CRISP DM-TECHNICAL PRESENTATION

Project: MovieLens Recommendation System (CineStream)

Team: Pandas and Chill (GROUP 2)

Methodology: CRISP-DM (Cross-Industry Standard Process for Data Mining)

Step 1: Business Understanding

Introduction

- **Overview/Background:**
  The project focuses on "CineStream," a movie streaming platform that currently hosts a library of over 9,000 movies. With 100,836 ratings provided by 610 users, the platform seeks to leverage this data to enhance user experience. In the current digital landscape, users often feel overwhelmed by the sheer volume of available content, leading to decision paralysis.
- **Challenges:**
  The primary operational challenge is high user churn and low engagement due to inefficient content discovery. Users currently spend 20+ minutes browsing before selecting a movie, which negatively impacts retention. The business is trying to solve the "choice overload" problem, where the vast catalogue reduces satisfaction rather than increasing it.
- **Proposed Solution:**
  The proposed solution is to develop a **Collaborative Filtering Recommendation System**. This engine will create a user-item matrix to detect patterns in user behaviour and recommend a personalised list of "Top-5 Movies" to each customer based on their historical ratings and the preferences of similar users. The strategy involves using these recommendations to drive a 30% increase in monthly watch time.
- **Brief Conclusion:**
  The envisioned outcome is a deployed recommendation engine that reduces browsing time to under 5 minutes and reduces user churn by 25% through hyper-relevant content suggestions.

Problem Statement:

The business problem is the lack of personalisation in the current content delivery system. Users are presented with thousands of options without guidance, leading to excessive search times and disengagement. The goal is to utilise historical rating data to predict user preference for unseen movies and surface relevant content automatically.

Objectives:

i. **Mine the sales/rating data:** Extract and process 100k+ interactions from the MovieLens dataset.

ii. **Analyse customer transaction data:** Perform Exploratory Data Analysis (EDA) to understand rating distributions, sparsity, and user tendencies.

iii. **Create predictive models:** Build a Collaborative Filtering model (using scikit-surprise) to project customer purchase/viewing behaviour.

iv. **Make recommendations and conclusions:** Generate top-N movie lists for users and evaluate model performance using RMSE metrics.

Step 2: Data Understanding

- **Load Data:**
  - Data was acquired from the GroupLens Research Lab (MovieLens "Small" dataset).
  - Loaded three key CSV files: ratings.csv, movies.csv, and tags.csv using Pandas.
- **Inspect Data:**
  - **Shape:** The dataset consists of **100,836 ratings**, **610 unique users**, and **9,724 unique movies**.
  - **Info:** Checked data types (Int64 for IDs, Float64 for ratings) and structure.
  - **Descriptive Stats:** The average movie rating is approximately **3.50**, with a standard deviation of 1.04. The rating scale ranges from 0.5 to 5.0 stars.
- **Data Characteristics Checked:**
  - **Uniformity:** Verified rating scales (0.5 to 5.0).
  - **Sparsity:** Calculated the User-Item matrix sparsity, found to be **98.3%** (meaning users rate only ~1.7% of available movies). This was identified as normal and healthy for recommendation datasets.

Step 3: Data Preparation

- **Data Cleaning:**
  - Checked for **Missing Values:** df.isnull().sum() confirmed zero missing values in the ratings and movies datasets.
  - Checked for **Duplicates:** .duplicated().sum() confirmed zero duplicate entries.
- **Feature Engineering & Formatting:**
  - **Matrix Construction:** Prepared the data for the Surprise library, which requires specific formatting of user-item interactions (User ID, Movie ID, Rating).
  - **Merged Data:** Merged ratings with movies to associate numeric movieIds with actual movie titles for human-readable EDA and final recommendations.
- **Exploratory Data Analysis (EDA) & Visualisation:**
  - **Univariate Analysis (Rating Distribution):**
    - Plotted a **Histogram** and **Pie Chart** of rating values.

        - **Insight:** The distribution is left-skewed (negative skew), with the most frequent ratings being 4.0 and 3.0. This indicates users are generally positive and tend to rate movies they like.
    - **Multivariate Analysis:**
        - Analysed the relationship between Users and Movies via **Sparsity Analysis**. The high sparsity (98.3%) dictates the need for matrix factorisation or dimensionality reduction techniques rather than simple lookups.

Step 4: Modelling

- **Building Base Models:**
    - Utilised the **scikit-surprise** library, an industry-standard package for recommender systems.
    - **Collaborative Filtering:** Implemented a memory-based approach (e.g., K-Nearest Neighbours or KNNBasic) to compute similarity between users/items.
    - **Algorithm:** Utilised **Cosine Similarity** to measure the angle between user preference vectors.
- **Optimization:**
    - **Train-Test Split:** Split the data to ensure the model is evaluated on unseen ratings to prevent overfitting.
    - **Scalability Mitigation:** Addressed the $O(n^2)$ computational complexity of similarity calculations by planning for Matrix Factorisation (SVD) and approximate nearest neighbours for larger-scale deployments.

Step 5: Evaluation

- **Testing on Metrics:**
    - **RMSE (Root Mean Squared Error):** Used as the primary metric to measure the average deviation between predicted ratings and actual user ratings.
    - **MAE (Mean Absolute Error):** Calculated to understand the average magnitude of errors.
- **Business Risks identified:**
    - **Filter Bubble:** The risk of users getting stuck in a preference loop (mitigation: introduce 20% exploration/randomness).
    - **Cold Start:** Difficulty recommending to new users with no history (mitigation: hybrid approach using genre metadata).

Step 6: Deployment

- **Deployment Strategy:**
    - **Integration:** The model is designed to be integrated into the CineStream platform to generate the "Top-5" shelf on the user homepage.
    - **Monitoring:** Continuous A/B testing to verify if the offline RMSE improvements translate to online user satisfaction (e.g., click-through rates).
    - **Retraining:** Implemented a schedule for monthly retraining to capture changing user preferences and new movie additions.