SUMMARY & INSIGHTS
ON TASKS
OF
DATA SCIENCE INTERNSHIP at ARCHTECHNOLOGIES

Intern: Sana Azeem
Intern ID: ARCH-2509-0783

TASK 3: Customer Segmentation

Use a customer dataset to group customers into different segments based on their purchasing behavior. Apply clustering techniques like KMeans to identify patterns and describe the characteristics of each segment. Preprocess the data, perform clustering, and visualize the results.

Summary & Insights:

In this project, we segment customers into meaningful clusters based on their behavioral and transactional attributes.
We aim to uncover patterns that can help a business understand its customers — e.g., loyal buyers, occasional shoppers, or low-engagement users.

Steps Completed:

1. Data Loading & Selection

- Loaded `shopping_trends_updated.csv` (≈ 3900 rows × 18 columns).
- Selected only behavioral and transactional attributes relevant to segmentation.

2. Feature Engineering

- Converted **purchase frequency** from words (e.g., “Weekly”, “Monthly”) to numeric scales (e.g., 52, 12).
- Replaced rare product categories and payment methods with **“Other”** to reduce noise.

3. Missing Value Handling

- Numeric: replaced missing values with the **median**.
- Categorical : replaced missing values with the **mode**

4. Encoding & Scaling

- Label-encoded categorical variables for numerical representation.
- Scaled numeric features using **MinMaxScaler** to maintain equal importance.

5. UMAP + HDBSCAN Clustering

- **UMAP (Uniform Manifold Approximation and Projection)** was used to reduce high-dimensional data (numerical + categorical) into a 5D latent space.
- **HDBSCAN (Hierarchical Density-Based Spatial Clustering)** was then applied to detect natural customer groupings.

6. Model Evaluation

- Calculated the **Silhouette Score** for non-noise points — measuring intra-cluster cohesion vs inter-cluster separation.
- Identified the number of clusters formed and the proportion of points classified as noise.

7. Cluster Labeling & Profiling

- Each customer was assigned a cluster ID.
- Calculated:
    - **Mean values** for numeric attributes (Age, Purchase Amount, etc.)
    - **Most common categories** for categorical attributes (Category, Payment Method, etc.)
    - **Cluster sizes** (number of customers and percentage share).

Insights:

- UMAP embeddings effectively separated customer behavior patterns.
- HDBSCAN automatically discovered 3 major clusters without predefining k.
- The **Silhouette Score (0.765)** indicates moderately good cluster separation — acceptable for real-world behavioral data.

Conclusion:

The **UMAP + HDBSCAN clustering model** effectively grouped all **3,900 customers** into **three well-defined clusters**, with **no data points classified as noise**.
The model achieved a **Silhouette Score of 0.765**, reflecting a **high level of cluster quality**, where customers within each group share similar purchasing behaviors, and the clusters are clearly separated from one another.

Overall, these results indicate that the chosen **UMAP + HDBSCAN approach** produced meaningful and reliable customer segments that can provide valuable insights for targeted marketing and customer relationship strategies.

TASK 4: Movie Rating Prediction

Using a movie ratings dataset, build a model to predict how a user might rate a movie they haven't seen yet. Preprocess the data, use collaborative filtering or a regression model, and evaluate its performance.

Summary & Insights:

In this project, we built a high-accuracy hybrid movie recommender system that leverages both collaborative filtering (SVD) and content-based learning (LightGBM).
The hybrid approach aims to combine user–item interaction patterns with engineered user/movie-level features for more personalized and robust predictions.

Steps Completed:

1. Data Loading and Cleaning:

Datasets used:

ratings.csv— user–movie ratings (userId, movieId, rating, timestamp)
movies.csv— movie metadata (movieId, title, genres)

Data validation & cleaning steps:

- Removed duplicate (userId, movieId) pairs.
- Dropped missing or invalid ratings.
- Filtered ratings to stay within the valid 0.5–5.0 range.
- Removed users and movies with fewer than **5 interactions** to ensure statistical reliability.

Result: Cleaned dataset shape → **(90,274 rows × 6 columns)**, ready for modeling.

2. Feature Engineering:

To improve model expressiveness, several new features were created:

User and Movie Statistics:

user_avg – Average rating given by each user.
user_count – Total number of ratings given by each user
movie_avg – Average rating received by each movie
movie_count – Total number of ratings received by each movie.

Interaction Features

rating_gap – Absolute difference between the user's and movie's average ratings.
interaction_strength – Product of user and movie average ratings.
popularity_ratio – Ratio of movie popularity to user activity

Genre Encoding

- Extracted the first listed genre for each movie and one-hot encoded it, allowing the model to learn genre-specific trends.

SVD Predictions

- Trained a Singular Value Decomposition (SVD) model to capture latent user–item interactions.

- The predicted SVD rating (svd_pred) was added as a new feature for the LightGBM model, integrating collaborative filtering signals into the final supervised learner.

3. Model Training Pipeline:

Step 1: SVD Model

- **Algorithm:** Surprise `SVD`
- **Hyperparameters:** `n_factors=50`, `n_epochs=40`, `lr_all=0.005`, `reg_all=0.02`
- **Training:** 80/20 split with random seed = 42
- Generated `svd_pred` predictions for all `(user, movie)` pairs in the dataset.

Step 2: LightGBM Regressor

- **Algorithm:** Gradient-boosted decision trees (LightGBM)
- **Parameters:**
  - `n_estimators=800`, `learning_rate=0.05`, `num_leaves=40`, `subsample=0.8`, `colsample_bytree=0.8`, `reg_alpha=0.1`, `reg_lambda=0.1`
- **Input features:**
  - User/movie statistics, interaction metrics, genre dummies, and SVD prediction.
- **Target:** Actual movie rating.
- **Split:** 80/20 training/testing.

4. Model Evaluation:

Following evaluation metrics obtained:
RMSE: 0.4361 — Very low error; the model predicts ratings within ±0.43 on average.
MAE: 0.3363 — Average absolute deviation of approximately 0.33 rating points.
R^2: 0.8199 — Explains about 82% of the variance in user ratings.

Insights:

The hybrid structure thus provides a meaningful synergy between collaborative and content-based signals.

Conclusion

The **Hybrid SVD + LightGBM Movie Recommender** successfully merges collaborative and content-based paradigms into a unified predictive model.

- It delivers **high accuracy (RMSE ≈ 0.43, R^2 ≈ 0.82)**
- The feature engineering pipeline effectively captures **user behavior, movie popularity, and genre influence.**

- The SVD-based embeddings and LightGBM ensemble together produce **robust, interpretable, and scalable** predictions.