk 2077 + 1080p + DLSS Shut Down

The linear regression prediction is 28 FPS, and the real is 25 FPS, with a slight overestimation.

Random forest prediction 26 FPS, almost accurate.

These cases show that random forests not only have better overall performance, but also more reliable in specific scenarios.

3.6. Problems and Challenges Faced

Although the method is effective, there are still several problems:

• Insufficient authenticity of data: Because the data set is synthetic, it lacks direct support for real benchmark testing;

• Limited feature dimensions: the impact of CPU, memory, temperature, etc. on performance is not considered;

• The generalization ability needs to be verified: the applicability of the model in real scenarios still needs to be further evaluated through actual measurement data.

In summary, the methodology of this study takes into account the efficiency of modeling while ensuring lightweight and interpretability, but there is still room for further improvement and expansion.

4. Conclusion

This study develops a lightweight system for predicting game frame rates by transforming GPU performance, game demand, resolution, and DLSS/FrameGen features into quantifiable indicators. A synthetic dataset of 1296 samples was constructed, covering diverse configuration combinations. Linear regression served as the baseline, while random forest provided improved accuracy and stability, as shown in prior ensemble learning research. An interactive tool was implemented, allowing users to input GPU and game information to obtain predicted FPS. Although based on synthetic data, the approach demonstrates practical value: players can quickly evaluate graphics settings without complex benchmarks. Future improvements include integrating real-world benchmark data, using external reviews, expanding hardware and game coverage, and exploring advanced AI models capable of analyzing in-game screenshots. This highlights the feasibility of FPS prediction via data-driven modeling and outlines a pathway toward more intelligent and comprehensive tools.

References

[1] Breiman L. Random forests. Machine Learning, 2001, 45(1): 5–32. https://doi.org/10.1023/A:1010933404324

[2] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Duchesnay E. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 2011, 12: 2825–2830. https://scikit-learn.org/

[3] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. New York: Springer, 2009. https://hastie.su.domains/ElemStatLearn/

[4] TechPowerUp. GPU Database. 2023. https://www.techpowerup.com/gpu-specs/

[5] NVIDIA Corporation. NVIDIA DLSS Technology Overview. Whitepaper, 2021. https://developer.nvidia.com/dlss

[6] Chicco D, Warrens MJ, Jurman G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. Peerj computer science. 2021 Jul 5;7: e623.

[7] Serefoglu Cabuk K, Cengiz SK, Guler MG, Topcu H, Cetin Efe A, Ulas MG, Poslu Karademir F. Chasing the objective upper eyelid symmetry formula; R2, RMSE, POC, MAE, and MSE. International Ophthalmology. 2024 Jul 2;44(1):303.