Query, Don't Train: Privacy-Preserving Tabular Prediction from EHR Data via SQL Queries

Josefa Lia Stoisser [*1] **Marc Boubnovski Martell** [*1] **Kaspar Märtens** [1] **Lawrence Phillips** [1]
Stephen Michael Town [1] **Rory Donovan-Maiye** [1] **Julien Fauqueur** [1]

Abstract

Electronic health records (EHRs) contain richly structured, longitudinal data essential for predictive modeling, yet stringent privacy regulations (e.g., HIPAA, GDPR) often restrict access to individual-level records. We introduce **Query, Don't Train** (QDT): a **structured-data foundation-model interface** enabling **tabular inference** via LLM-generated SQL over EHRs. Instead of training on or accessing individual-level examples, QDT uses a large language model (LLM) as a schema-aware query planner to generate privacy-compliant SQL queries from a natural language task description and a test-time input. The model then extracts summary-level population statistics through these SQL queries, and the LLM performs chain-of-thought reasoning over the results to make predictions. This inference-time–only approach enables prediction without supervised model training, ensures interpretability through symbolic, auditable queries, naturally handles missing features without imputation or preprocessing, and effectively manages high-dimensional numerical data to enhance analytical capabilities. We validate QDT on the task of 30-day hospital readmission prediction for Type 2 diabetes patients using a MIMIC-style EHR cohort, achieving F1 = 0.70, which outperforms TabPFN (F1 = 0.68). To our knowledge, this is the first demonstration of LLM-driven, privacy-preserving structured prediction using only schema metadata and aggregate statistics—offering a scalable, interpretable, and regulation-compliant alternative to conventional foundation-model pipelines.

[*]Equal contribution [1]Novo Nordisk, London, UK. Correspondence to: Josefa Lia Stoisser <ofsr@novonordisk.com>, Marc Boubnovski Martell <mbvk@novonordisk.com>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

1. Introduction

EHRs store richly structured, longitudinal data spanning diagnoses, laboratory results, procedures, medications, and outcomes—resources that are critical for predictive modeling and clinical decision support (Kim et al., 2019; Tsai et al., 2025). However, regulations such as the U.S. HIPAA Privacy Rule and the EU GDPR impose strict safeguards for protected health information, including consent, minimization, and access controls, with substantial legal and institutional constraints on data use (Cohen & Mello, 2018; Voigt & Von dem Bussche, 2017). These policies often prohibit direct access to patient-level records, creating significant barriers for model development, particularly in cross-institutional settings where data-sharing agreements are difficult to establish or enforce.

Despite these constraints, public datasets such as MIMIC-III have enabled research in EHR-driven prediction under carefully controlled conditions, supporting tasks such as mortality forecasting, hospital readmission risk, and treatment efficacy modeling (Johnson et al., 2020; Meng et al., 2022). Traditional supervised models—especially tree-based methods like XGBoost—continue to dominate tabular prediction tasks due to their robustness to heterogeneous features, irregular target functions, and missing data (Grinsztajn et al., 2022; Yu et al., 2024; McElfresh et al., 2023). Transformer-based in-context learners, such as TabPFN, offer classification via training-set conditioning, though they still require access to raw examples at inference time (Hollmann et al., 2022; den Breejen et al., 2024; Qu et al., 2025; Bai et al., 2023).

LLMs have recently demonstrated strong performance both in structured reasoning tasks, including text-to-SQL translation (Gao et al., 2023), and tabular prediction tasks (Hegselmann et al., 2023; Kim et al., 2025). Recent advancements have introduced privacy-preserving techniques and agent-based frameworks to address these challenges (Liu et al., 2025). Deep learning models can be trained across decentralized datasets using federated learning, enabling collaborative prediction without sharing sensitive data (Abadi et al., 2016; Chua et al., 2024; Kuang et al., 2024; Wang et al., 2025). Agent-based frameworks allow models to au-

tonomously perform multi-step reasoning over structured data, facilitating complex clinical decision-making (Liu et al., 2025).

LLMs have recently demonstrated strong performance both in structured reasoning tasks, including text-to-SQL translation (Gao et al., 2023), and tabular prediction tasks (Hegselmann et al., 2023; Kim et al., 2025). Recent advancements have introduced privacy-preserving techniques and agent-based frameworks to address these challenges (Liu et al., 2025). Deep learning models can be trained across decentralized datasets using federated learning, enabling collaborative prediction without sharing sensitive data (Kuang et al., 2024; Wang et al., 2025). Additionally, advancements in neural network-based techniques emphasize algorithmic strategies for learning while safeguarding sensitive information through differential privacy (Abadi et al., 2016; Chua et al., 2024). Another promising approach is CRYPTEN, a software framework that enables secure multi-party computation (MPC) for machine learning, allowing parties to collaboratively train models on private datasets while ensuring data privacy (Knott et al., 2021). Agent-based frameworks allow models to autonomously perform multi-step reasoning over structured data, facilitating complex clinical decision-making (Liu et al., 2025).

These capabilities suggest a new opportunity: using LLMs not just for text generation, but for **schema-aware query planning** that operates under privacy constraints. SQL serves as a controlled, interpretable interface that enables LLMs to retrieve relevant aggregate statistics—without exposing individual-level data—thereby preserving compliance with HIPAA and GDPR (Cohen & Mello, 2018; Voigt & Von dem Bussche, 2017).

In this work, we introduce **Query, Don't Train**, a two-stage, framework for clinical tabular prediction without direct access to raw EHR data. Our approach is grounded in three pillars:

- **Privacy preservation**, by ensuring only policy-compliant SQL queries are issued and no patient-level data is revealed.
- **Structured reasoning**, which derives interpretability from two key sources: (1) LLM-mediated chain-of-thought predictions over query results, and (2) the symbolic, auditable queries themselves.
- **Robustness to missing data**, as the model dynamically selects and conditions on available features at inference without imputation.

We validate our approach on 30-day readmission prediction in a MIMIC-style cohort for Type 2 diabetes patients, showing that it obtains an F1-score of 0.70 while offering interpretability and compliance out of the box.

2. Methodology

2.1. Problem Formulation

We consider a tabular classification task under strict access constraints. Let $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ denote a training set of patient records $x_i \in \mathbb{R}^d$ and associated outcomes $y_i \in \mathcal{Y}$. Direct access to $\mathcal{D}_{\text{train}}$ is prohibited due to regulatory or institutional privacy restrictions. Given a test-time instance x^{test} from the test set and the database schema of $\mathcal{D}_{\text{train}}$, the goal is to predict its label y^{test} by interacting with $\mathcal{D}_{\text{train}}$ exclusively via a privacy-compliant SQL interface that enforces data governance policies.

2.2. Framework Overview

Our method adopts a two-stage architecture in which an LLM serves as both a query-generation agent and a predictor through structured reasoning. The process, illustrated in Figure 1, proceeds as follows:

1. **Input:** The LLM receives (i) a natural language prompt describing the prediction task (e.g., "Predict 30-day readmission for Type 2 diabetes"), and (ii) the test-time patient record x^{test}.
2. **Query Generation:** Based on the prompt and x^{test}, the agent generates SQL queries targeting the database containing $\mathcal{D}_{\text{train}}$. These queries are designed to retrieve summary-level statistics (e.g., "average length of stay for similar patients").
3. **Privacy Filtering:** Only queries that comply with predefined privacy constraints (e.g., returning aggregates over groups of at least 2 individuals) are executed.
4. **Query Loop:** The agent may iteratively generate follow-up queries to refine its understanding of relevant cohort-level statistics.
5. **Prediction:** The outputs of the executed queries are returned to the LLM, which uses chain-of-thought reasoning to produce a prediction for y^{test}.

This inference-time-only framework enables structured prediction without accessing raw patient data. The agent implicitly performs dynamic feature selection by deciding which summary statistics to request during the Query Loop.

3. Experiments

3.1. Experimental Setup

In our experimental setup, we utilize OpenAI's o4-mini model as the LLM agent, which also serves as the basis for our LLM-only baseline. To implement the agent, we

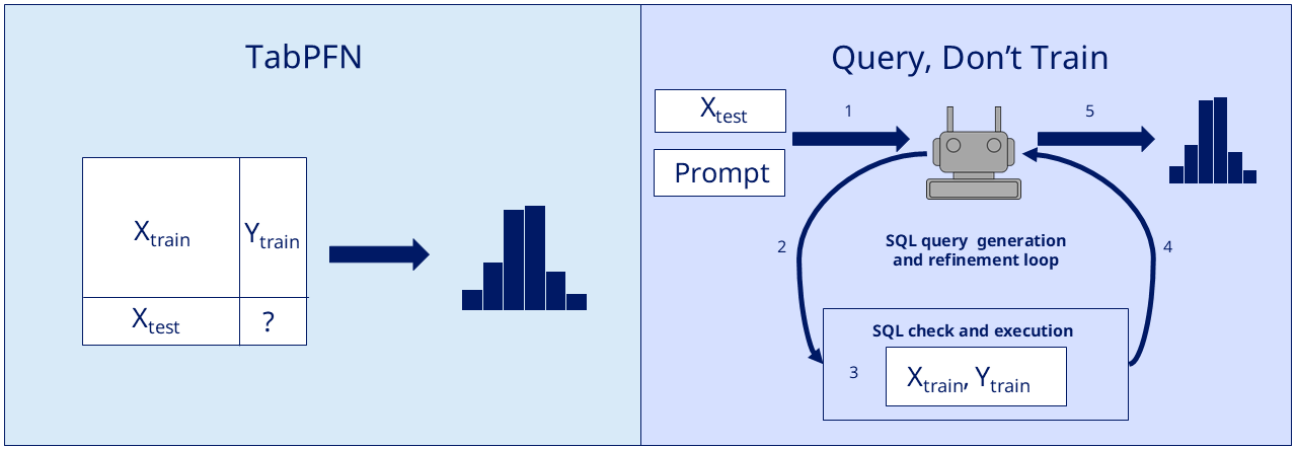

Figure 1. **Comparison of TabPFN and our "Query, Don't Train" (QDT) approach.** TabPFN uses the training set directly during inference. In contrast, QDT follows: (1) receive test record and task prompt, (2) generate SQL queries, (3) enforce compliance with privacy policies, (4) execute approved queries to retrieve summary statistics, (5) predict using chain-of-thought reasoning. QDT enables privacy-preserving, interpretable inference without raw data access.

leverage the LangChain library[1] . For an example of a run, please refer to Appendix B, where we provide details on each SQL command used for interpretability.

To comply with privacy policies, we restricted queries via the system prompt to provide only summary-level statistics, defined as data aggregated and averaged over two or more patients. This constraint is enforced using a separate LLM to validate that only queries requesting summary-level statistics are allowed to proceed to execution. The LLM is prompted with a predefined list of rules (e.g. no queries on cohorts, limited repeated or overlapping queries, monitor query patterns) and can determine whether a query meets the specified criteria. In practice, this validation would be implemented through a firewall to prevent unauthorized data access (Kruse et al., 2017).

3.2. Datasets

We focus on predicting 30-day hospital readmissions for patients with Type 2 Diabetes in US hospitals (Clore & Strack, 2014)[2]. The dataset consists of patient records x_i, which include demographics, laboratory results, procedures, and prior admissions, with binary outcome labels $y_i \in \{0, 1\}$. We conducted K-fold cross-validation with 5 iterations, utilizing a subset of 2,000 randomly sampled patients in each fold. With approximately 12% of patients readmitted within 30 days in our evaluation subset, this approach effectively addresses the imbalanced nature of the readmission task.

[1] https://python.langchain.com/api_reference/community/agent_toolkits/langchain_community.agent_toolkits.sql.toolkit.SQLDatabaseToolkit.html

[2] https://www.kaggle.com/c/1056lab-diabetes-readmission-prediction/data

3.3. Baselines

We compare our method against three baselines: TabPFN (Hollmann et al., 2022) is a pre-trained transformer-based predictor trained to perform tabular classification by conditioning on the training set at inference time. It is particularly relevant as it accesses $\mathcal{D}_{\text{train}}$ during inference, similar in spirit to our method, albeit without privacy constraints. XGBoost (Chen & Guestrin, 2016) is a widely-used gradient boosting framework for tabular data. We train XGBoost on the training set $\mathcal{D}_{\text{train}}$ and evaluate it on the test set, representing the standard supervised learning baseline with full access to training data. Additionally, we compare our method with an LLM-only baseline that receives only x^{test} and a prompt containing the problem formulation.

3.4. Classification Results

We compare our approach against TabPFN (Hollmann et al., 2022) and XGBoost (Chen & Guestrin, 2016). Despite never accessing the raw data, our method achieves competitive performance in predicting 30-day readmissions, as indicated by the metrics presented in Table 1. Specifically, our Query, Don't Train methodology demonstrates strong precision and recall, underscoring the effectiveness of structured reasoning over aggregate statistics. These results highlight the potential of our approach to provide accurate predictions while utilizing minimal training resources.

Table 1. Performance comparison of different models on 30-day readmission prediction for Type 2 Diabetes patients predicted for a subset of 2,000 patients. Evaluation metrics include Precision, Recall, and F1-score. The errors are represented as standard deviations (±). Query, Don't Train (QDT) refers to using SQL queries to perform predictions without direct access to patient-level data.

Model	Precision	Recall	F1-score
TabPFN	0.63 ± 0.05	**0.76** ± 0.07	0.69 ± 0.06
XGBoost	0.65 ± 0.04	0.68 ± 0.06	0.66 ± 0.05
LLM	0.54 ± 0.03	0.51 ± 0.04	0.52 ± 0.03
QDT	**0.68** ± 0.05	0.73 ± 0.06	**0.70** ± 0.05
QDT (30% less features)	0.65 ± 0.04	0.69 ± 0.05	0.67 ± 0.04
QDT (70% less features)	0.62 ± 0.05	0.65 ± 0.04	0.64 ± 0.04

3.5. Ablation Study on Missing Features

To investigate the impact of feature availability on model performance, we conducted an ablation study by systematically removing features from x^{test}. The findings illustrate that our method maintains robust performance even with reduced feature sets. When 30% of the features were omitted, the performance metrics showed only a modest decrease in the F1-score, dropping to 0.67. This demonstrates that, despite missing features, the agent effectively utilized the remaining features in x^{test} to identify relevant similar examples, which it uses to reason for accurate predictions. When a feature like 'age' is missing, the LLM omits age-based filters and instead generates queries using only available features, enabling robust predictions without imputation. However, with a substantial reduction of 70% of features, the performance was impacted more significantly, resulting in an F1-score of 0.64. These results attempt to solve the challenges posed by incomplete data in real-world EHR scenarios (Yu et al., 2024).

4. Conclusion

This work introduces QDT, a new framework that reimagines structured prediction through symbolic interaction rather than model training. Our findings demonstrate that LLMs can serve as foundation models for structured data without requiring access to raw examples or parameter tuning. By pairing LLM-generated SQL queries with cohort-level aggregation and chain-of-thought reasoning, QDT constructs implicit, task-conditioned table representations entirely at inference time. This paradigm offers a practical and conceptually distinct alternative to pretraining: it scales across tasks with no model updates, provides interpretability through auditable query outputs, and complies with privacy regulations by design.

The approach is particularly suited to high-stakes domains like healthcare, where individual-level data is sensitive and institutional data-sharing is often infeasible. QDT offers clear advantages in deployment flexibility, explainability, and robustness to missing data, as the system dynamically selects what to query based on feature availability. These attributes make it a compelling candidate for real-world clinical decision support under strict data governance. While demonstrated in healthcare, this abstraction readily extends to other structured domains such as finance, education, and public policy.

In sum, QDT represents a step toward a new class of foundation model interfaces for structured data—ones that emphasize reasoning over memorization, and symbolic querying over supervised optimization.

5. Limitations and Future Work

Despite these strengths, several limitations must be addressed. First, the computational efficiency of LLM-driven query generation remains uncertain, particularly as tasks become more complex. Inference time for QDT increases with the number of features, schema complexity, and database size, as each additional element may require extra queries and reasoning. Second, while our experiments focus on structured tabular data, extending this framework to multimodal EHRs (e.g., imaging or unstructured clinical notes) may require further innovations in prompt engineering and query design.

The privacy constraints we implement allow access only to aggregated results for two or more patients. These constraints can be adjusted to enforce stricter censoring policies, and more fine-grained privacy-preserving mechanisms can be incorporated as needed. Our k-anonymity constraint offers limited privacy and is vulnerable to inference attacks; future work will explore combining this method with Differential Privacy for stronger guarantees. The privacy agent also blocks queries on small cohorts, repeated or overlapping queries, and monitors query patterns to prevent differencing attacks and other indirect disclosures.

Another consideration is the potential for adversarial or suboptimal queries generated by LLMs, which pose risks to the reliability of QDT, particularly in healthcare. To enhance reliability, we are developing automated query validation and error detection to mitigate these risks. Ensuring the reliability of the query-generation process in high-stakes environments is crucial for future work. Additionally, although our method has been tested in US hospitals, broader validation across diverse healthcare systems is essential for establishing generalizability. A few-shot LLM baseline with anonymized in-context examples would provide a fairer comparison, and we plan to include this in future evaluations.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Bai, Y., Chen, F., Wang, H., Xiong, C., and Mei, S. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36:57125–57211, 2023.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Chua, L., Ghazi, B., Huang, Y., Kamath, P., Kumar, R., Liu, D., Manurangsi, P., Sinha, A., and Zhang, C. Mind the privacy unit! user-level differential privacy for language model fine-tuning. *arXiv preprint arXiv:2406.14322*, 2024.

Clore, John, C. K. D. J. and Strack, B. Diabetes 130-US Hospitals for Years 1999-2008. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5230J.

Cohen, I. G. and Mello, M. M. Hipaa and protecting health information in the 21st century. *Jama*, 320(3):231–232, 2018.

den Breejen, F., Bae, S., Cha, S., and Yun, S.-Y. Why in-context learning transformers are tabular data classifiers. *arXiv e-prints*, pp. arXiv–2405, 2024.

Gao, D., Wang, H., Li, Y., Sun, X., Qian, Y., Ding, B., and Zhou, J. Text-to-sql empowered by large language models: A benchmark evaluation. *arXiv preprint arXiv:2308.15363*, 2023.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520, 2022.

Hegselmann, S., Buendia, A., Lang, H., Agrawal, M., Jiang, X., and Sontag, D. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, pp. 5549–5581. PMLR, 2023.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L. A., and Mark, R. Mimic-iv. *PhysioNet. Available online at: https://physionet.org/content/mimiciv/1.0/(accessed August 23, 2021)*, pp. 49–55, 2020.

Kim, E., Rubinstein, S. M., Nead, K. T., Wojcieszynski, A. P., Gabriel, P. E., and Warner, J. L. The evolving use of electronic health records (ehr) for research. In *Seminars in radiation oncology*, volume 29, pp. 354–361. Elsevier, 2019.

Kim, M. J., Lefebvre, F., Brison, G., Perez-Lebel, A., and Varoquaux, G. Table foundation models: on knowledge pre-training for tabular learning. *arXiv preprint arXiv:2505.14415*, 2025.

Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., Ibrahim, M., and van der Maaten, L. Crypten: Secure multi-party computation meets machine learning. *Advances in Neural Information Processing Systems*, 34: 4961–4973, 2021.

Kruse, C. S., Smith, B., Vanderlinden, H., and Nealand, A. Security techniques for the electronic health records. *Journal of medical systems*, 41:1–9, 2017.

Kuang, W., Qian, B., Li, Z., Chen, D., Gao, D., Pan, X., Xie, Y., Li, Y., Ding, B., and Zhou, J. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5260–5271, 2024.

Liu, B., Li, X., Zhang, J., Wang, J., He, T., Hong, S., Liu, H., Zhang, S., Song, K., Zhu, K., et al. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990*, 2025.

McElfresh, D., Khandagale, S., Valverde, J., Prasad C, V., Ramakrishnan, G., Goldblum, M., and White, C. When do neural nets outperform boosted trees on tabular data? *Advances in Neural Information Processing Systems*, 36: 76336–76369, 2023.

Meng, C., Trinh, L., Xu, N., Enouen, J., and Liu, Y. Interpretability and fairness evaluation of deep learning models on mimic-iv dataset. *Scientific Reports*, 12(1): 7166, 2022.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Qu, J., Holzmüller, D., Varoquaux, G., and Morvan, M. L. Tabicl: A tabular foundation model for in-context learning on large data. *arXiv preprint arXiv:2502.05564*, 2025.

Tsai, M.-L., Chen, K.-F., and Chen, P.-C. Harnessing electronic health records and artificial intelligence for enhanced cardiovascular risk prediction: A comprehensive review. *Journal of the American Heart Association*, 14 (6):e036946, 2025.

Voigt, P. and Von dem Bussche, A. The eu general data protection regulation (gdpr). *A practical guide, 1st ed., Cham: Springer International Publishing*, 10(3152676): 10–5555, 2017.

Wang, R., Wang, Z., Huang, C., Wang, R., Yu, T., Yao, L., Lui, J., and Zhou, D. Federated in-context learning: Iterative refinement for improved answer quality. *arXiv preprint arXiv:2506.07440*, 2025.

Yu, Z., Chu, X., Jin, Y., Wang, Y., and Zhao, J. Smart: Towards pre-trained missing-aware model for patient health status prediction. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 63986–64009. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/751ef1e7f557a8a88f0837b61bf5070f-Paper-Conference.pdf.

A. Example Agent Run

This section provides an example agent run. The example demonstrates how the LLM produces interpretable outputs. The interpretability comes from the reasoning being shown step-by-step, which helps users understand how the model arrives at its conclusion. Moreover, SQL queries are generated and executed as part of this process, allowing users to trace back the logic and validate intermediate reasoning steps. Finally, the uncertainty of the model's predictions is visible in the final step, where the SQL query results reflect the confidence in the outcome. This ensures that the outputs are explainable and not just black-box results.

Step 1: Prompt

Task: Use a given schema and patient record to predict if the patient is likely to be readmitted. This involves querying the database using SQL for patients with similar feature values and inferring the outcome.

Input Sample:

```
Patient Record:
  ...
  time_in_hospital: 8
  num_lab_procedures: 71
  num_procedures: 6
  num_medications: 37
  max_glu_serum: 1.0
  ...
```

Database Schema:

```
Table: diabetes
Columns:
  ... time_in_hospital, num_lab_procedures, num_procedures,
  num_medications, max_glu_serum, readmitted ...
```

Step 2: Agent Thought

I have the schema of the diabetes table and the patient record features. The key features of the patient record seem to be `time_in_hospital`, `num_lab_procedures`, `num_procedures`, `num_medications`, `max_glu_serum`. To predict if the patient will likely be readmitted, I can find similar patient records in the database with similar feature values and check their readmission status. I will query the diabetes table for records with same values for these key features and infer the likely outcome based on the `readmitted` column.

Step 3: Action

```
SELECT time_in_hospital, num_lab_procedures, num_procedures,
       num_medications, max_glu_serum, readmitted
FROM diabetes
WHERE time_in_hospital = 8.0
  AND num_lab_procedures = 71.0
  AND num_procedures = 6.0
  AND num_medications = 37.0
  AND max_glu_serum = 1.0
LIMIT 10
```

Step 4: Output

```
[]
```

Step 2: Agent Thought

Thought: The query to find exact matches for the patient's key features returned no results. This means there are no exact matches for all these values in the database. I should relax the query to find patients with similar values. For example, patients with `time_in_hospital` close to 8, `num_lab_procedures` close to 71, `num_procedures` close to 6, `num_medications` close to 37, and `max_glu_serum=1`. Then I can check their readmission status to infer the likely outcome for this patient.

Step 3: Action

```
SELECT time_in_hospital, num_lab_procedures, num_procedures,
       num_medications, max_glu_serum, readmitted
FROM diabetes
WHERE time_in_hospital BETWEEN 6 AND 10
  AND num_lab_procedures BETWEEN 60 AND 80
  AND num_procedures BETWEEN 4 AND 8
  AND num_medications BETWEEN 30 AND 40
  AND max_glu_serum = 1.0
LIMIT 10
```

Step 4: Output

```
{data_samples}
```

Step 2: Thought

Now let's check the readmission status of these patients.

Step 3: Action

```
SELECT readmitted, COUNT(*) as count
FROM diabetes
WHERE time_in_hospital BETWEEN 6 AND 10
  AND num_lab_procedures BETWEEN 60 AND 80
  AND num_procedures BETWEEN 4 AND 8
  AND num_medications BETWEEN 30 AND 40
  AND max_glu_serum = 1.0
GROUP BY readmitted
```

Step 4: Output

```
[(0,160),(1,140)]
```

Step 5: Thought

Among patients with similar features to the given patient (`time_in_hospital` 6–10, `num_lab_procedures` 60–80, `num_procedures` 4–8, `num_medications` 30–40, `max_glu_serum=1`), 160 were not readmitted and 140 were readmitted. The majority were not readmitted, so the patient is more likely not to be readmitted.

Final Answer: No

B. Hyperparamters XGBoost

In this study, we utilized the **XGBoost** library (Chen & Guestrin, 2016) in Python to perform 30-day readmission prediction for Type 2 Diabetes patients. To enhance the model's performance, we conducted systematic hyperparameter tuning utilizing the `GridSearchCV` class from the **Scikit-learn** library (Pedregosa et al., 2011).

The following hyperparameters were tuned, along with their respective ranges:

- **Learning Rate (eta)**: Explored in the range of 0.01 to 0.3. The learning rate controls the contribution of each new tree to the overall prediction, where a smaller value generally requires more boosting rounds and allows the model to learn more cautiously, reducing the risk of overshooting optimal parameter values.
- **Max Depth**: Tested values ranged from 3 to 10. This parameter affects the complexity of the individual trees, with deeper trees capable of capturing intricate patterns at the potential cost of increased overfitting.
- **Min Child Weight**: Values were varied from 1 to 10. This parameter sets a minimum threshold for the sum of instance weights required in a child node, thus helping to control overfitting.
- **Subsample**: Evaluated rates of 0.5, 0.7, and 1.0. This parameter determines the fraction of the training data used to grow each tree, with lower values potentially diminishing overfitting through randomization.
- **Colsample_bytree**: Investigated values included 0.3, 0.5, and 0.8, indicating the fraction of features that are sampled for each individual tree.

The hyperparameter tuning process involved performing a grid search combined with **5-fold cross-validation**, which ensured a thorough assessment of model performance across various hyperparameter combinations. The optimal configuration identified through this process was subsequently used to train the final model, facilitating improved predictive accuracy in the 30-day readmission outcomes among patients with Type 2 Diabetes.