**PERSONALIZED RESTAURANT RECOMMENDER SYSTEM**

****

**Team Members:**

1) Harris Lukundi   - harris.lukundi@student.moringaschool.com

2) Henry Rono  - [henry.rono@student.moringaschool.com](mailto:henry.rono@student.moringaschool.com)

3) Beryl Agai - beryl.agai@student.moringaschool.com

4) Laaria Chris - laaria.chris@student.moringaschool.com

5) Lynete Wangari - lynette.wangari@student.moringaschool.com

6) Brian Muthama - brian.muthama@student.moringaschool.com

# Business Understanding

## Business Overview

The dining industry in various states across the United States is a dynamic and diverse landscape, offering a wide array of options from local eateries to upscale restaurants. However, both locals and tourists often struggle to find restaurants that align with their specific preferences in terms of location, cuisine, and quality. The sheer number of choices, coupled with the lack of a centralized platform for personalized recommendations, makes it challenging for users to make informed dining decisions.

Traditional methods of discovering restaurants, such as relying on word-of-mouth or general review platforms, often fall short in delivering tailored suggestions that cater to individual tastes. These solutions tend to be too broad and do not provide real-time, location-based recommendations, leaving users—especially those in unfamiliar areas—frustrated in their search for the perfect dining spot. The need for a more intelligent, user-centric solution is increasingly evident as the dining scene continues to grow.

The personalized restaurant recommender system addresses this gap by offering users customized dining suggestions based on their preferences and historical data. Utilizing advanced technologies like machine learning, the system provides real-time recommendations that are highly relevant and location-specific. The platform’s database includes detailed information about a vast range of restaurants across various states, capturing essential attributes like business details, reviews, ratings, and operational hours.

By analyzing this comprehensive dataset and integrating it with user profiles, the system can deliver personalized recommendations that match user preferences for cuisine, ambiance, proximity, and more. Whether a user is looking for a cozy brunch spot in California or an authentic BBQ joint in Texas, the system can quickly suggest top-rated options that meet their criteria. This tailored approach not only enhances the dining experience but also supports local businesses by driving targeted traffic to their establishments.

The platform's intuitive interface ensures that users can easily refine their search and explore new dining options with confidence. This personalized restaurant recommender system stands out as a valuable tool, helping users navigate the rich culinary landscape with ease and satisfaction.

## Stakeholder Definition

The possible stakeholders in this project include:

•   Users: Individuals seeking personalized dining recommendations based on their unique preferences.

•   Restaurant Owners: Local businesses aiming to attract targeted customers and increase foot traffic.

•   Investors/Partners: Entities interested in the platform’s growth, scalability, and profitability.

## Problem Statement

As the dining industry expands, finding restaurants that match personal preferences is challenging due to the overwhelming number of options. Current platforms often provide generalized recommendations and lack real-time, location-based suggestions, particularly for users in unfamiliar areas. There is a growing need for a smart, user-friendly recommender system that offers personalized, real-time dining suggestions to help users quickly and efficiently navigate diverse culinary choices and improve their dining experiences.

## Objectives

### Main Objective

Develop an intelligent, user-friendly restaurant recommender system that provides personalized recommendations based on user location and cuisine preferences, and additional data sources to ensure accuracy and relevance.

### Specific Objectives

1. Analyze whether businesses in specific categories or locations tend to receive higher or lower average ratings and examine if a higher number of reviews correlates with more positive ratings.
2. Provide content about different cuisines and dining etiquette to enrich the user experience.
3. Create machine learning algorithms to rank restaurants based on user location, cuisine preferences, and ratings.
4. Develop an intuitive, responsive web/mobile application.

## Proposed Solution

We propose developing a Restaurant Recommender System specifically for the United States. The system will:

* **Comprehensive Mapping:** Systematically map and categorize restaurants in the United States, grouping them by cuisine type, location, and other relevant factors.
* **Personalized Recommendations**: Enable users to input their current location and preferred cuisine type, delivering a ranked list of nearby restaurants based on their ratings and proximity.
* **Enhanced User Experience:** Offer an intuitive, user-friendly interface with integrated map features to simplify navigation and help users easily explore and select dining options.

## Metrics of Success

Metrics of Success will include:

•   **User Engagement**: Measured by the number of active users, frequency of use, and user retention rates.

•   **Recommendation Accuracy:** Evaluated through user feedback and satisfaction scores, focusing on how well the recommendations match user preferences.

•   **Business Impact:** Analyzed by the increase in traffic and revenue for partnered restaurants.

•   **Scalability:** Assessed by the system’s ability to handle an expanding user base and restaurant data across various states.

## Challenges

Some of the challenges we may experience during this project include:

* **Data Collection and Quality:** Ensuring the platform has access to accurate and up-to-date restaurant data is crucial, as incomplete or outdated information can lead to poor recommendations and user dissatisfaction.
* **Scalability:** As the user base and the number of restaurants grow, the platform must efficiently handle increased data processing and maintain real-time performance across various states, ensuring a seamless user experience.
* **Competition:** Differentiating the platform from established competitors like Yelp and TripAdvisor is essential. The platform needs to offer unique value propositions to attract users in a crowded market.
* **Privacy and Data Security:** Protecting user data and ensuring compliance with privacy regulations is a significant challenge. Building trust with users regarding data usage is vital to maintaining their engagement and loyalty.

# Data Understanding

The data used to develop this project was collected from a vast Yelp database where reviews and business data was utilized. The business data contained a wide variety of businesses and was filtered to contain only the data from restaurants, coming to a total of 52,286 rows and 14 columns and stored as restaurant data. The reviews data was filtered to only include restaurant reviews and only for one year. This brought the total number of entries to 429,771 rows and 9 columns and was stored as users’ data.

The data was loaded using a dataloader class and a datainfo class to gather all the information together. A datacheck class was employed on the datasets with the following results: The restaurant data contained no duplicates or outliers but had some missing values. There were no duplicates, outliers or missing values in the users’ data.

Exploratory data analysis (EDA) was carried out on the restaurant data where the following was explored, counts of restaurants per state, frequency counts of ratings, frequency of restaurant names, distribution of star rating per state, and review counts vs stars rating.

EDA was also carried out for the users’ reviews data which showed seasonal trends with peaks around April and July and decline in reviews in between September and December. A word cloud was explored for the reviews and most common words included food, place, restaurant, order, service, good.

# Data Preparation

The users’ dataset was first prepared. There were no missing values, outliers or duplicates to handle. The columns useful, funny, cool were dropped as they did not contain any useful information for the recommendation system.

For the restaurant data, the attributes column was missing 0.8% of its dataset. These rows were dropped as attributes column is crucial for modeling. The other rows missing values i.e. address, postal code, hours were filled with Unknown as this information was not to be used in modelling and hence dropping then would result in losing out of data points. A value count of state column revealed 4 states with 1 restaurant each and these rows were dropped i.e. states containing NC, HI, XMS, MT and CO as recommending in a state with 1 restaurant will be flawed. The remaining 14 states were mapped with their full name for better understanding e.g. CA changed to California.

Feature Engineering was carried out on the categories column to explode the categories so as to filter each restaurant. The exploded column was then standardized to remove spacing between words. This was done so as to achieve a dataframe with only the specific cuisine/s in that column e.g. a row having Vietnamese, Food, Restaurants, and Food Trucks as a category was engineered to have Vietnamese as its category.

Further engineering was performed to combine city, state and address into one column called location which would be used to for modeling. The user data containing 429771 rows and 6 columns was saved in a cleaned user data csv, while the restaurant data containing 38,550 rows and 16 columns was pickled for ease of deployment.

# Modelling

The approach take to model the recommender system was split into 3:

1. Content Based Filtering
2. Collaborative Based Filtering
3. Deep Neural Networks
   1. **Content Based Filtering**

To perform content based filtering, The restaurant's features such as types of cuisine they offer and attributes such as Wi-Fi, Alcohol, Happy Hour, Noise Level, Restaurants Attire, Wheelchair Accessible, Restaurants Table Service etc., were able to provide information to use cosine similarity to recommend the restaurants with the closest similarity.

A preprocessing function was generated that combines the columns containing the categories and the attributes into one column and return a combined features columns. A create feature vectors function was also created that takes these features and vectorizes them using TfidfVectorization as it captures term importance across the document. This sparse array is then converted to an array and stacked together with the stars ratings along the horizontal columns to return a 2D array which has the combined features

A recommendation function was then created that takes in a dataframe, the state where the recommendation shall take place, name of restaurant or cuisine/category type. From where two actions would take place, either recommendation based on cuisine or restaurant name. By filtering using cuisine type, the state filters the dataframe and then these data is then filtered using the desired cuisine and the results are a recommendation based on star ratings while returning a dataframe containing the name, state, city, state, address and categories/cuisine

If the recommendation is based on restaurant name, the dataframe is preprocessed using the preprocess function from which the index of the desired restaurant is gotten and used to get the specific row of this restaurant and converted to a 1 row dataframe. A specific state dataframe is then generated by only using the restaurants from the desired state. The 1 row dataframe and the specific state dataframe are concatenated into specific state dataframe and their index reset. From which the restaurant’s new index is gotten.

The create feature vectors function is run on this specific state dataframe and the cosine similarity gotten for the combined features that were generated. The restaurant index is then used in the cosine similarity results and a similarity score is gotten. From which these scores are sorted in descending order and the top indices used to locate the recommended restaurants. The function then returns the name, state, city, address and cuisine/categories.

Testing of the recommendations show them in line with requirements i.e. while seeking cuisine recommendations within a specific state, it recommends appropriately and while seeking recommendations based on restaurant name, it gives recommendations in the chosen state and with similar attributes.

* 1. **Collaborative Based Filtering**

For the collaborative based filtering, which utilizes users specifics, business id and ratings, the user reviews data was used. This data contained the desired columns i.e. user id, business id and stars. It was merged together with the restaurant data to ensure that the business ids used in modeling were shared between both dataset.

Preprocessing for modeling was performed which involved picking the 3 desired columns i.e. user id, business id and ratings (stars) from the merged dataframe and assigned to a new dataframe i.e. new df. The surprise module was used as it is tailor made for recommendation systems. The reader class was instantiated to parse and interpret the ratings data and the dataset.load\_from\_df function was used to convert the dataframe to a format that the surprise library can interpret. From which the data was trainset, testset split.

A normal Predictor class was used as a dummy prediction model from which an RMSE of 0.819 was achieved, next an NMF with default parameters was used as it is ideal when ratings are non negative. This achieved an RMSE of 0.3489. Next an SVD model with default parameters was used as it works well with explicit feedback. This achieved an RMSE of 0.114.

From here we performed hyper parameter tuning on the SVD model using the number of epochs, regularization and latent factors. From which we achieved a much improved RMSE of 0.068. This then saved into a pickle for deployment.

Integration of the collaborative filtering involved using the collect ratings function that would filter the dataframe based on state, from which random samples would be taken. The user would provide their ratings or skip if they have never been to those restaurants. The restaurants are rated on a scale of 1-5. Once rated, the results would be appended to a ratings list as a tuple containing the restaurant id and the rating. Once the desired restaurants have been rated, the function would break.

A recommend restaurant function was also created that would take in the user id, the rated restaurant, the complete restaurant df and state. The dataframe would be filtered according to state from which the unique business ids were gotten. A new dataframe would be generated from the rated restaurants using the collect ratings function. This user rating dataframe would include the user id which would have to be input by the user. The new dataframe and the global dataframe new df are concatenated into one and filtered using the business ids that were derived previously based on the restaurants only in that state. This data is then trained in the function using SVD and the parameter we had previously achieved as we want the dataframe to keep updating each time based on the new information it receives. The unrated restaurants ids are gotten by filtering out the rated restaurants ids and these are used together with the user id to predict how a user would rate these unrated restaurants. From which the business id and predicted rating is gotten and set in a dataframe. This is then merged with the restaurant dataframe and sorted by the predicting ratings which is then returned as the recommendations.

Testing of the system involved running the collect ratings function and feeding it varying ratings on the restaurants provided. This resulted in the recommendation dataframe providing varying results based on the ratings given as well as giving a different prediction rating to these restaurants.

* 1. **Deep Neural Networks**