**Ecommerce Recommendation Engine**

**Team 26**

**Notebook Objective**

This notebook performs extensive data cleaning, feature engineering, and exploratory analysis on an **Amazon product review dataset**, preparing it for downstream machine learning or data analytics tasks. The process includes deduplication, missing value handling, feature generation, and user-product interaction pattern analysis.

**Data Overview**

* **Dataset size:** Over 568,000 rows, 17 columns
* **Entities covered:** Product IDs, User IDs, profile information, review scores, product categories, review text, and review metadata
* **Memory usage:** Approx. 65.1 MB after initial loading
* Matrix Sparsity**: 100% (typical for ecommerce)**

**Data Preparation and Cleaning**

* Used df.info, df.shape, and df.describe to explore the dataset structure and null values.
* Most columns had no missing values; only "ProfileName" and "Summary" contained a small number of missing entries.
* Applied robust duplicate removal for both product and user review records using df.drop\_duplicates and ensured no missing rows post-cleaning.
* Consolidated and converted time-related columns, extracting year, month, and day from review timestamps and dropping redundant columns to streamline the working dataframe.

**Data preprocessing**

* **Deduplication**: No duplicate rows remain
* **Missing Value Handling**: All missing profile names and summaries resolved
* **Type Casting**: All data columns correctly typed (dates, numerics, categories)
* **Outlier Handling**: Outliers analyzed and retained to preserve real user behavior

**Feature Engineering**

* **temporal Features**: Extracted year, month, day, dayofweek from timestamps
* **Helpfulness Features**:
  + helpfulness\_ratio = HelpfulnessNumerator / HelpfulnessDenominator
  + time\_weight = 1 / (1 + (year - df["year"]))
  + final\_score series: utilized for scoring strategies (multiplicative, logarithmic, and root transforms)
* **Category Assignment**: Automatic product naming and realistic taxonomy
* **Recency**: Feature encoding the recency of reviews to improve recommendations
* **User/Product Deviations**: Ratings compared to user and product averages for more refined personalization

**Exploratory Data Analysis and Visualization**

* **Growth Patterns**: Reviews grow year-over-year
* **Skew Analysis**: 60% ratings = 5, indicating positive skew
* **User/Product Outliers**: 4 users with unusually high review counts; one product stands out with exceptional review volume
* **Weekday Effects**: Ratings dip mid-week, rise at week's end
* **Yearly Ratings**: 1999-2000 were best; 2001 the worst; post-2012, ratings stabilize but fluctuate
* **User-Category Matrix**: Cross-category purchase analysis and visualization
* **Outliers Visualized with Boxplots**: Rationale for retaining anomalies in Helpfulness statistics

**Key Results and Outputs**

| **Step** | **Output Summary** | **Notes** |
| --- | --- | --- |
| **Initial Row Count** | **568,454** | **All product review records before cleaning** |
| **Post-Cleaning Row Count** | **568,401** | **Most duplicates removed, negligible null profile entries** |
| **Product Uniqueness** | **74,257 unique Product IDs** | **Supports robust aggregation at the product level** |
| **Review Helpfulness Flags** | **420,056 normal, 148,345 suspicious reviews** | **Enables fraud analysis or robust model training** |
| **User Rating Patterns** | **Interaction matrices per user & product categories** | **Readiness for recommender or clustering models** |

**Interpretation and Data Integrity Considerations**

* Review outliers are maintained as their removal could bias any ML models trained on this data, due to many products having naturally high review/interactions.
* Product and user interaction insights provide a strong foundation for both classical ML models and modern deep learning recommenders, especially those employing user-product matrices methods.
* All transformations are kept reproducible, with backup copies where relevant for pipeline reversibility.

**Interpretation of the Visual Dashboard**

* **Recommendation Mode Selector**  
  The system allows switching between "User ID" and "Product Search" recommendation modes, making it adaptable to different user actions and supporting both personalized and search-driven recommendations. This flexibility is key for robust real-world use cases.
* **Filter Controls**  
  The sidebar filters, such as minimum rating (with a slider) and maximum discounted price, enhance user experience by enabling fine-grained control over displayed results. For production, this ensures that users can efficiently narrow down recommendations to fit their needs.
* **Product Cards**  
  Each recommended product is shown with an image, title, pricing details (discounted and actual prices, as well as percentage off), rating, and review count. This presentation method closely matches modern e-commerce standards, supporting trust and user engagement by surfacing comprehensive and actionable information.

**Collaborative Filtering Models**

* **Algorithm**: Matrix Factorization SVD (best after hyperparameter tuning)
* **Best Params**:  
  {'n\_factors': 100, 'n\_epochs': 60, 'lr\_all': 0.0155, 'reg\_all': 0.04}
* **Training Data**: Uses (UserID, ProductID, final\_score2)
* **Results**:  
  RMSE=0.9322RMSE=0.9322  
  MAE=0.8112MAE=0.8112
* **Additional Benchmarks**: SVD++, NMF, KNNBasic models compared and validated

**Content-Based Models**

**A. Text/Summary Model**

* **Combination:**Text + Summary → text\_all column
* **TF-IDF:** Feature extraction and Nearest Neighbors search for product similarity
* **Workflow:**
  + User supplies UserID & ProductID
  + Compute α based on user activity
  + System finds content-similar products
  + SVD scores blended with text similarity for final ranking

**B. Category-Based Model**

* **Column Added:** Category
* **Workflow:**
  + Input: UserID + ProductID for category extraction, or UserID + manual category input
  + Candidate pool: Products from the chosen category, excluding products the user already rated
  + Filtering: Products must meet a minimum rating threshold
  + SVD predicts ratings for candidates, shown in interactive UI

**Hybrid Models**

* **Team 1:**Text + Summary + Collaborative Filtering (SVD) using weighted hybrid approach (α*α* based on user activity)
* **Team 2:**Category + Collaborative Filtering using weighted approach (α*α* based on user review count)
* **Model Fusion:**Weighted blend of content-based and collaborative scores for each user/item
* **Robustness Features:**Temporal recency, outlier handling, and fraud/suspicion flags add stability in prediction

**Model Evaluation and Serialization**

* **Quantitative Metrics**: RMSE, MAE tracked and model selection justified by benchmarks
* **Reproducibility:**All models and trained datasets are serialized and saved for workflow automation and deployment
* **Scalability:**Dataset/output formats support retraining, inference, API integration, and live dashboard usage

**Deployment**

**Reliability in Real-World Deployment**

* **Scalability and Sparsity Handling**  
  The dataset is highly sparse (100% matrix sparsity is reported), a common trait in real-world recommendation tasks. SVD- and category-based algorithms are chosen because they scale well and remain robust as the user-item matrix grows.
* **User Trust and Transparency**  
  Each recommendation card shows discounts, actual prices, reviews, and ratings, making results explainable and actionable for users. Well-designed visuals like these build trust, which is essential for e-commerce deployments.
* **Evaluation and Suitability**  
  The notebook includes quantitative metrics like RMSE and MAE for all main models, with iterative model comparison to select optimal parameters for deployment. The models demonstrated good accuracy on test splits, supporting confidence in their predictions.
* **Robustness Features**
  + Time-based features and recency aggregations help keep recommendations relevant as products and user behaviors evolve.
  + Outlier handling and review fraud detection (using engineered flags) are in place to catch abnormal patterns and enhance recommendation reliability.
* **Interactive Dashboard**: Shows real-time recommendations, filter controls, and product details
* **Chatbot**: AI-powered assistant supports users in getting product suggestions and explanations
* **Usage Example**:  
  Templates/scripts included for inference and production deployment
* **Transparency & Trust**:  
  Recommendations displayed with stats and explanations, supporting user confidence