RECOMMENDATION SYSTEM PROJECT

Introduction

Recommendation systems are essential for delivering personalized user experiences across a variety of platforms, including e-commerce, streaming services, social media, and news websites. By utilizing historical and user-specific data, these systems predict the most relevant products, services, or content, thereby enhancing user engagement, satisfaction, and conversion rates.

Business Understanding

As businesses and platforms continue to depend on personalized recommendations, they face the challenge of ensuring that their systems deliver accurate, diverse, and relevant suggestions, all while managing large volumes of data and maintaining real-time performance.

This project aims to develop a recommendation system that leverages historical user data to provide tailored recommendations across different domains, such as product recommendations, content suggestions, and service optimization

Business objectives

- Predict item properties relevant to add-to-cart events using a user's viewing history.
- Recommend items a user is likely to add to cart (content-based + behavioral similarity).
- Detect abnormal users to improve model accuracy and remove noise.
- Identify top-selling items & categories.
- Understand user browsing-to-purchase conversion flow.
- Measure the time gap between a user viewing and adding items to the cart.
- Analyze seasonal/hourly trends in user interactions.

Business Analytical Questions

1. Which products/content are most frequently interacted with by users?
2. Which user segments have similar purchase/viewing behaviors?
3. What products/content have high engagement but low recommendation exposure?
4. How do seasonal trends affect product/content interactions?
5. What time of the day does purchase activity increases?
6. What are the most frequent item id predicted?
7. What is the percentage of visitors who made purchase?
8. What are the total transactions over time?

Data Understanding

The analysis utilizes three datasets:

- events.csv: Contains user interaction events with timestamps, visitor IDs, event types (view, addtocart, transaction), item IDs, and transaction IDs (for transaction events).
- item_properties.csv: Contains item properties with timestamps, item IDs, property names, and property values.
- category_tree.csv: which describes category tree

Key steps:

- Load events, item_properties, category_tree.
- Inspect shape, missing values, and distribution of event types.
- Map timestamps to human-readable dates.

Count:

- unique users
- unique items
- interactions per event type

Join item_properties with category_tree to understand hierarchy.

Visualizations:

- Top categories by views/add-to-cart/transactions.
- Heatmap of interactions by hour of day.
- Conversion funnel (views → add-to-cart → purchase).

Data Preparation

For Task 1 (Recommendation):

- Filter relevant users/events for training (views before add-to-cart).
- Merge events with latest known item_properties.
- Feature engineering:
    1. Category counts per user
    2. Average viewed price
    3. Vendors viewed
- Prepare train/test split by user (avoid leakage).

For Task 2 (Anomaly Detection):

- Create per-user features:
    1. Total views/add-to-cart ratio
    2. Items viewed per minute
    3. Categories diversity
    4. Session length distribution
- Label potential anomalies:
    1. High frequency clicks
    2. Extreme ratios (views without add-to-cart)
- Scale & normalize features for modeling.

Modeling

Anomaly Detection

- Feature Engineering: Features are created from user events to characterize behavior (e.g., number of events, time spent, items viewed, items added to cart, transactions).
- Model Selection: An Isolation Forest model is chosen for detecting abnormal users.
- Model Training: The Isolation Forest model is trained on the engineered user features.
- Abnormal User Identification: The trained model is used to predict anomaly labels (-1 for outliers) for each user.
- Analysis: The characteristics of abnormal users are analyzed and compared to normal users to understand the patterns flagged as anomalous.
- Abnormal User Removal: Abnormal users are removed from the dataset for subsequent analysis and model building.

Item Property Prediction

- Feature Engineering: Features are created from 'view' events to predict properties in 'addtocart' events, such as the number of viewed items, total views, and unique viewed properties per visitor.
- Data Splitting: The data is split into training and testing sets for model development and evaluation.
- Model Selection: Classification models like RandomForest, LightGBM, TNN (TensorFlow Neural Network), and XGBoost are explored for predicting item properties.
- Model Training: The selected models are trained on the training data.
- Model Evaluation: Model performance is evaluated using metrics such as accuracy, precision, recall, and F1-score.
- Prediction: The best-performing model is used to predict item properties for the test set.

Recommendation System Exploration

- Content-Based Filtering (CBF): Item profiles are built from properties, and cosine similarity is used to recommend items similar to a given item.
- Collaborative Filtering (CF): A user-item matrix is created, and user-user similarity (cosine similarity) is used to recommend items based on the preferences of similar users.
- Hybrid Recommendations: A simple hybrid approach is explored, combining scores from CF and CBF.
- Category-Based Recommendations: Recommendations are generated based on the category hierarchy, suggesting items within the same or related categories.

Evaluation

Evaluation Results and How They Inform Recommendations

The evaluation of the predictive models provides insights that can inform the recommendation system:

- **Item Property Prediction**: The classification models (RandomForest, LightGBM, TNN, XGBoost) were evaluated based on their ability to predict item properties. The performance metrics (accuracy, precision, recall, F1-score) indicate how well the models can infer properties from viewing behavior. A higher F1-score, as observed with the RandomForest model, suggests a better balance between precision and recall, which is important for recommending items with accurately predicted properties. While the accuracy might seem high, it's crucial to consider the class distribution of properties, as some properties might be more frequent than others. The weighted average metrics provide a better picture of performance across all property classes. These results suggest that viewing behavior does provide some signal for predicting item properties, which can be used to enhance content-based recommendations or provide implicit property information when explicit data is missing.
- **Anomaly Detection**: The Isolation Forest model's evaluation is primarily based on the analysis of the characteristics of the identified abnormal users compared to normal users. The distinct behavioral patterns observed in abnormal users validate the model's ability to flag unusual activity. This informs the recommendation system by allowing for tailored experiences for these users, potentially by reducing or altering the recommendations shown to mitigate potential malicious activity or simply provide a different type of content.

The evaluation of the recommendation algorithms themselves (CBF, CF, Hybrid, Category-based) is demonstrated through examples, showing the types of recommendations generated by each method. A more rigorous evaluation would involve offline metrics (e.g., precision@k, recall@k, diversity) and online A/B testing in a live environment to measure their impact on user engagement and conversion. The current examples serve to illustrate the different approaches and how they can be combined.