Status Report & Risk Assessment

Auteurs :

Vincent MORIN
Antonin DOAT
Augustin MOUTON
Nicolas LAINE
Alexis LULIN

Enseignant :

M. MEDEIROS MACHADO

Nous attestons que ce travail est original, qu'il est le fruit d'un travail commun et qu'il a été rédigé de manière autonome.

PARTIE I. Dataset

A. Synthetic Data Generation Protocol

Due to the small size and high sparsity of the first dataset, we regenerated a new synthetic dataset, this time guided by distributions derived from real data. We introduced a user dataset to record user preferences, which then serve as latent variables for coherent trip generation. The number of trips per user now follows a realistic normal distribution, fixing inconsistencies from the earlier version. We also corrected destination weighting, preventing unrealistic biases such as over-visiting Kathmandu compared to Paris or New York. Finally, we increased the overall dataset size while enforcing strict constraints to ensure coherence, realism, and better suitability for training the HGIB model.

B. PostGre

All project datasets are now hosted on the free Heroku cloud, ensuring accessibility, centralization, and smooth integration. At the moment, only the travel dataset is fully connected to the application and used by the current features. The last remaining step is to link the user dataset to the system, especially to enable the user login module and preference handling within the Streamlit interface.

PARTIE II. HGIB Framework

A. Model and training

The implementation is organized around three modular components designed to ensure full MLOps readiness. The overall robustness of the architecture comes from its graph-based formulation, in which all User and Destination IDs are encoded using learnable Identity Embeddings. This mechanism provides a stable and consistent representation for every entity in the graph.

The central innovation lies in the Edge Behavior Encoder. This module aggregates the contextual information of each trip by combining categorical attributes (Accommodation type, Transport mode, Season) with normalized numerical features (Costs and Duration). These elements are fused into a dense behavior vector, which is then used by the Graph Attention Networks (GAT) to weight user–item interactions according to their contextual importance. The resulting node embeddings are processed through a Variational Graph Autoencoder (VGAE), applying the Information Bottleneck principle: the KL Divergence term enforces compression and encourages more robust generalization.

The final output of the train.py script corresponds to the deployment artifacts. This folder contains three essential files:

- hgib_model.pth: the optimized model weights,
- mappings.pkl: the dictionary mapping internal integer IDs back to human-readable labels (e.g., user names, cities), required by the API,

- config.json: the set of hyperparameters needed by the Systems Engineer to correctly reload and serve the model in production.

PARTIE III. PreProcessing Pipeline

Since the dataset was already clean, we did not need to handle null values or typos. However, we extracted new features from our columns, such as:

- The season (from the start date).
- The country (from the destination).

We have also mapped cities to countries and vice versa if a line was missing either one of the two for the destination feature.

For the remaining preprocessing, we encoded all categorical data into numerical values by mapping each unique category to a number.

PARTIE IV. Metrics

Recommendation relevance is measured through ranking indicators such as PrecisionK, RecallK, HitK, and NDCG, which evaluate whether the most suitable destinations appear among the top suggestions.

The quality of user–destination link prediction is captured with metrics like AUC, Average Precision, and MRR, which are well suited to graph-based models.

Finally, user experience is evaluated through measures of diversity, novelty, and serendipity to ensure that recommendations are varied, original, and meaningful, along with context-aware metrics that account for factors such as cost constraints or transport preferences.

Together, these indicators provide a comprehensive view of the system’s performance and overall value.

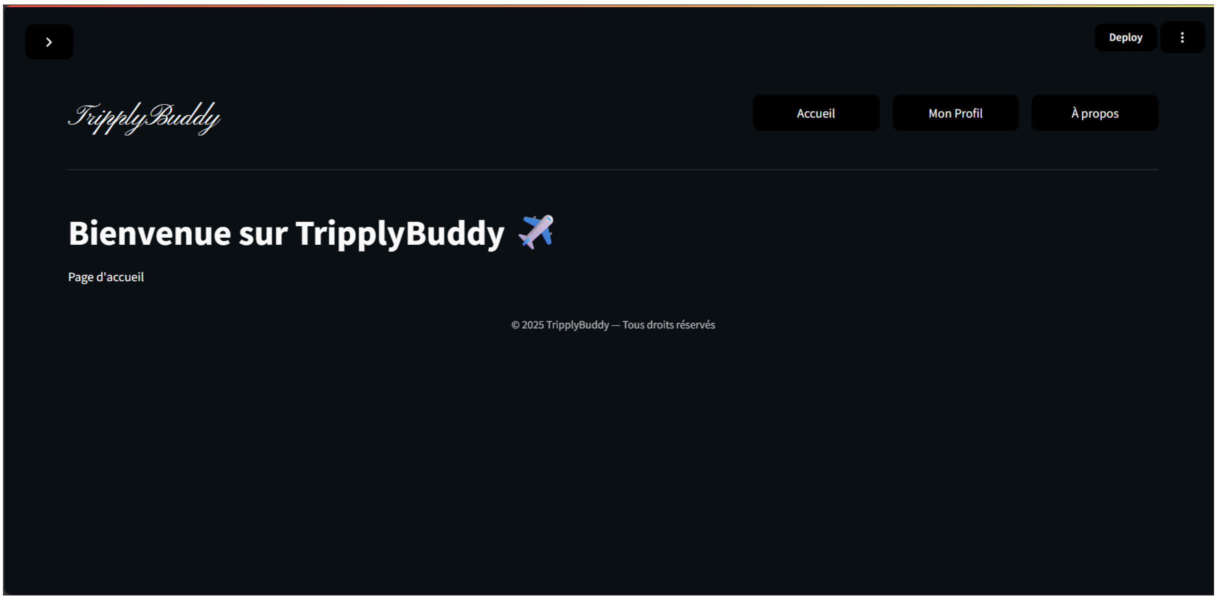
To track the model performance, we could store these metrics for every prediction and have a dashboard keeping track of the evolution of the metrics.

PARTIE V. Streamlit & API

Streamlit development is currently **paused** before the final integration of login, history, and recommendations. This is due to two critical dependencies: pending **PostgreSQL DB access** (needed for user features and the `person` table) and waiting for **Model Predictions**. I have started research on the Backend API (FastAPI/Flask). **Note on Streamlit Frontend:** I encountered issues with **global variables** and styling, specifically that I can no longer use custom headers for navigation; I will switch to using **buttons** for page navigation instead. I will maintain Streamlit's default background style and the standard classic white text color, but I will ensure the **font**

style matches the Pynion (Canva) template. The immediate priority is to gain DB access to unblock the project, allowing me to begin API development and subsequently integrate the login/history features into the Streamlit application.

1. Accueil Page

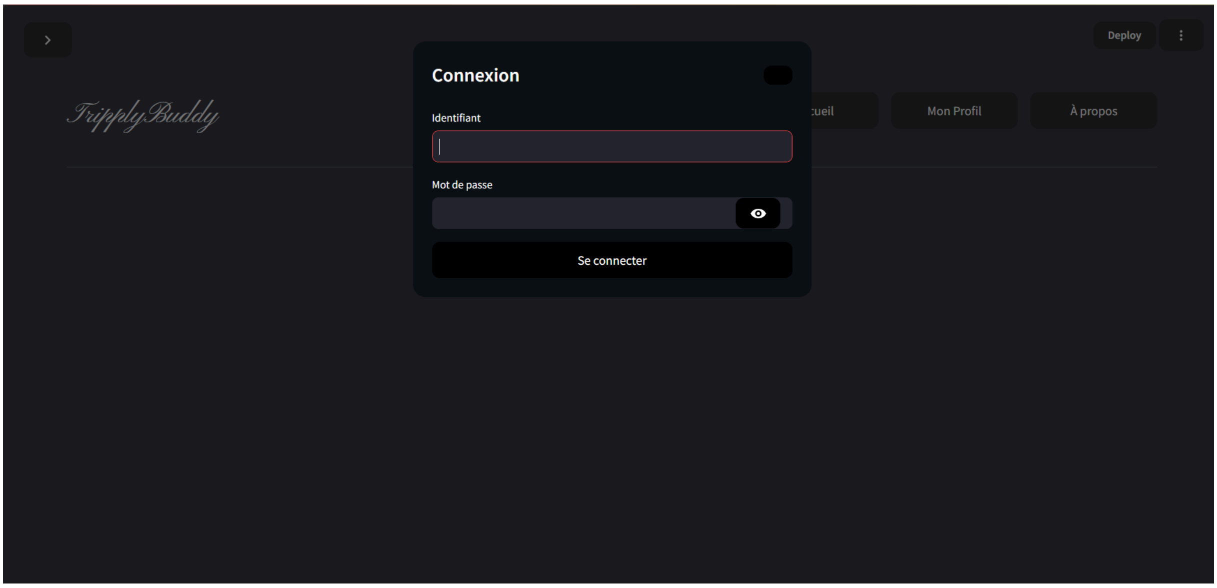


Bienvenue sur TripplyBuddy ✈️

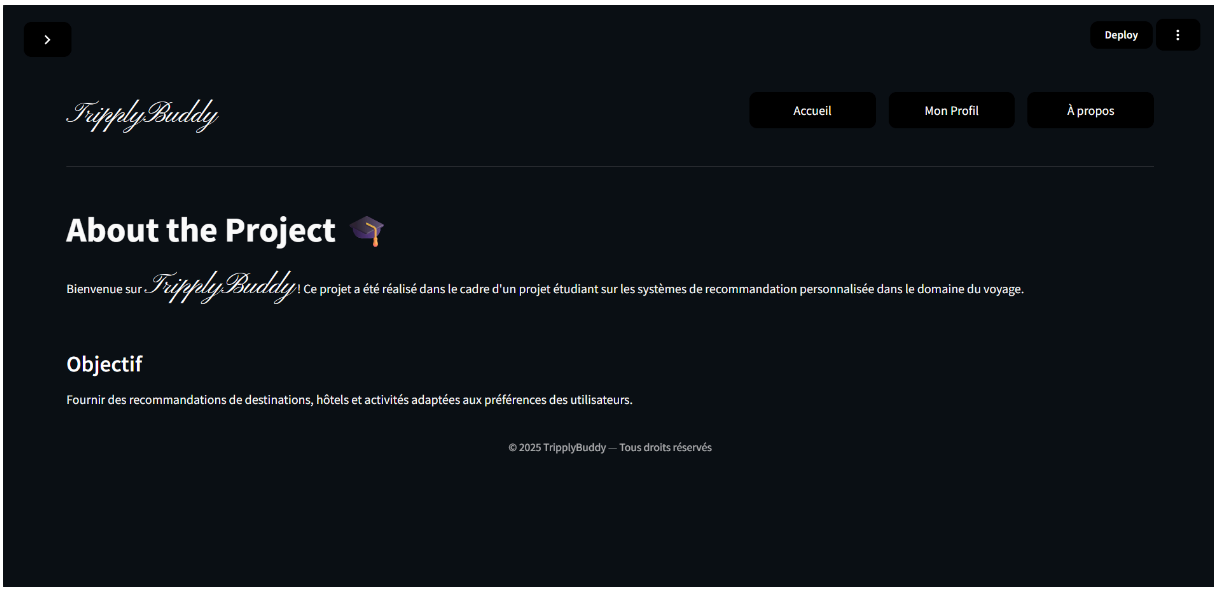
Page d'accueil

© 2025 TripplyBuddy — Tous droits réservés

2. Connexion page



3. A propos



About the Project 🎓

Bienvenue sur TripplyBuddy ! Ce projet a été réalisé dans le cadre d'un projet étudiant sur les systèmes de recommandation personnalisée dans le domaine du voyage.

Objectif

Fournir des recommandations de destinations, hôtels et activités adaptées aux préférences des utilisateurs.

© 2025 TripplyBuddy — Tous droits réservés

As previously noted, our current delay in both **dataset processing** and **database implementation** directly explains the lack of completed features on our Streamlit interface. The next steps are focused on creating the function to connect using the database, implementing a register page, developing a "My Profile" page that will eventually display recommendations and past travels, and finalizing an "Accueil" (Home) page that lists all possible destinations.