The American Express Campus Challenge 2025

Pritish Saha
Upal Mazumder
Ankit Meda

IIT Kharagpur

Spacebar Sketchers

Summary of the Final Solution

Objective Function / Dependent Variable

Learning-to-Rank (LTR) using a `lambdarank` objective. The goal is to predict the optimal *prioritized list* of offers for each customer, maximizing engagement

The **dependent variable** is the relative ordering of offers within a customer's session, which is learned by the model to produce a ranked list.

Sampling

We have used a **Stratified Group K-Fold** cross-validation technique.

To tackle **data leakage**, we performed an initial **85/15 time-based split**, where the most recent customers were held out for validation. This simulates a real-world scenario of predicting on new, unseen users. Within the training set, we applied **Stratified Group K-Fold**, ensuring all interactions from a single customer (`id2`) stay in the same fold to preserve group integrity.

To address **class imbalance**, the stratification maintained a consistent **click-through rate (y=1)** across all folds. This consistency is crucial for producing stable and reliable **MAP@7 scores**.

Feature Engineering

.**Vectorized Features:** Creation of fast, efficient features like cyclical time-based data (e.g., day of the week) and basic text features from offer descriptions.

Static Offer Profiles: Deriving intrinsic, unchanging features for each offer from metadata, such as offer duration, brand, industry, and channel.

Dynamic Customer Profiles: Building time-aware, leakage-free customer profiles by calculating historical aggregations (e.g., past click-through rates, interaction frequency) using expanding windows to create a true point-in-time snapshot of customer behavior.

Modelling Technique

Primary Model (LightGBM Ranker): A Gradient Boosted Decision Tree (GBDT) model using the LambdaMART algorithm (`lambdarank` objective) serves as the primary engine to generate an initial ranking of offers.

Hyperparameter Tuning: The `Optuna` library is used for automated and efficient Bayesian hyperparameter optimization to find the best-performing configuration for the LightGBM model.

Residual Correction (Transformer Network): A Transformer-based neural network is trained to predict the *residual errors* of the primary LightGBM model. This allows the deep learning model to learn and correct complex, contextual patterns that the GBDT model systematically gets wrong.

Any Other Dimension?

From Ranking Scores to Actionable Probabilities

The final ensemble doesn't just predict rank—it outputs calibrated 0–1 probabilities, enabling direct, threshold-based decisions for acceptance and integration into business logic.

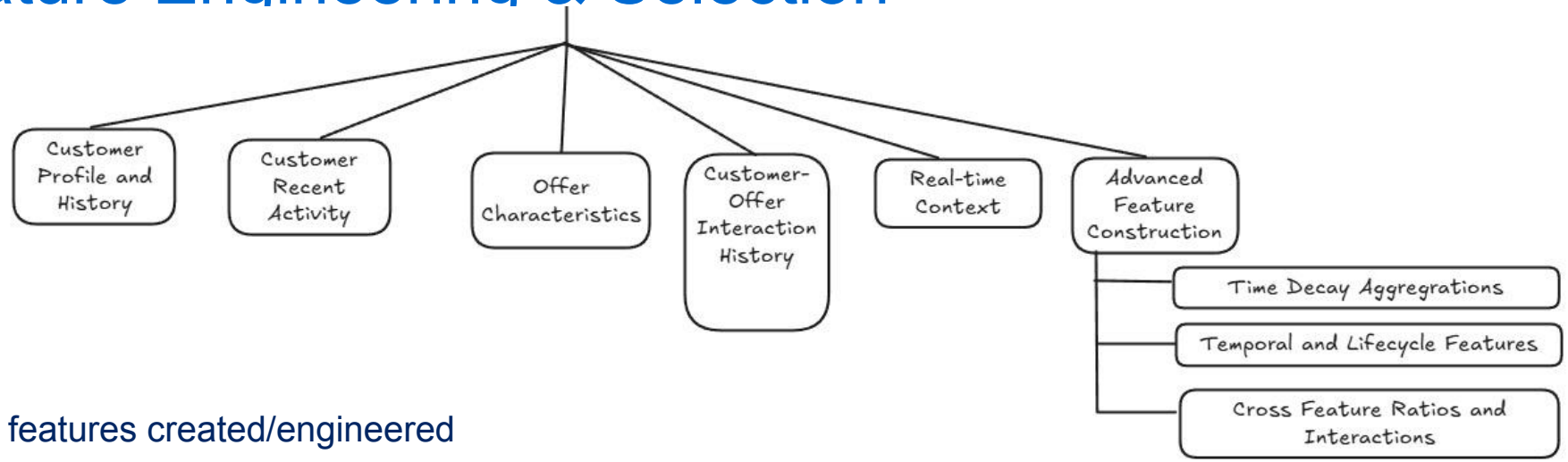
Production-Grade Data Pipeline

Built on a scalable, parallelized framework leveraging 64-core processing, the pipeline ensures efficient handling of large-scale data while preventing leakage through point-in-time joins and customer-aware processing.

Scalable Feature Engineering

Parallelized customer-wise processing using `multiprocessing`, optimized memory via downcasting and incremental merging (`pyarrow`), enabling fast, efficient computation on standard hardware.

Feature Engineering & Selection



✔ Some features created/engineered

S. No.	Feature Description	Examples
1	Point-in-Time (PiT) Customer Transaction History	customer_total_spend, customer_avg_trans_amount, days_since_last_transaction
2	Dynamic Customer Profiles	dynamic_f43_mean (evolving mile balance), dynamic_f77_std (evolving stability of engagement), dynamic_f59_max (evolving max time spent on site)
3	Behavioral Velocity Metrics	balance_velocity, engagement_ratio_velocity, time_spent_velocity
4	Lagged Historical Click-Through Rates (CTR)	customer_ctr_before (overall CTR), customer_brand_ctr_before (brand-specific CTR), time_since_last_click_seconds
5	Personalized Offer Popularity	popularity_vs_customer_norm, offer_historical_ctr_fixed, offer_customer_reach
6	Time-Series Behavioral Stability	f218_stability_5, f219_lumpiness_5, f220_rolling_shannon_entropy

✔ Feature selection used:

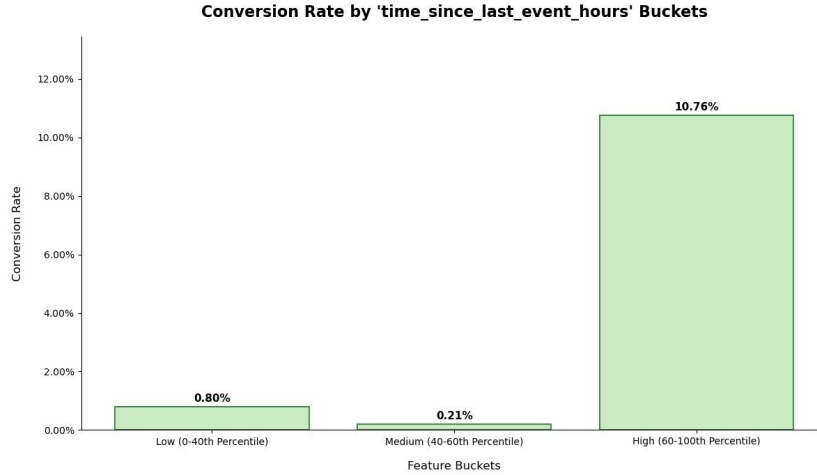
A robust **Permutation Importance** technique was used for feature selection, specifically tailored to the MAP@7 evaluation metric. This process was significantly accelerated using **RAPIDS FIL (Forest Inference Library)** to rapidly calculate the importance of hundreds of features. The top 200 features that caused the largest drop in performance when shuffled were selected for the final models.

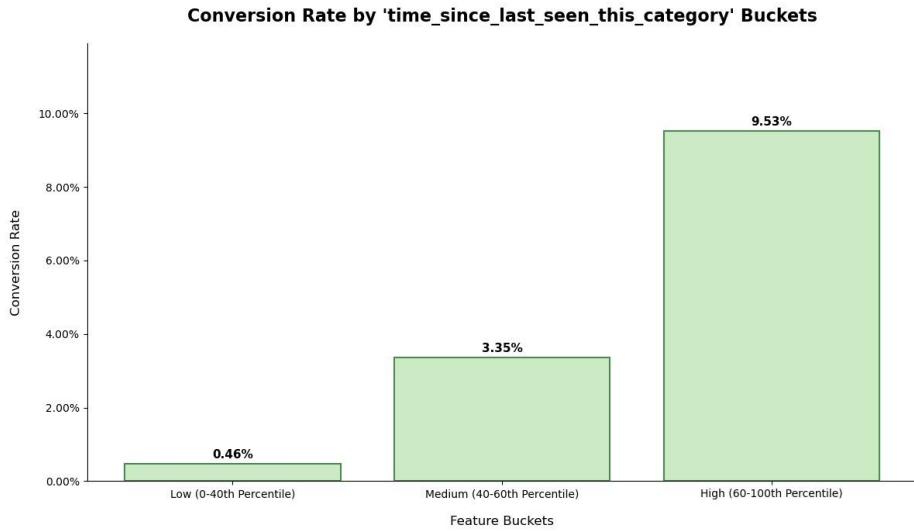
Feature Engineering & Selection

Top 10 Features in the Final Solution

Rank	Feature	Imp
1	time_since_last_event_hours	0.031396
2	f210	0.004055
3	time_since_last_seen_this_category	0.003708
4	f216	0.002679
5	f350_diff1	0.001419
6	f210hma5	0.001203
7	f225	0.001014
8	time_since_session_start_mins	0.000878
9	f366hma5	0.000666
10	f314_diff1	0.000665

Top Engineered feature Trends





Sampling Technique Used

Detailed overview of the Sampling Technique

Stratified Group K-Fold

❖ ***The reason behind the choice of such sampling technique :***

This approach was designed to solve the two biggest challenges in this dataset: data leakage and class imbalance.

Group Integrity: Prevents data leakage by ensuring all events for a single customer (id2) remain in the same fold, stopping the model from seeing a customer's future behavior.

Stratification: Handles class imbalance by maintaining a consistent click rate (y=1) across all folds, which is essential for stable and reliable MAP@7 evaluation.

Time-Based Split: An initial 85/15 split using the most recent customers was performed to simulate a real-world scenario of predicting on new, unseen users.

❖ ***Data Splits by the Numbers***

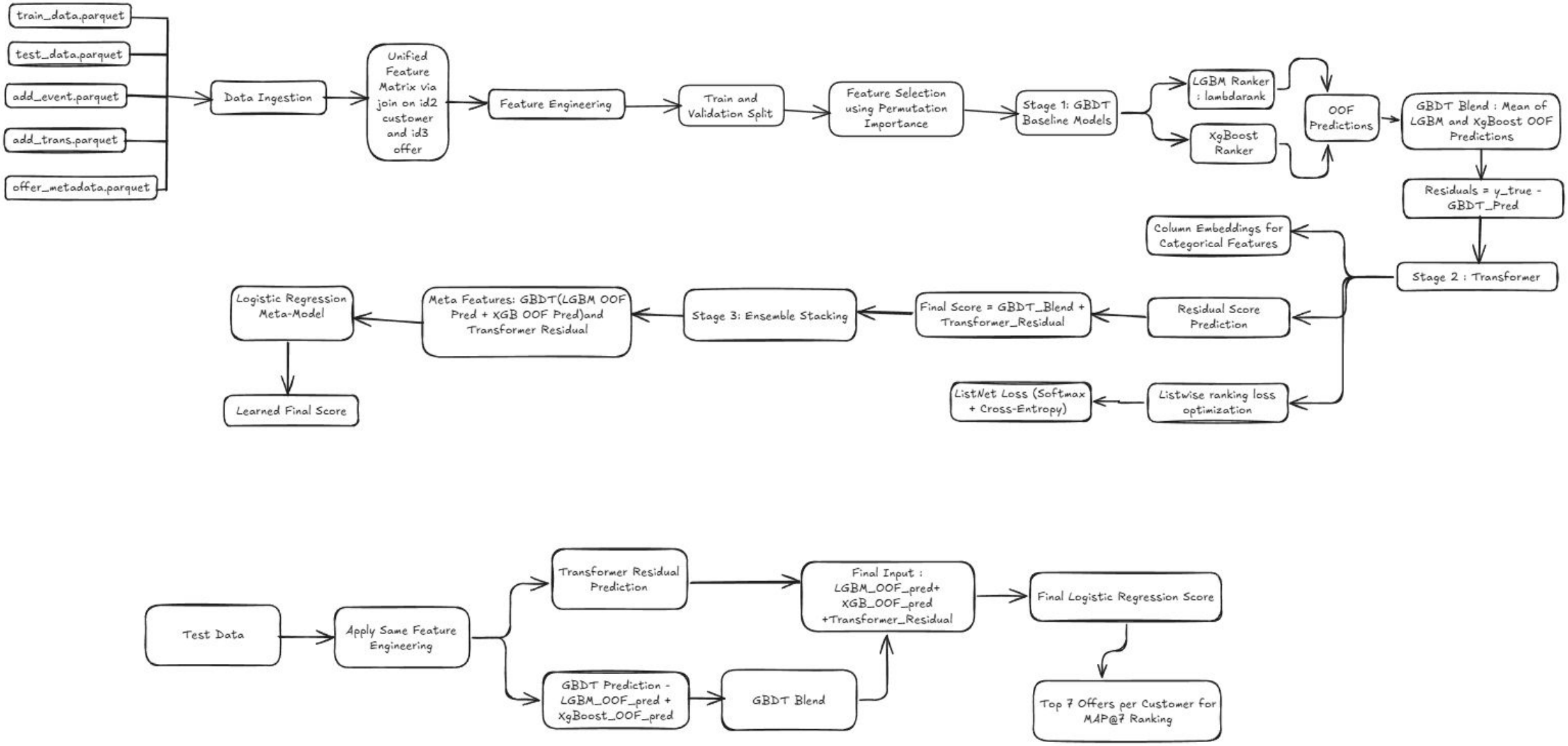
Initial Split Ratio: 85% Training / 15% Validation

CV Number of Folds (k): 5

CV Split Ratio: 80% Training / 20% Validation

Group Integrity: No customer (`id2`) data is ever shared between training and validation sets in any fold.

Model Technique/Algorithm Details



Algorithms and Optimisations Used

- **XGBoost** and **LightGBM (LGBM)** are powerful gradient boosting tools; XGBoost grows trees level-wise for robust accuracy, while LGBM's leaf-wise growth makes it significantly faster and more memory-efficient.

- **Transformer** is a deep learning architecture whose core self-attention mechanism enables it to learn complex, contextual, and highly non-linear relationships between features that are difficult for other models to capture.

- **Out-of-Fold (OOF)** predictions are scores generated for data points from a model that was not trained on them, typically created during a cross-validation process to prevent information leakage.

- **ListNet** is a listwise learning-to-rank algorithm that operates on an entire list of items simultaneously. It works by first converting both the true labels and the model's predicted scores into probability distributions using the Softmax function. The loss is then calculated as the Cross-Entropy between the model's predicted distribution and the "ideal" distribution from the true labels.

- **Bayesian Optimization** is a strategy for finding the maximum of an expensive-to-evaluate "black-box" function. It works by building a probabilistic model (a surrogate) of the objective function, which is then used by an acquisition function to decide the most promising next point to sample.

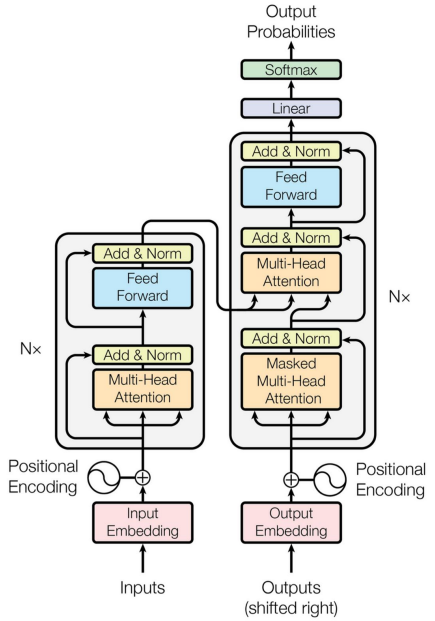


Figure 1: The Transformer - model architecture.

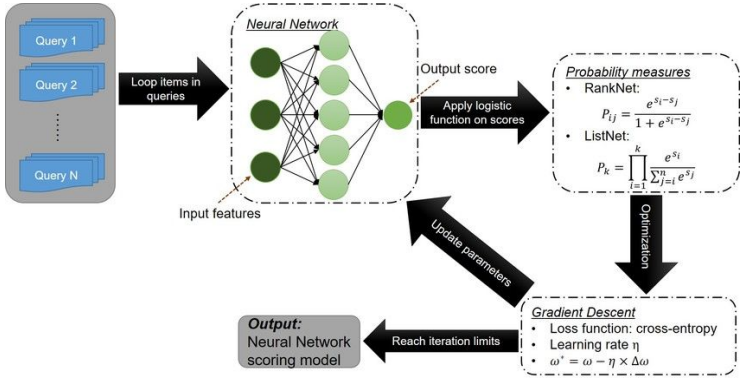


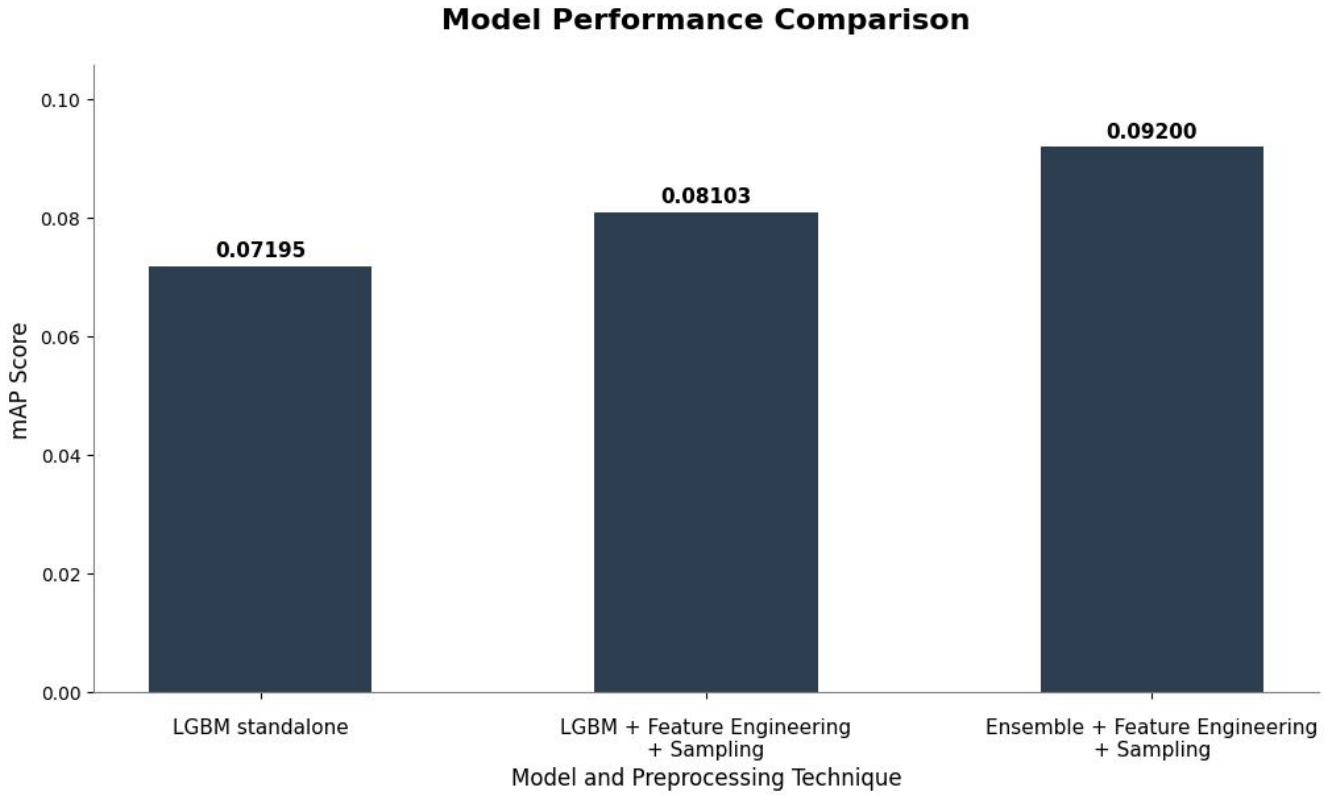
Figure 2: ListNet algo

Model Performance – All Iterations

- ✔ **Baseline Model**: A standalone LightGBM established the initial performance benchmark.
- ✔ **Feature Engineering**: This stage delivered the most significant jump in the MAP@7 score, proving the critical impact of data enrichment.
- ✔ **Final Ensemble**: Ensembling our diverse models (LGBM, XGBoost, Transformer) provided the final incremental lift to achieve our peak performance.

MAP score

[Out of Sample/Validation Results]



*Output from a out of sample/validation data that they created from the training data

More Potential to Improve

1 ML Enhancements

- **Semantic Text Embeddings:** Utilize pre-trained language models (e.g., Sentence-BERT) on offer descriptions to capture semantic meaning, moving beyond simple keyword matching.
- **Graph-Based Network Features:** Model the data as a customer-offer-brand network and use graph embedding algorithms (e.g., Node2Vec) to learn features that represent complex, multi-step relationships.

2 Feature Engineering

- **Two-Stage Re-ranking:** Implement a two-stage architecture: a fast candidate-generation model to select the top ~50 offers, followed by our complex ensemble to re-rank only this smaller set for maximum precision.
- **Model Diversity:** Introduce a CatBoost or TabNet model into the final ensemble to leverage different approaches to handling categorical data and tabular structures, further improving robustness.

3 Any other dimension?

- **Automated Ensemble Weighting:** Systematically optimize the final blending weights of the LGBM, XGBoost, and Transformer models by training a meta-learner on the out-of-fold predictions to maximize the MAP@7 score.
- **Prediction Calibration:** Apply post-processing techniques like Isotonic Regression to the final prediction scores, ensuring the relative ranking is as accurate as possible before submission.