Menu Engineering Supercharged

MDS Capstone Proposal Report

Hankun Xiao, Yasmin Hassan, Jiaxin Zhang, Zhiwei Zhang

Table of contents

1 Executive Summary

This project supports Heymate's mission to empower restaurant clients with smarter, data-driven tools. We propose a recommendation engine that identifies popular menu items based on market trends, offering a practical and flexible approach to menu optimization. While the initial version uses popularity-based scoring, future enhancements will leverage pre-trained models and richer data sources to improve accuracy and impact. Heymate currently lacks a recommendation feature and is looking to us to build a framework that can support future development. This tool will help restaurants optimize offerings, reduce waste, and boost sales.

2 Introduction and Project Motivation

Many restaurants operate with limited insight into which dishes are popular. This knowledge gap can lead to bloated menus, ingredient waste, and missed revenue opportunities.
Heymate, our project partner, is a Canadian tech company that offers cashback rewards and a full-suite business management platform for small and medium-sized enterprises (SMEs). Since

many of Heymate's clients are restaurants, they are a primary focus for added value through smart tools.
This project helps Heymate enhance its role as a strategic partner by providing restaurant clients with a menu recommendation tool designed to answer two key questions:

- How can menus match market trends?
- How can market data help restaurant owners make better-informed decisions?

The solution consists of the following two components:

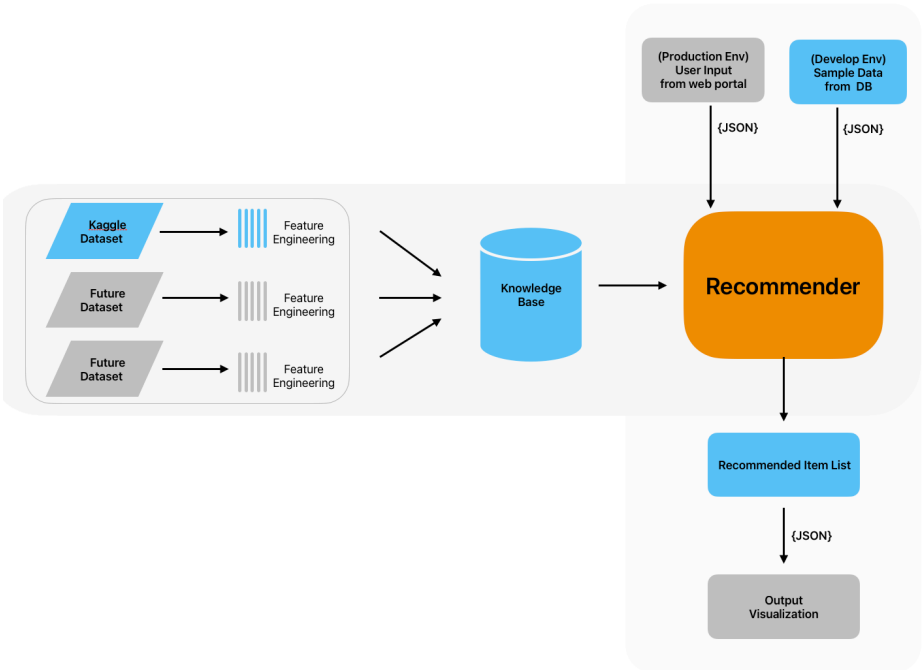
- Develop a recommender system that suggests menu adjustments based on existing items and restaurant type
- Build a data pipeline that continuously ingests new data to enhance the model's accuracy and relevance over time.

3 Solution Overview

Figure 1 shows the system architecture. The **vertical flow** represents the Recommendation Pipeline, which takes a restaurant's type and menu and returns a list of recommended items through two APIs. In production, the input comes from a web request from the restaurant management portal. In development, the input is accessed directly from Heymate's database.

The **horizontal flow** represents the Knowledge Base Construction Pipeline. It starts with public restaurant menu datasets, applies feature engineering using LLMs to standardize names and tags, and stores the cleaned data with source labels. This pipeline includes an API for feature extraction and wee will build the initial knowledge base. For the **Recommender**, we will develop a query-based baseline algorithm using cuisine popularity.

Figure 1: Solution Architecture

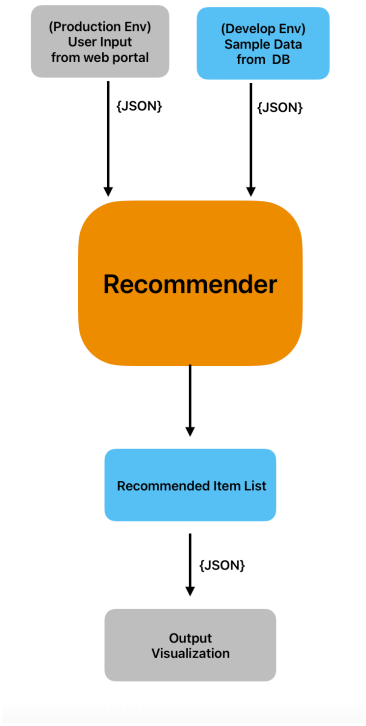


3.1 Data Sources

To support the development of our restaurant menu recommendation system, we aggregate and validate data for two primary use cases:

3.1.1 Input Data

Figure 2: Input Data



The internal Heymate database and user-submitted forms serve as foundational inputs, providing:

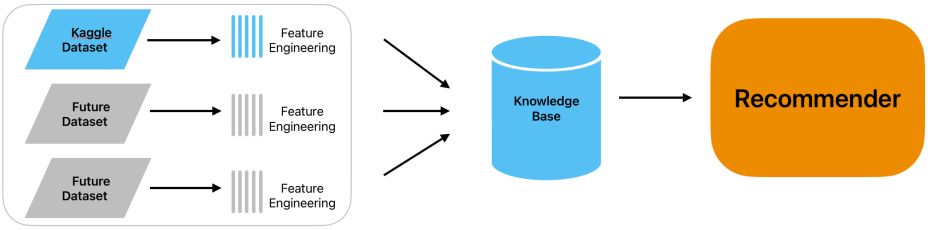
- Restaurant Information: Name, Type, and Location
- Menu Information: Item Name, Category, and Price

3.1.2 Modelling Data

To enhance our understanding of menu popularity and strengthen the foundation of our recommendation model, we plan to incorporate external data sources including:

- Uber Eats USA Restaurants and Menus (Sakib (2024) from Kaggle): Offers modern restaurant-level data including ratings, menu items, and restaurant types
- What's on the Menu? (Library (2016) from Kaggle): Provides historical menu data including item frequency, restaurant names, categories, and prices

Figure 3: Modelling Data



3.2 Data Validation (Initial EDA)

Our initial exploratory analysis revealed two key challenges that must be addressed before modelling.

3.2.1 Data Entry Challenge

Menu item names are inconsistently formatted across restaurants as some include item numbers, others use different naming styles or even different languages. For example, the same dish may appear as "Seafood Fried Rice", "149. Fook Chow Seafood Fried Rice", or a translated version. These inconsistencies make it difficult to compare, group, or analyze menu items reliably.

Table 1: Sample menu name from Heymate database

ProductName	StoreName
Seafood Fried Rice	Fisherman's Terrace
G25. Sauteed Seafood Fried Rice	Yue Ting Seafood Restaurant
149. Fook Chow Seafood Fried Rice	Double Double Restaurant & Wonton Ltd.
160. XO Sauce W/ Seafood Fried Rice XO	Double Double Restaurant & Wonton Ltd.
161. Dried Scallop & Seafood Fried Rice	Double Double Restaurant & Wonton Ltd.
Seafood Fried Rice	Nishiki Sushi Bar
Assorted Seafood Fried Rice	So Good Restaurant
Seafood Fried Rice	Fortune Lamb Dining
Seafood Fried Rice	Richmond
Seafood Fried Rice	Lougheed
Fu-Chow Style Seafood Fried Rice (Topped with sauce)	Lougheed
Foo Chow Style Seafood Fried Rice /	Pelican Seafood Restaurant
Seafood Fried Rice with Pineapple /	Pelican Seafood Restaurant
Deluxe Seafood Fried Rice /	Pelican Seafood Restaurant
35. Seafood Fried Rice	Summer House
Seafood Fried Rice	Valendine
43. Baked Creamy Seafood Fried Rice or Linguine	Neptune Chinese Kitchen (UBC)
921. Dried Scallop & Mixed Seafood Fried Rice	Neptune Chinese Kitchen (UBC)
926. Diced Mixed Seafood Fried Rice	Neptune Chinese Kitchen (UBC)

ProductName	StoreName
3. Seafood Fried Rice Served In Whole Pineapple	Grand Crystal Seafood Restaurant
Seafood Fried Rice	Pinyuexuan Seafood Restaurant
931. Seafood Fried Rice	Lucky Fortune Seafood Restaurant
102. Fu-Chow Style Seafood Fried Rice	Lucky Fortune Seafood Restaurant

To address this, we plan to leverage the capabilities of large language models to clean and standardize menu item names. Inspired by industry practices such as CloudKitchens' approach to multi-location menu management (CloudKitchens (2024)), we will use GPT to:
- Normalize inconsistent item names
- Extract consistent tags

```
{
  "Base Item": "Fried Rice",
  "Flavor Tag": "Seafood"
}
```

3.2.2 Modeling Challenge

A key challenge in our modelling process is integrating multiple datasets that use inconsistent or non-standard identifiers for restaurants and menu items.
To make these datasets work together, we need to:

- Align records using partially matching keys like restaurant and item names
- Normalize variations in naming using LLMs
- Enrich menu profiles by merging structured data from both sources, enabling robust popularity scoring

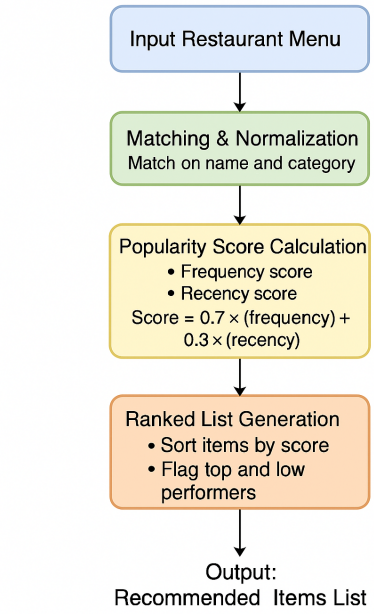
The final output is a unified and cleaned dataset enriched with relevant features such as item frequency, price, and standardized names, laying the groundwork for our Recommender Module.

3.3 Recommender Module

The recommender system operates through the following four steps as shown in figure:

1. **Input Data Collection**
    - Restaurant Menu Data: A list of the restaurant's current menu items (names, categories, and optional descriptions).
    - External Dataset: Built from the Kaggle menu dataset (and future sources), this captures broader market patterns such as how frequently and recently dishes appear across other restaurants.

Figure 4: Flow chart



2. **Matching & Normalization**
   The Menu item names vary widely across restaurants(e.g., "BBQ Chicken Pizza" vs "Barbecue Chicken Pizza"). To address this:

   - We apply name normalization and category matching to link each restaurant's menu item to the closest equivalent in the knowledge base.
   - To do this, we use a query-based matching approach that searches the knowledge base for the most similar items, even if names vary across restaurants.
   - This enables alignment despite variations in wording or format.

3. **Scoring**
   Once matched, each item receives a popularity score based on:

   - Frequency: How often the dish appears across menus in the dataset.
   - Recency: How recently it has been featured

   We use a weighted formula below to combine these factors:

   Scoring formula:
   `Score = 0.7 × (normalized frequency) + 0.3 × (normalized recency)`

   Note: These initial weights are heuristic and subject to refinement as we validate with partner feedback.

4. **Recommendation**
   After computing the popularity scores for each menu item, a ranked list to identify top and underperforming dishes is provided.

3.3.1 Example Use Case

Suppose a restaurant uploads its menu, which includes “Hawaiian Pizza. The recommender system: - Matches “Hawaiian Pizza” to its closest entry in the knowledge base.
- Calculates its popularity score.
- Returns insights such as:
Popularity score: 62 (below market median)
It suggests adding trending dishes like “BBQ Chicken Pizza” that are more popular.

4 Final Data Product

Our solution is a menu recommendation engine and a simple visualization demo, showing the top 20 popular items tailored to each restaurant. The engine is powered by a query-based recommender trained on external datasets. To simulate real-world use, we will test the tool using historical data from Heymate’s existing restaurant clients.

The anticipated business impact includes:

- Helping restaurant clients identify popular items, improve sales, and reduce waste
- Supporting smarter menu decisions and client retention
- Strengthening Heymate’s value proposition as a data-driven platform partner

4.1 Success Criteria

The project will be considered successful if internal tests using Heymate’s internal dataset sample produce results that align with business expectations, and the visualization demo operates as intended. Furthermore, the popularity scoring schema and recommender system will be delivered as open-ended, modular components, enabling Heymate to adapt and expand the solution in future development.

5 Project Timeline

Week	Date	Tasks
Week 1	May 5–9	Proposal, Design Knowledge Base, Database setup
Week 2–4	May 12–30	LLM research and tuning, workspace setup, API framework, core logic
Week 5	June 2–5	Integration and modularization
Week 6	June 9–13	Final review and report

6 Appendices

6.1 Addressing the Matching Challenge

Accurately matching menu items is one of the most challenging aspects due to variability in naming conventions.
Initially, we will use query-based techniques such as string normalization and category filtering.

To improve this process, we plan to explore pretrained models for semantic matching, including:

- Sentence Transformers (SBERT)
- spaCy similarity pipelines
- FastText embeddings

These models will improve the ability to recognize and link creatively named menu items to their standardized counterparts in the knowledge base.

References

CloudKitchens. 2024. "Multi-Channel, Multi-Location Menu Management." https://techblog.cloudkitchens.com/p/multi-channel-multi-location-menu.

Library, New York Public. 2016. "What's on the Menu?" https://www.kaggle.com/datasets/nypl/whats-on-the-menu.

Sakib, Ahmed Shahriar. 2024. "Uber Eats USA Restaurants & Menus." https://www.kaggle.com/datasets/ahmedshahriarsakib/uber-eats-usa-restaurants-menus.