Project Rapport

Week 1 : The Blueprint & The Pitch

Auteurs :

Antonin	DOAT
Augustin	MOUTON
Vincent	MORIN
Nicolas	LAINÉ
Alexis	LULIN

Enseignant :

M.MEDEIROS MACHADO

We certify that this work is original, that it is the result of joint work by the pair, and that it has been written independently.

Paris, le 11/11/2025

PART I. The Startup Team

Our team is composed of Alexis LULIN, Project Lead, responsible for managing the project plan, schedule, and repository; Augustin MOUTON, Data Engineer, in charge of the data pipeline and storage; Antonin DOAT, Lead ML Engineer, overseeing the core model implementation; Nicolas LAINE, ML Engineer, handling model training and serving; and Vincent MORIN, Systems Engineer, managing the full-stack application and deployment.

PART II. The Problem & Domain

A. Domain: Personalized Travel Recommendation

Our startup, TripplyBuddy, operates in the travel technology domain, aiming to revolutionize how travelers discover their next destination. With millions of people overwhelmed by choices and advertisements, modern travelers often struggle to plan trips that align with their personal preferences, budget, and time constraints.

B. Problem definition

Current travel recommendation systems often suggest destinations based on popularity or averages, leading to repetitive recommendations and low satisfaction. Travelers now shift preferences dynamically across adventure, relaxation, and cultural experiences. Our goal is to build a personalized system that adapts to these evolving interests, providing context-aware suggestions for destinations, hotels, and transportation.

C. Target Users

Our target audience includes frequent travelers seeking automated trip curation, casual vacationers looking for themed recommendations (family-friendly, romantic, budget), and digital nomads seeking destinations that fit work-life balance factors such as cost, connectivity, and lifestyle.

D. Value proposition

TripplyBuddy provides personalized travel recommendations by learning from users' past trips and evolving preferences. It also promotes sustainable tourism by suggesting authentic, less-crowded destinations. Positioned at the intersection of AI and travel, TripplyBuddy aims to redefine how people discover and plan their journeys, addressing a growing market with strong investor potential.

PART III. Data Strategie

A. Dataset V1

Our Version 1 dataset uses the Kaggle Traveler Trip Dataset, containing over 2,000 records with traveler demographics, trip details (destination, country, duration, purpose), transportation, accommodation, and cost. It enables modeling sequential user behavior across multiple trips and predicting each traveler's preferred future destinations.

B. Future Data Collection Plan

To extend the dataset, we will collect real-world feedback through user surveys and generate synthetic trip sessions using LLM-based simulators. These approaches create a continuous data pipeline that supports model training and evaluation under dynamic, real-world conditions.

PART IV. The innovation

A. Introduction to the paper

We selected the paper "*Hierarchical Graph Information Bottleneck for Multi-Behavior Recommendation*" by Zhang et al. (2025). The central idea is the HGIB framework, which "preserves discriminative information relevant to target behavior prediction while eliminating irrelevant noise." (Zhang et al., 2025, Section 4.1.1). It combines graph neural networks with the Information Bottleneck principle to retain label-relevant information while compressing redundant input data.

HGIB models multi-behavior interaction data hierarchically, capturing complex relationships between behaviors. It uses a Graph Refinement Encoder (GRE) to remove noisy user-item interactions and auxiliary losses to preserve task-relevant information while compressing irrelevant patterns.

In our adaptation, HGIB will be applied to travel destination recommendations. Auxiliary behaviors such as transportation mode, accommodation type, and budget will support predicting the main behavior: destination selection. This hierarchical approach aims to mitigate negative transfer and improve recommendation accuracy within the travel domain.

B. Context-Awareness

Following HGIB's strategy of using auxiliary behaviors, we incorporate temporal and traveler-specific features as contextual signals. Trip start date and duration capture seasonal preferences and typical lengths, while traveler profiles (age, nationality) enable personalized recommendations. These features act as auxiliary signals within the hierarchical GNN, enhancing prediction while filtering irrelevant information, similar to HGIB's approach with views and carts.

C. Multi-Behavior Graph Construction

In e-commerce, HGIB builds separate graphs for behaviors like view, cart, and buy. For our travel dataset, auxiliary behaviors are reinterpreted to capture traveler preferences: accommodation type (hotel, hostel, Airbnb), transportation mode (plane, train, car), and budget (total accommodation and transportation cost) each form distinct interaction graphs. This structure allows the hierarchical GNN to model complex interactions between traveler profiles and trip choices, mirroring HGIB's multi-behavior approach and improving recommendation accuracy.

D. Explainability

Using the attention weights from the hierarchical GNN, the model can deliver interpretable recommendations. For instance, it might recommend Tokyo because the traveler previously used trains, preferred Airbnb stays, and selected longer-duration trips. This parallels HGIB's ability to emphasize task-relevant patterns in embeddings while filtering out irrelevant noise.

E. Hybridization

To address the cold-start problem, we combine HGIB's collaborative embeddings with content-based features of destinations, such as cultural tags, climate, and city/country type. This hybrid approach leverages multiple information sources to produce robust embeddings while filtering irrelevant noise, following HGIB's principle.

F. Conclusion

In summary, our HGIB adaptation for travel builds on four key components: incorporating temporal and traveler-specific contextual signals, constructing multi-behavior graphs from travel features, integrating attention-based explainability, and using a hybrid approach to support cold-start users. Together, these leverage HGIB's hierarchical multi-behavior representation for effective travel destination recommendations.

PART V. Technical Stack

G. Libraries

For the implementation, we use standard data processing libraries such as NumPy, Pandas, and Scikit-learn, with Matplotlib and Seaborn for visualizations. The model is built using PyTorch, supported by PyTorch Geometric (or DGL) for graph-based computations. Streamlit is used to display results and provide an interactive user interface.

Additional libraries may be added as needed throughout the project to support emerging requirements

H. Framework

The system is based on the HGIB (Hierarchical Graph Information Bottleneck) framework, which learns compact yet informative representations for multi-behavior user interactions.

The Graph Refinement Encoder (GRE) is integrated to dynamically clean the interaction graph by pruning redundant or noisy edges, mitigating negative transfer across heterogeneous behaviors.

I. Database

For this project, we will use **PostgreSQL** as the main database, which is robust, scalable, and fits the structured format of our dataset while supporting future extensions such as user feedback and synthetic sessions.

The Kaggle Traveler Trip Dataset will initially be imported into a single main table, which can later be enriched with additional fields or split into multiple tables if needed. A separate user account table will support authentication, storing user IDs, securely hashed passwords, and optional metadata. Model-generated embeddings can be stored using **pgvector** for efficient similarity search and recommendation queries.

A Python integration layer will manage all database interactions—reading trip histories, adding feedback, retrieving user data, and validating accounts—ensuring a clear separation between backend, model, and frontend.

PART VI. Schedule & Task Breakdown

A. Overview

The TripplyBuddy project spans five weeks, with weekly tasks and deliverables to complete, progressing from concept to implementation, testing, and a working prototype (MVP).

B. Week 1 – November 05–11, 2025

Session: November 12 (18h30–20h45)

Theme: The Blueprint & The Pitch

Objectives: Define startup concept and problem statement, establish domain (Personalized Travel Recommendation), assign team roles, study HGIB paper and plan adaptation, outline data strategy and sources.

Deliverables: Initial report; Pitch summary and project direction validation.

C. Week 2 – November 12–16, 2025

Session: November 21 (08h30–12h45)

Theme: Data Exploration & Graph Design

Objectives: Acquire and clean Traveler Trip Dataset, perform EDA (demographics, destinations, travel modes, budgets), design multi-behavior graph (nodes, edges, relationships), start preprocessing pipeline, define startup visual style and begin Streamlit development.

Team Outputs: Cleaned dataset and schema; Preliminary EDA report with visuals and insights; Draft graph-based data model; Streamlit prototype.

Deliverables: 2-page Status Report & Risk Assessment.

D. Week 3 – November 17–23, 2025

Session: Independent Work Week (No Scheduled Class)

Theme: Model Development – HGIB Adaptation

Objectives: Implement HGIB model using PyTorch Geometric, integrate auxiliary behaviors (transportation, accommodation, budget), implement contextual embeddings (age, nationality, seasonality), finalize Streamlit, and begin backend API research (FastAPI/Flask).

Team Outputs: Working prototype of HGIB model; Code documentation; GitHub version control validation.

Deliverables: 2-page Status Report & Risk Assessment.

E. Week 4 – November 24–30, 2025

Session: No Scheduled Class

Theme: Model Training, Evaluation & Explainability

Objectives: Train and evaluate model (baseline vs HGIB), implement attention-based explainability visualization, address cold-start issues with hybrid features, finalize performance metrics, deploy full system (Streamlit + API) on cloud.

Team Outputs: Evaluation report and graphs; Explainability examples; Updated technical report; Cloud-deployed version.

F. Week 5 – December 1–7, 2025

Sessions: December 5 (08h30–12h45 & 15h15–17h15)

Theme: Finalization & Presentation

Objectives**:** Conduct final testing and optimization, prepare project presentation (slides + demo), submit complete report and code repository, finalize deployment and end-to-end integration.

Deliverables**:** Final written report; Stable deployed version (production-ready); Oral presentation deck.