BITS PILANI DIGITAL

FIRST TRIMESTER 2025-26

ADVANCED APEX PROJECT 1

Project Title	AisleAffinity - Cartographer - Mapping Tastes to Carts	
Supervisor Name	Mr. Shubham Gupta	
Name of the Learner (with BITS ID)	Name of the Learner	BITS ID
	Debjit Adak	2025EM1100283
	Anisha Das	2025EM1100192
	Akash Ghosh	2025EM1100017
	Kritanhik Biswas	2025EM1100104

Courses Relevant for the Project & Corresponding Trimester

Sl. No.	Subject Name	State the relevance to Project
1	Statistical Modelling & Inferencing	Frames the problem and metrics (Recall@K/NDCG for recommendations; Silhouette/DBI for clusters), selects/validates models (KMeans baseline, UMAP→HDBSCAN) and quantifies expected AOV lift with rigor
2	Data Pre-processing	Ensures clean, trustworthy data via audits, PK/FK checks, dedup, type fixes, and leakage-free splits; builds baskets from prior orders as a reliable foundation for embeddings
3	Feature Engineering	Turns raw carts into signal: trains Product2Vec on baskets, aggregates customer embeddings (mean/recency-weighted), and adds behavioral features (order_count, avg_basket_size, reorder_ratio)
4	Data Visualization & Storytelling	Communicates insights and action: EDA distributions, co-purchase heatmaps, UMAP maps, and segment profiles in dashboards to justify bundle/reco strategies
5	Data Stores & Pipelines	Makes the work reproducible and shareable: Kaggle CLI ingestion; Parquet storage of clean tables/embeddings/holdout; simple loaders/pipelines to retrain, version, and serve recommendations at scale

Title

AisleAffinity — Cartographer — Mapping tastes to carts

Advanced Apex Project Proposal

Team & Supervisor

- ***Team: The cluster busters (Akash Ghosh, Anisha Das, Kritanhik Biswas, Debjit Adak)***
- ***Supervisor: Shubham Gupta***
- Date:-

1. ## *Problem Statement*

   Most online stores show generic “you may also like” items that ignore different shopper tastes. This leads to low cross-category adoption and flat average order value (AOV). We aim to learn product affinities from baskets, group shoppers by taste using embeddings and clustering, and use these segments to serve more relevant add-ons and bundles.

2. ## *Business Goal*

   Increase AOV by 5–8% and cross-category purchase rate by 8–10% on a holdout set by deploying segment-aware cross-sell bundles and recommendations. Deliver reusable customer embeddings, segment labels, and interpretable segment profiles (top aisles/departments).

3. ## *Data Source*

   We will use the Instacart 2017 grocery shopping dataset (public mirror) with orders, line-items, and product categories.

- ***Source Platform***: Kaggle (Dataset)
- Link: https://www.kaggle.com/datasets/psparks/instacart-market-basket-analysis
- Full Citation:
  Instacart. (2017). Instacart Market Basket Analysis [Data set]. Kaggle. Retrieved [insert date], from https://www.kaggle.com/datasets/psparks/instacart-market-basket-analysis

4. Tools & Technologies

- Programming Language: Python
- Core Libraries:
  - Data Manipulation & Analysis: Pandas, NumPy
  - Machine Learning: scikit-learn, Gensim (Word2Vec), umap-learn, hdbscan
  - Data Visualization & Storytelling: Matplotlib, Seaborn, Plotly
- ***Development Environment:*** [Our team will primarily use: _COLAB_]
  (Recommended options: Google Colab, VS Code, Jupyter Notebook, etc.)
- ***BI Tools:*** Power BI (final dashboard)

5. ### *Project Workflow*

   We will follow a structured data science lifecycle:
   Data Acquisition → Data Cleaning & Preprocessing → Exploratory Data Analysis (EDA) → Feature Engineering → Model Building & Training → Model Evaluation → Reporting & Visualization

1. *Data Acquisition*

- Fetch the dataset from Kaggle using its API (no manual downloads).
- Verify presence of the six CSVs: aisles.csv, departments.csv, orders.csv, order_products__prior.csv, order_products__train.csv, products.csv.

2. *Preprocessing*

- Join product metadata (products → aisles → departments).
- Build baskets per order_id; use “prior” orders for modeling and treat each user’s “train” order as a holdout next basket.
- Handle rare products (min frequency threshold), duplicates, types, and basic consistency checks.

3. *EDA*

- Summary stats: orders per user, basket size distribution, reorder rates, top aisles/departments.
- Visualizations: product popularity, co-purchase heatmaps, weekday/hour patterns (order_dow/hour).

4. *Feature Engineering*

- Product2Vec: Train Word2Vec on baskets (co-occurrence) to learn product embeddings.
- Customer embeddings: Mean of purchased product vectors per user (optionally weighted by recency/frequency).
- Optional: Light behavioral features (prior order count, avg basket size, reorder ratio).
- Optional dimensionality reduction (UMAP/PCA) for clustering and visualization.

5. *Modeling & Training*

- Baseline: KMeans clustering on customer embeddings (simple, interpretable).
- Advanced: UMAP (10D) → HDBSCAN (auto-k, handles noise). Profile clusters with top aisles/departments and margin proxies.

6. *Model Evaluation*

- Clustering metrics: Silhouette Score, Davies–Bouldin Index (report best configuration).
- Recommendation metrics: Recall@K and NDCG@K using each user’s holdout “train” order as ground truth next basket; nearest-neighbor recommendations from embeddings.
- Sanity checks: distinct segment profiles (top aisles/departments), stability across random seeds.

7. *Reporting & Visualization*

- Power BI dashboard: segment distribution; top aisles/departments by segment; KPI cards (Silhouette/DBI, Recall@K/NDCG); basic filters (segment, department/aisle).
- Brief segment playbooks: how to target cross-sell for 2–3 key segments.

6. *Data Extraction*

The dataset is acquired directly from Kaggle via the official API (no manual download).

- Automate the Process: Use Kaggle CLI to authenticate and download to data/raw/.
- Ensure Reproducibility: Scripted pulls with the fixed dataset ID.
- Prepare for Analysis: Unzip and load into Pandas.
- Example commands:
  - kaggle datasets download -d psparks/instacart-market-basket-analysis -p data/raw
  - unzip -o data/raw/*.zip -d data/raw
- Notebook: data_extraction_instacart.ipynb

7. ***Schema/Data Dictionary***

   This data dictionary will be created after inspecting the dataset and joins.

- Excel sheet: Data_Dictionary_Instacart_Affinity.xlsx

Data model (initial structure)

- aisles.csv — PK: aisle_id
    - Columns: aisle_id (int), aisle (string)
    - Description: Aisle mapping
- departments.csv — PK: department_id
    - Columns: department_id (int), department (string)
    - Description: Department mapping
- products.csv — PK: product_id
    - Columns: product_id (int), product_name (string), aisle_id (int, FK), department_id (int, FK)
    - Description: Product catalog with category links
- orders.csv — PK: order_id
    - Columns: order_id (int), user_id (int), eval_set (prior/train/test), order_number (int), order_dow (int), order_hour_of_day (int), days_since_prior_order (float)
    - Description: Order metadata and sequencing
- order_products__prior.csv — PK: (order_id, product_id, add_to_cart_order)
    - Columns: order_id (int), product_id (int), add_to_cart_order (int), reordered (int/bool)
    - Description: Line items for historical (prior) orders
- order_products__train.csv — PK: (order_id, product_id, add_to_cart_order)
    - Columns: order_id (int), product_id (int), add_to_cart_order (int), reordered (int/bool)
    - Description: Line items for the labeled “next” order used as holdout

Join keys

- products.aisle_id → aisles.aisle_id
- products.department_id → departments.department_id
- order_products__*.order_id → orders.order_id
- orders.user_id groups orders to customers; “prior” used for modeling, “train” as holdout

Submission notes

- All analysis in Jupyter Notebooks (cleaning, EDA, feature engineering, modeling, visualization).
- README with steps to run; requirements.txt with versions.
- Power BI dashboard file or screenshots with brief captions.