13th International Conference on Applied Informatics
Eszterházy Károly Catholic University
Eger, Hungary, February 19–21, 2026

Practical Prediction of Query Execution Time in PostgreSQL Using Lightweight Machine Learning Models

Tamás Balla[a b]

[a]Department of Computational Science, Eszterházy Károly Catholic University, Eger, Hungary

[b]Doctoral School of Informatics, Eötvös Loránd University, Budapest, Hungary
balla.tamas@uni-eszterhazy.hu

Abstract

Accurate prediction of query execution time remains a critical challenge in modern database management systems. Traditional query optimizers rely on static cost models that estimate relative resource usage rather than actual wall-clock execution time. While these estimates are effective for plan selection, they often correlate poorly with real execution latency, particularly for complex analytical workloads involving joins and aggregations [1, 2]. This discrepancy limits the effectiveness of higher-level database functions such as workload scheduling, resource provisioning, and service-level agreement (SLA) management.

Recent research efforts in the area of AI for Databases have explored learned cost models and learned query optimization techniques to address these limitations [3, 4]. Although these approaches demonstrate promising accuracy, many rely on deep learning architectures or require invasive modifications to the database engine, which can hinder adoption in practical and resource-constrained environments. Consequently, there is an increasing demand for lightweight and interpretable solutions that can complement existing database systems without disrupting their core functionality [5].

In this paper, we present a practical framework for predicting query execution time in PostgreSQL using gradient boosting regression trees (GBDT). Unlike deep learning-based approaches that treat the database system as a black box, our method exploits interpretable features extracted directly from the stan-

dard `EXPLAIN (FORMAT JSON)` output. These features include operator types (e.g., Hash Join, Seq Scan), estimated cardinalities, index usage indicators, and structural characteristics of query execution plans. Importantly, all features are obtained without modifying the database engine or introducing additional runtime overhead.

We evaluate the proposed approach using the TPC-H decision support benchmark under a controlled single-user execution environment in order to isolate query performance characteristics. Preliminary experimental results indicate that the proposed lightweight machine learning model consistently outperforms PostgreSQL's native cost-based estimates when predicting actual wall-clock execution time. In particular, the learned model achieves a lower Mean Absolute Error (MAE) and a higher coefficient of determination (R^2) compared to a linear scaling of optimizer costs. Feature importance analysis further reveals that while the optimizer's cost estimate remains an informative input, plan-level structural features—such as the number and type of join operators and specific index access patterns—play a crucial role in correcting systematic biases of the cost model.

The key contribution of this work is demonstrating that accurate and practically deployable query execution time prediction can be achieved using lightweight, interpretable machine learning techniques built on standard database interfaces. The proposed framework complements traditional optimization mechanisms and enhances database observability in applied informatics settings.

References

[1] M. Akdere, U. Çetintemel, M. Riondato, E. Upfal, S. B. Zdonik: *Learning-based Query Performance Modeling and Prediction*, in: Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, ICDE '12, USA: IEEE Computer Society, 2012, pp. 390–401, isbn: 9780769547473, doi: 10.1109/ICDE.2012.64, url: https://doi.org/10.1109/ICDE.2012.64.

[2] A. Ganapathi, H. Kuno, U. Dayal, J. L. Wiener, A. Fox, M. Jordan, D. Patterson: *Predicting Multiple Metrics for Queries: Better Decisions Enabled by Machine Learning*, in: Proceedings of the 2009 IEEE International Conference on Data Engineering, ICDE '09, USA: IEEE Computer Society, 2009, pp. 592–603, isbn: 9780769535456, doi: 10.1109/ICDE.2009.130, url: https://doi.org/10.1109/ICDE.2009.130.

[3] R. Heinrich, X. Li, M. Luthra, Z. Kaoudi: *Learned Cost Models for Query Optimization: From Batch to Streaming Systems*, Proc. VLDB Endow. 18.12 (Aug. 2025), pp. 5482–5487, issn: 2150-8097, doi: 10.14778/3750601.3750699, url: https://doi.org/10.14778/3750601.3750699.

[4] R. Marcus, P. Negi, H. Mao, N. Tatbul, M. Alizadeh, T. Kraska: *Bao: Making Learned Query Optimization Practical*, in: Proceedings of the 2021 International Conference on Management of Data, SIGMOD '21, Virtual Event, China: Association for Computing Machinery, 2021, pp. 1275–1288, isbn: 9781450383431, doi: 10.1145/3448016.3452838, url: https://doi.org/10.1145/3448016.3452838.

[5] Z. Wu, R. Marcus, Z. Liu, P. Negi, V. Nathan, P. Pfeil, G. Saxena, M. Rahman, B. Narayanaswamy, T. Kraska: *Stage: Query Execution Time Prediction in Amazon Redshift*, in: Companion of the 2024 International Conference on Management of Data, SIGMOD '24, Santiago AA, Chile: Association for Computing Machinery, 2024, pp. 280–294, isbn: 9798400704222, doi: 10.1145/3626246.3653391, url: https://doi.org/10.1145/3626246.3653391.