XGBoost Based Recommendation Model (Github)

Summary

This document outlines the development and implementation of a recommendation system for phone plans, powered by XGBoost. The system predicts customer preferences based on input features, incorporates user feedback dynamically, and evolves to improve recommendations.

Key highlights include:

- Preprocessing synthetic customer data for training.
- Training the XGBoost classifier on multi-class classification.
- Real-time plan recommendations with detailed outputs.
- Continuous model improvement via a dynamic feedback loop.

What is XGBoost?

XGBoost (Extreme Gradient Boosting): It is a machine learning library optimized for speed and performance. It is based on the gradient boosting framework and is widely used for structured data tasks.

How does it work?

XGBoost builds an ensemble of decision trees in a sequential manner, where each tree aims to correct the errors of the previous one. This process optimizes an objective function using gradient descent. The combination of multiple weak learners (decision trees) results in a strong predictive model.

Why choose XGBoost?

With built-in regularization, XGBoost is robust to overfitting and provides insights into which features are most impactful, helping refine the model

Data Preparation

The synthetic dataset mimics real-world customer behavior and preferences. It allows us to explore a wide range of scenarios without needing access to sensitive or proprietary customer data. The dataset contains:

- age
- income
- data_usage
- budget
- international_calls
- high_speed_internet
- plan_id (Target Variable)

age	income	data_usage	budget	international_calls	high_speed_internet	plan_id
56	1458	8	67	0	0	4
69	3344	10	84	0	1	4
46	1522	17	113	0	1	4
32	721	12	44	0	0	3

Feature Engineering

Feature engineering enhances the dataset by creating meaningful and predictive variables that improve the model's performance

1. **Numeric Features:**
   - Normalized income, data_usage, and budget using MinMaxScaler.
   - Derived cost_per_gb and usage_to_budget_ratio for better insights.
2. **Categorical Features:**
   - Categorized age into "Youth", "Adult", "Middle-aged", and "Senior".
   - Applied one-hot encoding to age_group.

Feature Importances

A built-in mechanism for evaluating the importance of input features in the decision-making process. By analyzing feature importances, we can identify which features (e.g., income, data usage) have the most significant impact on the recommendations.

Model Training

Adjusting the Target Label: Making the plan_id zero-indexed simplifies multi-class classification and aligns with the requirements of the XGBoost library.

Train-Test Split: Separating data into training and testing sets evaluates how well the model generalizes to unseen data.

Training the Model: Training the **XGBoost classifier**

Recommendation Process

Once trained, the model predicts the best-fit plan based on input features.

The process:

- Uses customer input data (e.g., income, data usage) mapped to numerical values.
- Predicts probabilities for each plan.
- Selects the plan with the highest probability, ensuring the most relevant recommendation.

Feedback Loop

Buffer-Based Feedback: Storing accept/reject feedback in a buffer avoids frequent retraining while retaining valuable feedback. A buffer size of 100 entries balances efficiency and adaptability, ensuring enough data is collected to improve the model without excessive retraining overhead.

Threshold-Based Retraining: Retraining after every 100 feedback entries ensures the model incorporates fresh data in batches, balancing efficiency and adaptability.