Amex Offer Personalization: A Case Study in End-to-End Machine Learning

Author: Lovedeep Sharma

Executive Summary

This report details the end-to-end development of a machine learning-based recommendation engine designed to personalize American Express offers. The primary business objective was to accurately rank the top seven most relevant offers for each customer, with performance measured by the Mean Average Precision at 7 (MAP@7) metric.

The project began with a multi-gigabyte dataset characterized by significant data quality and environmental challenges. Initial exploration revealed corrupted data, sparse metadata, and severe software dependency conflicts within the cloud-based notebook environment. These real-world hurdles necessitated a strategic pivot from a monolithic script to a robust, modular four-notebook workflow, isolating data preparation from model training and analysis.

The core of the solution was advanced behavioral feature engineering, where millions of transaction and event logs were aggregated to create rich, predictive customer profiles. A comprehensive benchmark was then conducted, comparing a Decision Tree baseline, two state-of-the-art gradient boosting models (LightGBM, CatBoost), and a deep learning Keras Neural Network.

The final results demonstrated a clear victor: the **LightGBM model achieved the highest MAP@7 score of 0.03907**, proving to be the most effective tool for this specific tabular ranking task. The project not only yielded a high-performance model but also provided a key strategic insight: a customer's historical behavior is the single most powerful predictor of their future offer acceptance.

1. Problem Definition & Strategic Objective

The American Express Campus Challenge 2025 presented a classic but complex personalization problem. The goal was to move beyond simple classification and develop a system capable of creating a ranked list of recommended offers for each customer. The official evaluation metric, **MAP@7**, confirmed that the core task was one of **learning-to-rank**, where the precise order of the top recommendations is critically important.

The project was therefore defined by a single objective: **To build and evaluate a suite of machine learning models to identify the most effective architecture for maximizing the MAP@7 score in a real-world customer recommendation scenario.**

2. The First Battle: Initial Data Assessment & The Unseen Enemy

The project commenced with four large-scale datasets provided in Parquet format, totaling over 2GB. An initial exploratory data analysis (EDA) immediately revealed the true nature of the challenge. The data was not clean; it was a realistic and hostile environment.

Challenge 1: The "Object" Deception & Memory Overload

The first major discovery was that nearly every column across all datasets was loaded with the generic object data type. This had two severe consequences:

- **Computational Inefficiency:** Numerical and categorical data stored as strings prevented any mathematical operations.
- **Memory Bloat:** The initial memory footprint of the train_df DataFrame alone exceeded 2.1 GB, pushing the limits of the compute instance and threatening the viability of the project.

Challenge 2: The Data Wasteland

A comprehensive missing value analysis uncovered a data wasteland. Several columns were found to be 100% null (e.g., f112, f136), representing "dead features" that contained no information. Many others had over 98% missing values, making them more noise than signal.

This initial reconnaissance phase was critical. It proved that a simple "load and train" approach would be impossible. A rigorous, multi-step data preparation strategy was not just recommended; it was mandatory.

3. A Strategic Retreat: The Pivot to a Modular Workflow

As the complexity of the data and the fragility of the notebook environment became apparent, a critical strategic decision was made. Continuing with a single, monolithic notebook was a high-risk path, prone to catastrophic failure. A single error during a multi-hour model training run could corrupt the memory state, forcing a complete restart of the entire process.

To win this battle, we needed to change the battlefield. The project was restructured into a **professional, four-notebook modular workflow**:

1. **1_data_preparation.ipynb:** The "Factory." Its sole responsibility was to perform the slow, heavy, and error-prone task of turning raw data into clean, model-ready artifacts.
2. **2_traditional_ml_benchmark.ipynb:** The first "Assembly Line." A fast, lightweight environment for benchmarking classical and gradient boosting models.
3. **3_deep_learning_flex.ipynb:** A specialized, isolated "Clean Room" for the resource-intensive deep learning experiments, protecting other modules from its

complex dependencies.

4. **4_final_analysis.ipynb:** The "Boardroom." A simple notebook for loading the results from the assembly lines and presenting the final conclusions.

This modular architecture was the single most important strategic decision of the project. It provided **resilience, isolation, and efficiency**, allowing for rapid experimentation in the modeling phases without ever needing to re-run the time-consuming data preparation.

4. The Gauntlet: A War on Data & Dependencies

With a new strategy in place, the project moved into execution, where a series of formidable challenges were met and overcome.

4.1. The Data Cleaning Campaign (Notebook 1)

The first notebook was dedicated to winning the war on the data itself.

- **Victory:** A comprehensive type-optimization function was developed, intelligently converting object columns to the most memory-efficient numeric and categorical types. This single action **slashed the memory footprint of the data by over 90%**, making the entire project feasible.
- **Victory:** All features with more than 90% missing values were systematically eliminated.
- **Key Contribution:** The core of the project's success was born here. A powerful set of **behavioral features** were engineered by aggregating millions of rows from the add_trans.csv and add_event.csv files. Customer profiles were built summarizing their transaction counts, amounts, and platform interaction ratios.
- **The "Save Point":** The notebook concluded by saving the final, clean X, y, and temp_df DataFrames to the high-speed .feather format, creating the stable foundation upon which all future work would be built.

4.2. The Environment Hell (Notebooks 2 & 3)

As modeling began, the project faced its most frustrating and realistic challenge: the instability of the pre-configured cloud environment. This became a multi-front war on dependencies.

- **First Skirmish (ModuleNotFoundError):** The attempt to use advanced libraries like lightgbm and catboost immediately failed. The solution was tactical: using the %pip install magic command to add the necessary tools to the environment, followed by a kernel restart.
- **The Long Wait (ConvergenceWarning):** An attempt to use a standard Logistic Regression as a baseline resulted in the model "getting stuck" for over 1.5 hours, throwing ConvergenceWarnings. This was not an error, but a crucial **finding**: the 340+ dimensional feature space was too complex for a simple linear model. The strategic decision was made to **replace this problematic baseline with a much faster Decision Tree**, which provided a reliable benchmark in seconds.

- **The Final Boss (ValueError: numpy.dtype size changed):** The attempt to install tensorflow for the deep learning module triggered a catastrophic environment failure. This infamous "binary incompatibility" error proved that the pre-installed versions of pandas and numpy were in direct conflict with TensorFlow's requirements. This was the ultimate test.
- **The Ultimate Victory:** The battle was won by abandoning the default environment entirely. A final, bulletproof workflow was established where **each notebook's first cell explicitly installs a known-stable list of version-pinned libraries**. This created a self-contained, reproducible environment for each module, permanently solving all dependency conflicts. This was a rite of passage, transitioning the project from simple scripting to professional-grade environment management.

5. The Grand Tournament: Benchmarking the Champions

With a stable environment and clean data, the final tournament of models could begin.

5.1. The Competitors

- **The Baseline (Decision Tree):** A fast, simple model to set the minimum performance bar.
- **The Speed Champion (LightGBM):** A state-of-the-art gradient boosting model known for its exceptional performance on tabular data.
- **The Accuracy Specialist (CatBoost):** Another top-tier gradient boosting model, often a close competitor to LightGBM.
- **The Deep Learning Flex (Keras DNN):** A Multi-Layer Perceptron to test the capabilities of deep learning on this structured data problem.

5.2. The Arena

All models were evaluated using a rigorous **5-Fold Stratified Cross-Validation** process. Stratification was essential due to the severe class imbalance (~5% positive class) in the target variable, ensuring that every fold was a fair and representative test.

6. The Verdict: Results & Final Analysis

After all models completed their runs, the final analysis notebook loaded the saved results to generate the master leaderboard.

FINAL COMBINED LEADERBOARD

Model	Mean AUC	Mean MAP@7
LightGBM	0.95320	**0.03907**

CatBoost	0.94709	0.03874
Keras NN	0.94465	0.03741
Decision Tree	0.91583	0.03112

The Champion is LightGBM.

The results tell a clear and compelling story.

1. **Gradient Boosting is King:** Both LightGBM and CatBoost dramatically outperformed the baseline and the neural network on the primary MAP@7 metric. This provides strong evidence that for this type of tabular ranking problem, tree-based ensemble models are the superior architecture.
2. **Deep Learning's Nuance:** The Keras DNN achieved an excellent AUC score, proving it was highly capable of distinguishing between positive and negative classes in general. However, its slightly lower MAP@7 score indicates it was less effective at the fine-grained task of *ranking* the top 7 predictions, a common and insightful finding when comparing these architectures.

7. Conclusion: The Spoils of Victory

This project was a successful, end-to-end journey through the challenges of a real-world machine learning problem. It progressed from a chaotic data landscape to a highly organized, modular, and reproducible workflow, culminating in the identification of a champion model.

The key takeaway is unequivocal: **a LightGBM model, powered by meticulously engineered behavioral features, is the optimal solution for personalizing Amex offers under these conditions.** The project not only provides a high-performance predictive asset but also a battle-tested blueprint for tackling complex data science challenges in the future.