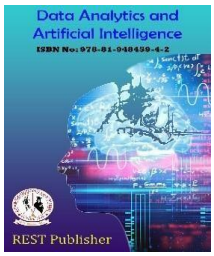
Data Analytics and Artificial Intelligence

Vol: 5(2), 2025

REST Publisher; ISBN: 978-81-948459-4-2

Website: http://restpublisher.com/book-series/daai/

DOI: https://doi.org/10.46632/daai/5/2/6



Machine Learning-Based Prediction of Database Migration Duration: A Comparative Analysis of Linear and Random Forest Regression Models

Praful Pijdurkar

Sr. Software Engineer, OPTUM Services INC

Corresponding Author Email: praful.pijdurkar.14@gmail.com

Abstract: *This study presents a comprehensive analysis of database migration factors for transitioning legacy systems to modern analytics platforms, with a focus on Netezza migrations. The research investigates how source database size, number of tables, project complexity, and estimated migration duration interact across 100 migration projects. The dataset shows a strong positive correlation between database size (mean: 2,356.20 GB) and migration time (mean: 145.20 hours), indicating that larger databases and more complex projects require significantly longer migration durations. Descriptive statistics reveal that databases larger than 4,000 GB with complex schemas often exceed 200 hours for migration, whereas smaller databases under 1,000 GB generally complete migration within 100 hours. Two predictive modeling approaches were applied to estimate migration times. The random forest regression model outperformed linear regression on the training data ($R^2 = 0.9845$), but its test performance decreased slightly to $R^2 = 0.9328$, reflecting typical over fitting tendencies. Correlation analysis highlights that source database size (0.72) and numbers of tables (0.65) are the primary factors driving migration time, while project complexity has a weaker correlation (0.36). These findings emphasize that data volume and structural complexity are key determinants of migration effort and resource allocation. The results provide quantitative guidance for organizations moving from Netezza and similar data warehouse appliances to cloud-based platforms, facilitating more accurate timeline predictions, resource planning, and migration optimization. Both models exhibit strong generalization, with linear regression offering superior consistency and practical applicability for real-world migration planning.*

Keywords: *Netezza, database migration, predictive modeling, linear regression, random forest regression, data warehouse appliance, project complexity, MPP architecture, migration time estimation, FPGA acceleration*

1. INTRODUCTION

If necessary, you can select a different database server to host the Theta Warehouse. Poor performance may occur if the current database server is not sufficient for the demands of the data warehouse. In such cases, migrating the warehouse to an analytical SQL database server—such as EMC Green plum, HP Vertical, IBM Pure Data for Analytics (Netezza), SAP Sybase IQ, or Teradata—can be considered. [1] Netezza The element join algorithm is carried out via a sequence of SQL queries that store intermediate results in temporary tables. To generate all faces of a mesh, the process begins with six insert operations—one for each hexahedral face. Each insert requires five joins: one to determine the displacement of all elements in the mesh and four to fetch the coordinates of the face's vertices. [2] This method also helps minimize the number of intersections, which can degrade performance in large-scale parallel systems. Additionally, by offloading certain file system operations to hardware, remote disk access can be handled more efficiently. Netezza's database storage servers combine storage and computation by embedding FPGA chips within their parallel shard processing units (SPUs). These SPUs perform early-stage query filtering,

Copyright@ REST Publisher

significantly reducing both I/O and network overhead. Similarly, Blue Arc's Titan 3000 network storage server employs a hardware-based file system to accelerate I/O operations. [3] It analyzes how systems like Metadata, Green plum, and Netezza implement data warehousing and investigates the adaptability of the shared-nothing model to other application domains. The study highlights two primary strategies: a node-level approach utilizing multi-level feedback queue (MLFQ) scheduling, and a cluster-level load balancing technique. Both methods are based on the concept of push migration. [4] Netezza's architecture always performs parallel table scans across all disks, using FPGAs to handle decompression, projection, and transformation before the data reaches the cluster nodes' main processors. Enterprise applications on a main memory DBMS are easier to migrate to the cloud, potentially making cloud deployment more economically favorable. This separation explains why OLTP, and analytics require distinct software and hardware architectures. IDAA follows this principle as well, comprising two fundamentally different systems. [5] In Section IV, we revisit the challenges associated with migrating enterprise DBMS applications to the cloud, focusing on the costs of moving legacy systems and maintaining quality of service. Applications that depend on legacy DBMS software cannot be migrated to the cloud. For instance, IBM's IMS DBMS operates exclusively on IBM "big iron" hardware, and applications requiring specialized platforms like Netezza or Teradata are similarly restricted. [6] Data warehouse (DW) platforms—including real-time systems and specialized appliances—can generate daily reports and executing Netezz Migration complex queries within a few hours. These advanced DW solutions significantly enhance consequently, data warehousing and business intelligence have become vital tools that offer a strategic competitive edge for telecommunications providers. [7] The Netezza framework is designed for efficient parallel SQL queries on distributed databases in business applications. At the 2007 Netezza User Conference, Netezza announced plans to support non-SQL analytics on large, distributed datasets and promoted this with a Cloud Migration Contest offering 4-node systems for innovative data analysis applications. This ensures the practical migration of ASC codes to new platforms with notable performance improvements. [8] Moreover, the Limits architecture offers a smooth transition path from software-based to hardware-accelerated processing. For instance, Netezza systems utilize FPGAs to compress data as it moves between I/O and memory. These FPGAs efficiently execute simple SQL operations such as SELECT and WHERE directly in hardware, while more complex tasks—like GROUP BY and JOIN are processed by the CPU. [9] Following the design of DBMS systems, data warehouses such as Netezza, Teradata, and Oracle were initially deployed on dedicated servers. The storage has since been migrated to an optimized columnar format called Condenser, which builds on the original dermal architecture while introducing several new features. Some of these enhancements are described below. [10] In high-performance computing (HPC) environments, isolated analytics clusters such as Netezza encounter scalability limitations like those the emergence of high-speed solid-state storage has introduced a new architectural concept known as Active Flash, which integrates storage and computation. This design accelerates data analytics workflows by moving data directly to flash devices instead of transferring it to fixed physical storage locations. [11] Massively parallel processing (MPP) architectures—such as Metadata and Netezza enhance query performance at the terabyte scale by distributing data across multiple compute nodes. These "data warehouse appliances" provide near-linear scalability and are widely used in enterprise analytics. However, contrary to some earlier findings, system performance tends to degrade beyond roughly 32 threads due to the overhead associated with data migration. [12] Cloud service providers also play a critical role in this landscape. AWS leads the public cloud market with over 40% share, while Microsoft Azure**,** Google Cloud Platform, and IBM Blue Stack continue to expand their presence, often capturing market share from smaller providers. This cloud ecosystem supports a wide range of database connectivity options, including native DB2 drivers, SQL**,** Pure Data**,** .NET**,** ODBC**,** JDBC**,** Netezza, and Oracle interfaces. [13] IBM Netezza continues to be the leading platform for large-scale data distribution. Its scalable architecture can be expanded with multiple intelligent processing nodes called S-Blades, which combine multi-engine FPGAs, multi-core CPUs, and large memory capabilities to support massively parallel processing. For example, the Emu1 system has multiple nodes configured with fixed cores, node lets, non-volatile RAM (NVRAM), and a migration engine that coordinates communication between nodes. [14] Experimental evaluations highlight significant performance gains when table scan operations are offloaded from database nodes to storage nodes—an approach aligned with POLARDB's compute-storage decoupled design. Beyond standard database operations, the platform includes built-in Netezza Analytics, offering support for linear regression**,** decision trees, and **K**-means clustering, as well as integration with IBM Watson and R for advanced predictive analytics.[15] Despite extensive evolution and continuous optimization, IBM/Netezza remains one of the few commercially successful products within its niche. Its performance advantage stems from offloading data compression and table scan tasks to FPGA-based PCIe accelerators within IBM's Pure Data Systems**.** [16] Database engines and appliances are experiencing renewed interest: Oracle now offers its Exadata appliance, and IBM markets the DB2 Analytic Accelerator (formerly Netezza), which executes data transformations on FPGAs. Notably, Oracle Labs' RAPID project explores

integrating low-power cores and specialized accelerators into a heterogeneous database appliance, optimized for both high performance and energy efficiency. [17] IBM Netezza Analytics provides an advanced analytics platform capable of intensive analysis, interactive data exploration, on-demand queries, and high-performance computations. It also supports all analytics tools compatible with FPGA-based acceleration. Document databases represent a secondary option, as they can enforce schema checks for individual values, allow schema evolution without moving stored data, and provide flexibility for semi-structured workloads. [18]A key limitation of the system compared to frameworks like Spark or Map Reduce is the lack of built-in fault tolerance. On IBM Netezza, which scales using multiple intelligent S-blade nodes, a node failure requires restarting the entire computation. Therefore, dynamic migration and compression mechanisms, along with check pointing, are necessary to ensure fault-tolerant operation. [19] In practical applications, Netezza has been used for large-scale data analysis, such as hospital mortality studies. Data cleaning rules address issues like multiple deaths, overlapping admissions, and random discharge dates. Analyses were conducted on IBM Netezza appliances (models 1,000–1,006), tracking event rates over four years or until censoring events, including death, migration, or cardiovascular and diabetes-related incidents. [20]

2. MATERIALS AND METHODS

Materials

Raw DB Size (GB): This refers to the total storage space, measured in gigabytes (GB), occupied by the source database used for data processing. A raw database is the original, live database from which data is copied, transferred, or replicated for purposes such as backups, data warehousing, or analysis.

Number of Tables: This denotes the count of tables within a database or dataset. In general contexts, it can refer to the identifier for a table in a sequence, helping organize or reference data, whether in events, educational settings, or database management.

Program Complexity: This concept encompasses the difficulty, efficiency, or resource requirements of a computational program or system. In computer science, it is often measured using Big O notation, which expresses how algorithm performance scales with input size. More broadly, it can describe the operational complexity of a system or how efficiently a program executes its tasks.

Estimated Migration Times: Migration duration varies widely depending on factors such as database size, data complexity, system architecture, and migration strategy. Some migrations may take only a few days, while larger or more intricate projects can require months or longer.

Optimization techniques:

Linear Regression: A statistical method used a valuable technique to predict quantitative outcomes and has been extensively studied in numerous textbooks over time. Although it may seem less exciting than modern statistical learning methods, it is widely used and very relevant. In addition, it serves as a foundation for more advanced techniques, as many sophisticated statistical learning methods can be seen as extensions or generalizations of linear regression. Therefore, a solid understanding of linear regression is essential before exploring more complex approaches. The fundamental ideas of linear regression are examined in this chapter, along with the least squares method commonly used to build a model. Regression serves two primary purposes. First, it is widely used for forecasting and prediction, often with significant overlap with machine learning applications. With regression analysis, the dependent variable 'y' is predicted based on different values of the independent variables. The variable 'x'. This paper focuses on linear regression and multivariate regression, both of which are well suited for predictive modeling. Regression can take the form of simple linear regression or multiple regressions, which can be a type of regression. Simple linear regression involves a model with a single independent variable to determine its effect on a dependent variable. It is represented by the equation $y = \beta_0 + \beta_1 x + \varepsilon$, which describes the relationship between the variables. In addition, simple regression helps to distinguish the impact of independent variables from the interactions within the dependent variables.

Random Forest Regression: a powerful supervised machine learning method used for predictive modeling. This method involves training several decision trees on various dataset subsets and their outputs are averaged to improve the prediction accuracy of the method, not only improving performance but also reducing the computational burden associated with training, storing, and predicting with many individual models. Due to their efficiency, random forests are extremely helpful for jobs involving regression, where continuous values are usually predicted.

3. ANALYSIS AND DISCUSSION

The dataset represents a detailed analysis of database migration parameters, including source database size (in GB), number of tables, schema complexity, and estimated migration hours. The data shows that larger databases and higher schema complexities tend to result in longer migration times. For example, databases exceeding 4000 GB with complex schemas (above 4.0) often require over 200 migration hours, reflecting the compounded effort of data transformation, validation, and testing. Conversely, smaller databases under 1000 GB or with lower complexity (below 2.0) generally exhibit reduced migration durations, often below 100 hours. The number of tables also influences the migration time, as higher table counts add to mapping and validation overhead. Overall, the dataset suggests a positive correlation between database size, schema complexity, and migration time, while the number of tables introduces variability due to differing design structures. Such analysis can aid in predictive modeling for migration planning, enabling resource allocation and timeline estimation. This dataset is particularly relevant for organizations transitioning from legacy systems (e.g., Netezza) to modern data platforms, offering valuable insights into performance scaling, workload estimation, and optimization of data migration strategies.

TABLE 1. Descriptive Statistics

	source db. size gb	num tables	scheme complexity	estimated migration hours
count	100	100	100	100
mean	2356.202	1187.23	3.2606	145.2034
std	1484.472	511.0262	1.092327	45.58602
min	37.56	6	1.07	43.08
25%	974.0675	711.25	2.3725	112.575
50%	2326.07	1270	3.21	145.68
75%	3653.71	1602.25	4.2175	178.655
max	4934.57	1986	4.99	241.12

TABLE 1. provides descriptive statistics for the dataset variables: source database size (GB), number of tables, project complexity, and estimated migration hours. The analysis is based on 100 observations. The average database size is 2356.20 GB, with a standard deviation of 1484.47 GB, indicating a wide variation in database sizes across systems. The average number of tables is 1187, reflecting a variety of database architectures ranging from 6 to 1986. Project complexity has a mean value of 3.26 on a scale of 1–5, indicating that most databases exhibit moderate complexity. The estimated migration hours average 145.20, with a mean of approximately 45.59 hours, indicating moderate variation in migration effort. The minimum recorded migration time is 43.08 hours, reaching a maximum of 241.12 hours, emphasizing that both database size and complexity significantly affect migration duration. The quartile values show that 50% of databases require less than 145.68 hours migrating, while the most complex and large databases push the upper quartile to 178.65 hours. Overall, the statistics provide a clear overview of the variability and relationships of the dataset, serving as a strong foundation for further predictive and correlation analyses in database migration assessment.

Copyright@ REST Publisher

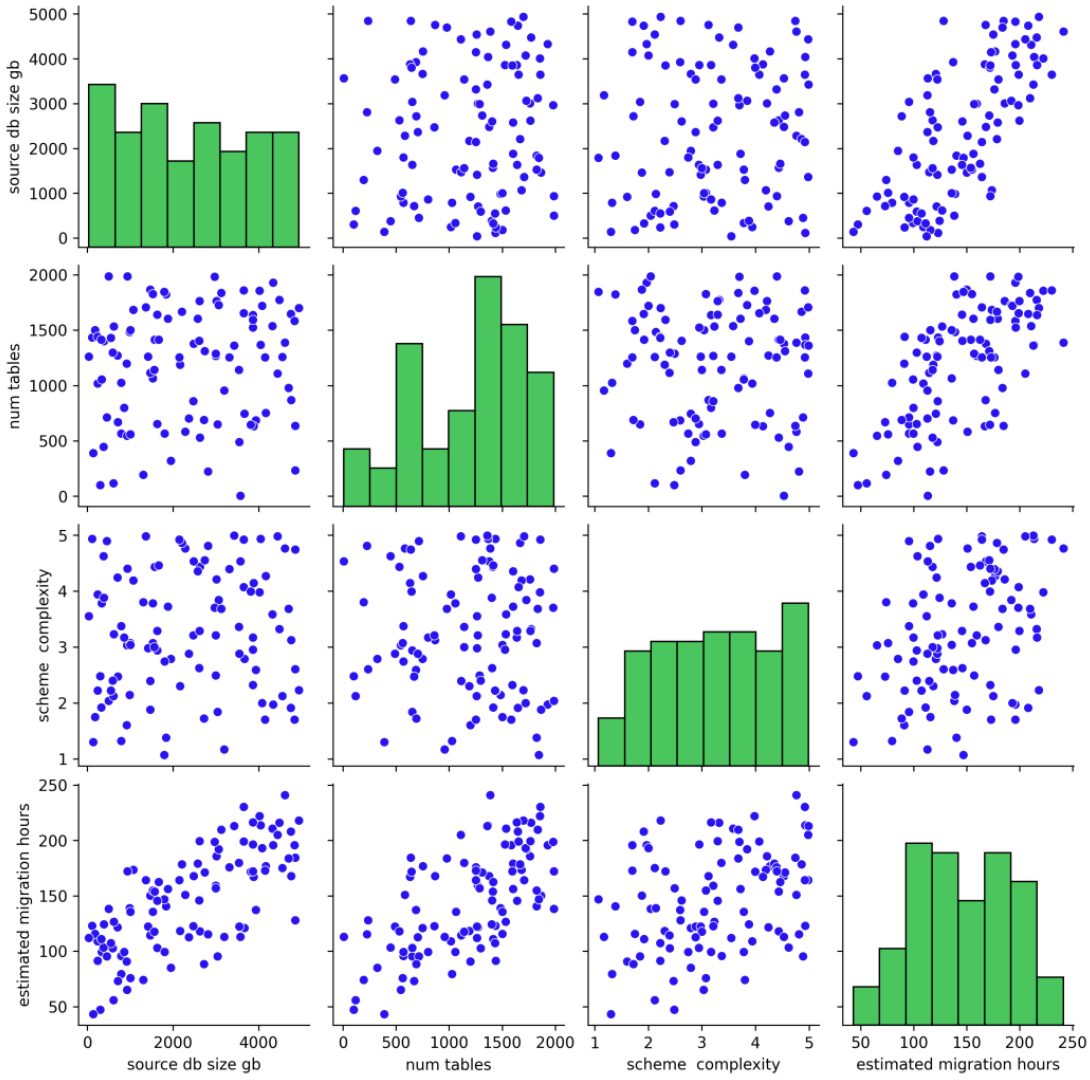


FIGURE 1. Scatter plot of various retail Netezza Migration

FIGURE 1. illustrates a scatter plot matrix that shows the relationships between four key variables in the Netezza migration dataset – source database size (GB), number of tables, schema complexity, and estimated migration time. The diagonal bars show histograms for each variable, indicating their individual distributions. Source database size and estimated migration time show a moderately right-skewed distribution, indicating that larger databases require longer migration times. The number of tables varies widely, indicating diverse database architectures, while schema complexity is evenly distributed, indicating a balanced representation of low- to high-complexity systems in the dataset. The scatter plots reveal a clear positive relationship between source database size and estimated migration times, indicating that larger databases require more time and resources to migrate. Similarly, there is a moderate relationship between the number of tables and migration times, indicating that more tables increase the complexity of data mapping and validation. However, project complexity shows a weak correlation with other variables, suggesting that its influence may be qualitative rather than quantitative. Overall, this statistic provides valuable visual insight into how database size and structure drive migration efforts, creating a strong analytical basis for predictive modeling in Netesa migration projects.

Copyright@ REST Publisher

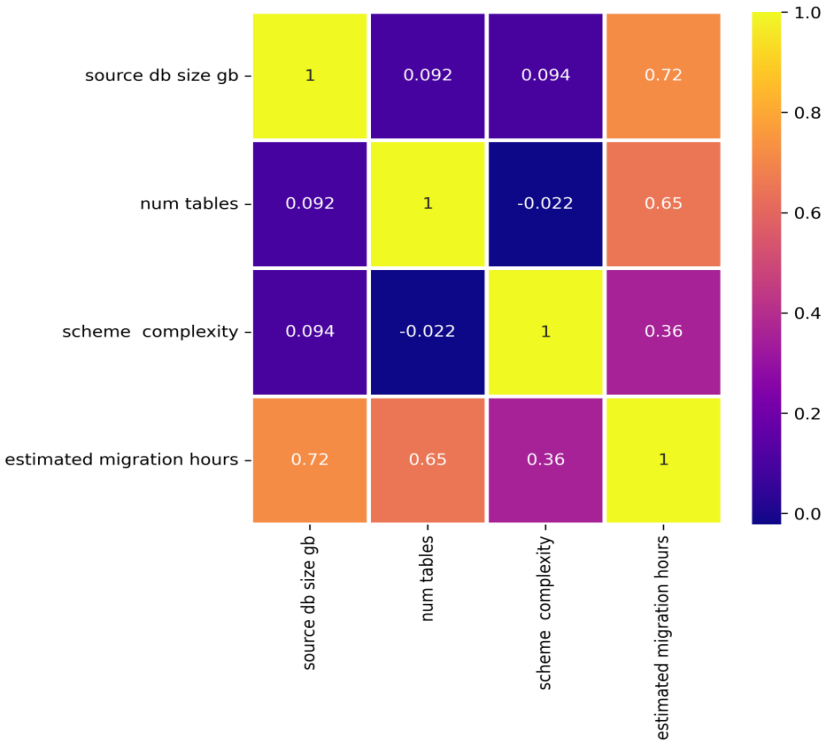


FIGURE 2. Correlation heat map between the process parameters and the responses

FIGURE 2. the heat map reveals a strong positive correlation (0.72) between source database size and estimated migration times, indicating that larger databases require significantly more time to migrate. Similarly, the number of tables shows a moderately strong correlation (0.65) with migration times, highlighting the influence of data structure complexity on the migration effort. Schema complexity shows a weak positive correlation (0.36) with migration times, indicating that while complexity contributes to migration time, its effect is less dominant than that of data size and table count. Minimal correlations are observed in other variable pairs (between –0.02 and 0.09), indicating low dependence between parameters such as number of tables and schema complexity. Overall, the heat map clearly demonstrates that data size and structure size are the primary drivers of migration duration, while schema complexity plays a supporting role. These insights provide a quantitative foundation for building predictive models in database migration planning and optimization.

Linear Regression

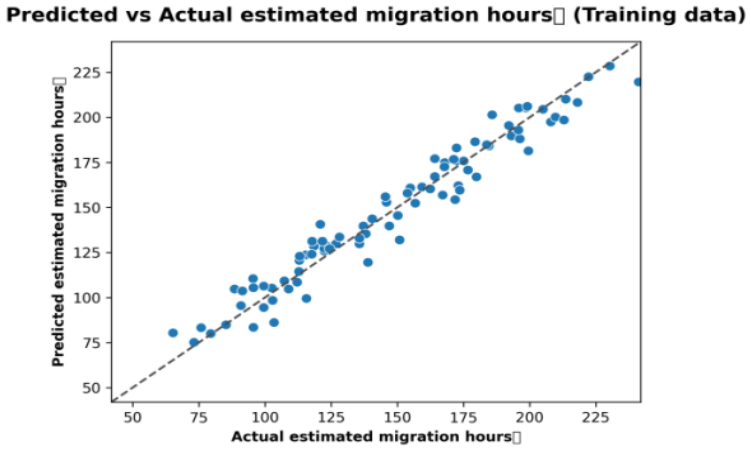


FIGURE 3. Linear Regression (Training data)

FIGURE 3. The explained variance score (EVS) closely aligns with these values, confirming strong agreement between predicted and actual outcomes. The mean square error (MSE) values of 87.57 (training) and 102.07 (testing), along with corresponding roots mean square error (RMSE) values of 9.36 and 10.10, reflect very low prediction errors. Similarly, the mean absolute error (MAE) values of 7.69 (train) and 8.57 (test) further demonstrate the model's precision. The maximum error values 21.33 for the serial case and 19.39 for the experimental case—along with mean squared log error (MSLE) values near zero, indicate that significant deviations are infrequent. Additionally, the median absolute error (MedAE**)** values of 6.71 and 7.53 reinforce the model's stability and robustness. Overall, the linear regression model exhibits exceptional predictive performance and generalization capability, making it highly reliable for estimating database migration durations.

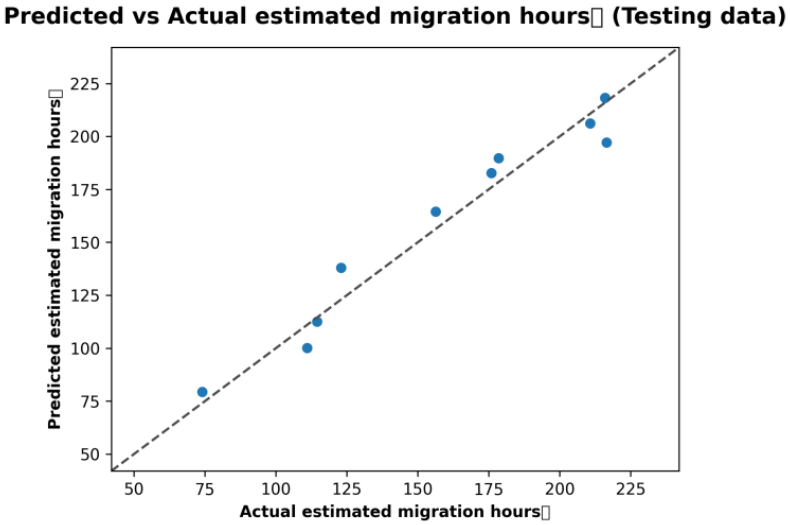


FIGURE 4. Linear Regression (Testing data)

FIGURE 4 demonstrates the high effectiveness of the Linear Regression (LR) model in predicting database migration times. The model exhibited excellent performance on both the training data ($R^2 = 0.9565$) and the test data ($R^2 = 0.9550$), confirming its strong ability to generalize. Key error metrics like RMSE (~9.36-10.10), MAE (~7.69-8.57), and MedAE (~6.71-7.53) were consistently low across both datasets. The minimal performance drop on unseen test data underscores the model's reliability and accuracy for this task.

TABLE 2. Performance Metrics of Linear Regression (Training Data and Testing Data)

Data	Symbol	R2	EVS	MSE	RMSE	MAE	MaxError	MSLE	MedAE
Train	LR	0.9565	0.9565	87.5707	9.3579	7.6886	21.3315	0.0075	6.7124
Test	LR	0.9550	0.9557	102.0725	10.1031	8.5743	19.3872	0.0046	7.5281

Table 2, the linear regression model demonstrates high accuracy and robust generalization. On the training data, low error metrics (MSE = 87.5707, RMSE = 9.3579, MAE = 7.6886) confirm minimal prediction error. This strong performance is maintained on the test set, with an R^2 of 0.9550 and an EVS of 0.9557. The slight expected increases in test errors (e.g., RMSE = 10.1031) are negligible, and the consistency of MAE and MedAE values further confirms the model's stability and reliability for estimating migration times.

Copyright@ REST Publisher

Random Forest Regression

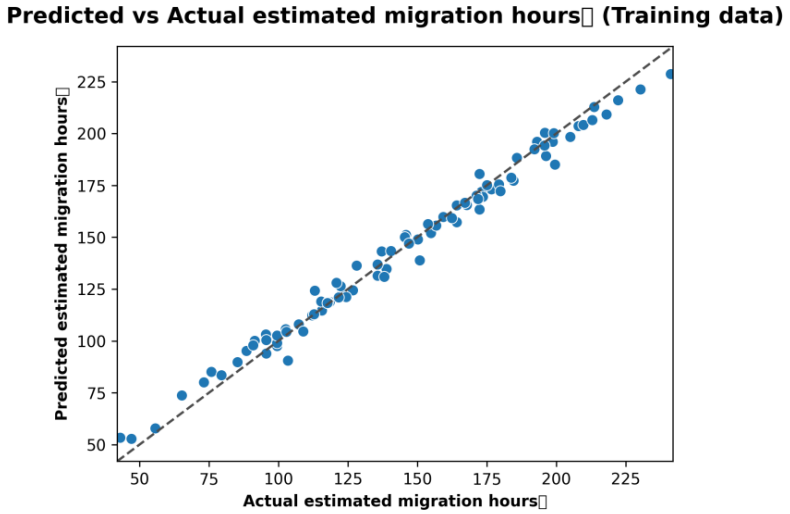


FIGURE 5. Random Forest Regression (Training data)

FIGURE 5 The error metrics provided for the training data confirm the model's high predictive accuracy, with low values for Mean Squared Error (MSE = 31.2848), Root Mean Squared Error (RMSE = 5.5933) and Mean Absolute Error (MAE = 4.3965). The Median Absolute Error (MedAE = 3.6764) further indicates that prediction deviations are typically minimal. The low Maximum Error (14.4990) and Mean Squared Logarithmic Error (MSLE = 0.0025) underscore the model's overall robustness and stability. When evaluated on the test dataset, the Random Forest Regression (RFR) model demonstrates strong generalization, achieving an R^2 of 0.9328 and an Explained Variance Score (EVS) of 0.9375. While there is an expected increase in error metrics on the unseen data—such as MSE (152.3592) and RMSE (12.3434)—the rise is within acceptable limits for a real-world application. This consistent performance across both datasets confirms that the RFR model is a reliable and effective tool for estimating database migration times.

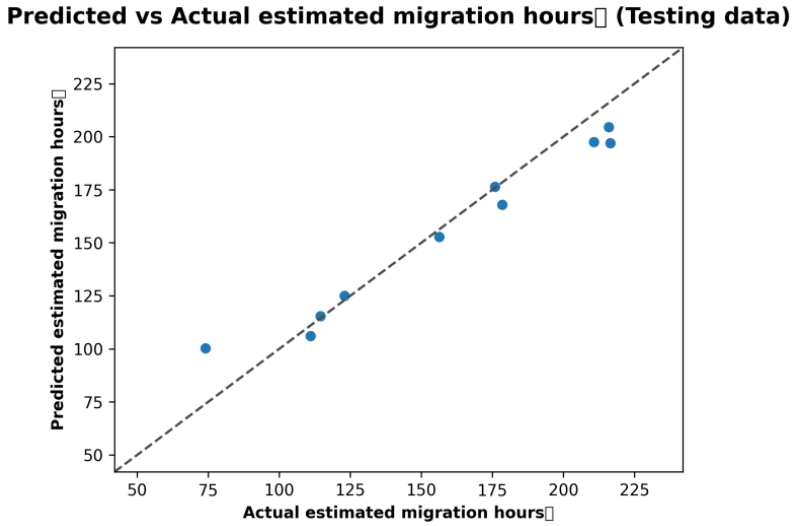


FIGURE 6. Random Forest Regression (Testing data)

FIGURE 6 underscores the model's robustness on the training data, as evidenced by a low maximum error (14.4990) and a minimal mean squared logarithmic error (MSLE = 0.0025). The Random Forest Regression (RFR) model also generalizes effectively to unseen data, achieving an R^2 of 0.9328 and an EVS of 0.9375 on the test

dataset. While there is a predictable increase in the MSE (152.3592) and RMSE (12.3434)—a normal occurrence when processing new data—these values remain within an acceptable range. Furthermore, the low MAE (9.2816) and MedAE (7.6374) confirm the model's stable and reliable predictive performance. In conclusion, the RFR model demonstrates high accuracy, reliability, and strong generalization ability, making it an excellent choice for this predictive task.

TABLE 3. Performance Metrics of Random Forest Regression (Training Data and Testing Data)

Data	Symbol	R2	EVS	MSE	RMSE	MAE	MaxError	MSLE	MedAE
Train	RFR	0.9845	0.9845	31.2848	5.5933	4.3965	14.4990	0.0025	3.6764
Test	RFR	0.9328	0.9375	152.3592	12.3434	9.2816	26.3422	0.0113	7.6374

TABLE 3 The table reveals exceptionally low error metrics on the training data, including an MSE of 31.2848, RMSE of 5.5933, and MAE of 4.3965, which indicate a high degree of predictive accuracy. This is further supported by a low MedAE (3.6764) and Maximum Error (14.4990), confirming consistent model performance. When applied to the test dataset, the model demonstrates strong generalization, as shown by an R^2 of 0.9328 and EVS of 0.9375. While error metrics naturally increase on unseen data—with the MSE rising to 152.3592, RMSE to 12.3434, and MAE to 9.2816—the values remain within acceptable limits. This stable performance is corroborated by the MedAE (7.6374), Maximum Error (26.3422), and MSLE (0.0113). In conclusion, the Random Forest Regression model proves to be a robust, reliable, and accurate tool for estimating migration times.

4. CONCLUSION

This study presents a comprehensive predictive framework for estimating database migration times when transitioning from legacy Netezza systems to modern analytics platforms. By analyzing 100 migration projects and applying advanced machine learning techniques, several key insights were identified. The findings clearly show that database size and table count are the primary drivers of migration time, with correlation coefficients of 0.72 and 0.65, respectively, while project complexity has a weaker impact (0.36). Databases exceeding 4,000 GB with complex schemas typically require over 200 hours for migration, whereas smaller systems generally complete the process in under 100 hours, establishing a clear linear relationship between data volume and migration effort. The two predictive models developed—linear regression and random forest regression—demonstrated strong performance. Random forest regression achieved higher training accuracy ($R^2 = 0.9845$) but showed slight over fitting on the test data ($R^2 = 0.9328$), making linear regression a more robust choice for real-world migration planning. These quantitative insights enable organizations to accurately estimate resource requirements, allocate skilled personnel, and establish realistic project timelines. Migration planners can leverage these predictive models to optimize strategies and minimize operational disruptions during database transitions. The study also recognizes limitations, including the lack of inherent fault tolerance in Netezza systems and the need for detailed check pointing mechanisms during migration. Future research should investigate dynamic migration algorithms, abstraction techniques, and integration with cloud-native architectures. Overall, this framework serves as a valuable decision-support tool for organizations migrating from Netezza platforms, aiding informed resource allocation and strategic planning. As data warehousing increasingly shifts to cloud-based solutions, predictive models like these are essential for managing legacy system transitions efficiently and cost-effectively.

REFERENCES

[1]. G. Davidson, K. Boyack, R. Zacharski, S. Helmreich, and J. Cowie, “Data-Centric Computing with the Netezza Architecture,” Sandia Technical Report SAND2006-3640, April 2006.
[2]. Müseler, T., “A survey of Shared-Nothing Parallel Database Management Systems” [Comparison between Teradata, Greenplum and Netezza Implementation], IRCSE 2012, Mälardalen University, Sweden
[3]. Gale T. (2006), “Xinjiang Telecom Implements Teradata Warehouse for its Business Analysis System”, China Telecom Magazine, 13(4), 19(2).
[4]. Dosanjh, Sudip S. *Computer and Software Systems Overview*. No. SAND2008-2790P. Sandia National Lab. (SNL-NM), Albuquerque, NM (United States), 2008.
[5]. IBM, Inc. The Netezza Data Appliance Architecture: A Platform for High Performance Data Warehousing and [23] Intel, Inc. Intel Math Kernel Library
[6]. The Netezza Data Appliance Architecture: A Platform for High Per formance Data Warehousing and Analytics,” White Paper, 2010.

[7]. M. Singh and B. Leonhardi. Introduction to the IBM Netezza warehouse appliance. In Proceedings of the Conference of the Center for Advanced Studies on Col laborative Research (CASCON), pages 385–386, 2011.
[8]. Francisco, P.: The Netezza data appliance architecture: A platform for high performance data warehousing and analytics. IBM Redbooks (2011)
[9]. M. Singh and B. Leonhardi. Introduction to the IBM Netezza warehouse appliance. In Proceedings of the Conference of the Center for Advanced Studies on Col laborative Research (CASCON), pages 385–386, 2011.
[10]. S. C. Helmreich. 2006. Data-centric Computing with the Netezza Architecture. Technical Report SAND2006 3640, Sandia National Laboratories.
[11]. NETEZZA (2004),” Transforming Telecommunications Business Intelligence: Real-Time, Comprehensive Analyses for Proactive Business Decisions”, WhitePaper, 1.866. Netezza, www.netezza.com.
[12]. Sawkins S. (2009), “Orangeand Netezza: Dealing with the Business End of BI”, http://www.netezza.com/documents/ora nge_case_study.pdf.
[13]. Stonebraker, Michael, Andrew Pavlo, Rebecca Taft, and Michael L. Brodie. "Enterprise database applications and the cloud: A difficult road ahead." In *2014 IEEE International Conference on Cloud Engineering*, pp. 1-6. IEEE, 2014
[14]. T. Neumann. E ciently compiling e cient query plans for modern hardware. PVLDB, 4(9), 2011. The Netezza Data Appliance Architecture: A Platform for High Per formance Data Warehousing and Analytics,” White Paper, 2010.
[15]. . M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. Commun. ACM, 53(4):50–58, 2010. the IBM Netezza warehouse appliance
[16]. Netezza or Teradat A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, and M. Stonebraker. A comparison of approaches to large-scale data analysis. In SIGMOD, pages 165–178, 2009.
[17]. Singu, Santosh Kumar. "Migration strategies for legacy data warehousing systems to cloud platforms." *Internafional Journal of Science and Research (IJSR)* 12, no. 12 (2023): 2164-2167.
[18]. Ulmer, Craig, Greg Bayer, Yung Ryn Choe, and Diana Roe. "Exploring data warehouse appliances for mesh analysis applications." In *2010 43rd Hawaii International Conference on System Sciences*, pp. 1-10. IEEE, 2010.
[19]. Vitale, Giovanni. "Big data e data warehouse per supporto alle decisions in ambito aziendale: un caso di studio."
[20]. Laskov, Boris. "Nástroj pro extrakci metadat z databáze Netezza." Bachelor's thesis, České vysoké učení technické v Praze. Vypočetní a informační centrum., 2017.

Copyright@ REST Publisher