Statement of Purpose (SoP)

DSL501: Machine Learning Project

Team Name: Amay Dixit

September 4, 2025

1. Project Details

- **Project Title:** Meta-Learning-Based Adaptive Model Selection for Learned Indexes
- **Code Repo Link (if available):** `https://github.com/amaydixit11/DSL501_ML_Project`
- **If Own Idea:** Yes
  Applying meta-learning techniques to automatically select optimal ML models for different data segments in learned database indexes, addressing the fundamental limitation of current one-size-fits-all approaches.

2. Problem Statement

Database indexing is critical for query performance in modern data systems. Traditional approaches such as **B-Trees** and **Hash Tables** use fixed structures that do not adapt to data patterns.

Recent *learned indexes* replace these structures with ML models that learn data distributions, often achieving better performance for structured data. However, current learned index implementations suffer from a critical limitation, They use a **single model type** across the entire dataset.

Motivation

Real-world data exhibits **heterogeneous patterns** within the same dataset. For example: In a **geospatial database**, downtown areas may have densely clustered coordinates requiring high-precision neural networks. In contrast, **highway corridors** often exhibit near-linear patterns that are best served by simple linear regression. Current approaches force the same model everywhere, leading to suboptimal performance.

Research Question

Our research question: *Can we improve learned index performance by using meta-learning to adaptively select the optimal model type for each data segment based on local statistical characteristics?*

Significance

This problem is significant because it addresses a fundamental inefficiency in how learned systems handle heterogeneous data. The implications extend beyond databases to any domain requiring **adaptive model selection**.

3. Methodology

Our approach introduces a novel meta-learning framework with four core components:

- **Feature Extractor:** Analyzes data segments and extracts statistical features including distribution shape (skewness, kurtosis), variability metrics (variance, coefficient of variation), information content (entropy), and structural properties (gap density, monotonicity). These features characterize the local data patterns that determine optimal model choice.
- **Model Zoo:** A curated repository of candidate models including Linear Regression (optimal for monotonic trends), Polynomial Regression (smooth curves), Decision Trees (irregular patterns with plateaus), and Shallow Neural Networks (complex non-linear patterns). Each model is pre-profiled for performance characteristics across different data pattern types.
- **Meta-Learner:** A Random Forest Classifier trained to map segment features to optimal model selection. The training process involves generating diverse synthetic datasets, segmenting them, testing all candidate models on each segment, and learning the mapping from statistical features to best-performing model.
- **Adaptive Index:** Integrates all components into a unified system that provides per-segment model selection while maintaining a consistent query interface. The system combines ML predictions with traditional structures for accuracy guarantees.

Key differentiators from existing methods: First application of meta-learning to learned indexes, segment-level rather than dataset-level optimization, automated feature-driven selection replacing manual heuristics, and comprehensive statistical characterization of optimal model patterns.

4. Dataset Details

We will use a multi-tiered dataset strategy:

Synthetic Datasets: Generated datasets with controlled patterns including piecewise linear, polynomial growth, step functions, and mixed noise patterns. These provide ground truth for validation and comprehensive coverage of pattern types. Size: 50+ datasets with 100K-1M records each.

Real-world Datasets:

- OpenStreetMap coordinates: Geographic data with natural clustering patterns
- NYC Taxi timestamps: Temporal data with seasonal and daily patterns
- Stack Overflow post IDs: User-generated sequential data with irregular gaps
- Financial time series: Stock prices with volatile and trend patterns

Preprocessing: Data will be segmented using multiple strategies (fixed-size windows, pattern-based boundaries, adaptive segmentation). Statistical features will be computed for each segment. Normalization and scaling will be applied to ensure consistent model training.

This diverse dataset collection ensures our meta-learning approach generalizes across different data types and pattern characteristics commonly found in production databases.

5. Required Resources

Hardware:

- GPU: NVIDIA RTX 4090 or equivalent (24GB VRAM) for neural network training
- CPU: Multi-core processor (16+ cores) for parallel segment processing
- RAM: 64GB for handling large datasets and concurrent model training
- Storage: 2TB SSD for dataset storage and intermediate results

6. Novelty of Approach

This work introduces several novel contributions to the learned index domain:

- **Meta-Learning Application:** First application of meta-learning principles to learned database indexing. While meta-learning has been applied to AutoML and model selection in other domains, its application to database indexing represents a novel intersection of database systems and meta-learning research.
- **Heterogeneous Architecture:** Unlike current approaches that enforce architectural uniformity, our system creates truly heterogeneous indexes where different segments use fundamentally different model types based on data characteristics rather than arbitrary choices.
- **Feature-Driven Selection:** Development of a comprehensive statistical feature set specifically designed for characterizing data segments in the context of learned index optimization. This replaces ad-hoc heuristics with principled, data-driven decisions.
- **Automated Optimization:** Complete automation of the model selection process, eliminating the need for manual tuning and domain expertise in model architecture selection.

Expected improvements: 15–25% reduction in lookup latency compared to static learned indexes, >85% accuracy in model selection decisions, and consistent performance gains across diverse data types. These improvements stem from better alignment between model capabilities and local data characteristics.

7. Team Composition and Individual Contributions

- **Member 1:** Amay Dixit, 12340220, Project Lead – System architecture design, meta-learning implementation, feature engineering, integration testing, and overall project coordination.

8. Expected Outcomes

Performance Targets:

- Achieve 15–25% improvement in lookup latency over static learned indexes
- Demonstrate >85% accuracy in meta-model selection decisions
- Maintain <15% memory overhead compared to single-model approaches
- Show consistent performance gains across 5+ diverse dataset types

Technical Deliverables:

- Complete Python implementation of the meta-learning framework
- Comprehensive benchmarking suite for learned index evaluation
- Feature extraction toolkit for data segment characterization
- Trained meta-models with documented performance characteristics
- Detailed experimental results and analysis

Research Contributions:

- Novel meta-learning methodology for adaptive index optimization
- Statistical characterization framework for data segment analysis
- Comprehensive evaluation of heterogeneous vs. homogeneous learned indexes
- Open-source implementation for future research building

9. References

1. Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. 2018. The Case for Learned Index Structures. *In Proceedings of the 2018 International Conference on Management of Data (SIGMOD '18). Association for Computing Machinery, New York, NY, USA, 489–504.* https://doi.org/10.1145/3183713.3196909

2. Michael Mitzenmacher. 2018. A model for learned bloom filters, and optimizing by sandwiching. *In Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18). Curran Associates Inc., Red Hook, NY, USA, 462–471.*

3. Ryan Marcus, Andreas Kipf, Alexander van Renen, Mihail Stoian, Sanchit Misra, Alfons Kemper, Thomas Neumann, and Tim Kraska. 2020. Benchmarking learned indexes. *Proc. VLDB Endow. 14, 1 (September 2020), 1–13.* https://doi.org/10.14778/3421424.3421425

4. Minguk Choi, Seehwan Yoo, and Jongmoo Choi. 2024. Can Learned Indexes be Built Efficiently? A Deep Dive into Sampling Trade-offs. *Proc. ACM Manag. Data 2, 3, Article 116 (June 2024), 25 pages.* https://doi.org/10.1145/3654919

5. Qin, Jiayong & Zhu, Xianyu & Liu, Qiyu & Zhang, Guangyi & Cai, Zhigang & Liao, Jianwei & Hu, Sha & Peng, Jingshu & Shao, Yingxia & Chen, Lei. (2025). Piecewise Linear Approximation in Learned Index Structures: Theoretical and Empirical Analysis. *arXiv preprint arXiv:2506.20139.*

6. Amarasinghe, K., Choudhury, F., Qi, J. and Bailey, J., 2024. Learned Indexes with Distribution Smoothing via Virtual Points. *arXiv preprint arXiv:2408.06134.*

7. Ali Hadian and Thomas Heinis. 2019. Considerations for handling updates in learned index structures. *In Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management (aiDM '19). Association for Computing Machinery, New York, NY, USA, Article 3, 1–4.* https://doi.org/10.1145/3329859.3329874

8. Ding, J., Minhas, U. F., Zhang, H., Li, Y., Wang, C., Chandramouli, B., Gehrke, J., Kossmann, D., & Lomet, D. B. (2020). ALEX: An Updatable Adaptive Learned Index. *SIGMOD.*

9. Li, C., et al. (2023). Learned Index: A Comprehensive Experimental Evaluation. *VLDB*, 16(8), 1992–2005.

10. Khan, S.A., et al. (2023). A Literature Survey and Empirical Study of Meta-Learning for Classifier Selection. *ScienceDirect.*

11. Gao, H., et al. (2022). CARMI: A Cache-Aware Learned Index with a Cost-based Model Selection Mechanism. *PVLDB*, 15(12), 2679–2692.

12. Yang, J., et al. (2023). Partial Index Tracking: A Meta-Learning Approach. *Proceedings of Machine Learning Research*, v232.