### **Database Systems Project Part III**

**Logical Schema Optimization and Machine Learning Model Creation**

**Team Members:** Aruzhan Bolatova, [Arya Kotibhaskar](mailto:aak10234@nyu.edu)

**Project Title:** Personalized Nutrition Recommendation System - MoodBite

### Report: Mitigating Risk in MoodBite Recipe Recommendations Using Machine Learning

**Introduction**

MoodBite is a recipe recommendation business that curates recipes based on users' moods, such as relaxing, boosting energy, or improving focus. While this personalized approach enhances user engagement, it comes with significant risks, such as:

- Recommending low-quality recipes which could lead to dissatisfaction and reduced customer retention.

- Filtering out good recipes such that they may get overlooked due to biases or inefficiencies in manual curation processes.

- In the long-term, as the recipe database scales, manually assessing recipe quality becomes infeasible.

To mitigate these risks, a machine learning model was implemented to assess recipe quality automatically based on key features like views, favorites, ratings, and user interactions. This report outlines the methodology and how the model addresses these risks effectively.

Following potential risks were identified in the project proposal level:

* Users may perceive healthy eating as expensive or unattainable.
* Users might lose interest or feel overwhelmed by recommendations.
* Users may hesitate to share sensitive health or dietary information.
* Competing apps might offer similar features.
* Poor prediction quality could lead to user dissatisfaction.

This implemented machine learning model is designed to mitigate risk of losing existing clients due to low-quality recommendations. The predictive model is continuously refined with real-time data from user interactions and external datasets, ensuring accuracy and relevance. So the accuracy of model is expected to increase with the increasing number of users and their interaction with app.

**Objective**

The primary goal of the ML model is to ensure that only high-quality recipes are recommended to users, reducing the chances of recommending low-quality or irrelevant recipes. By doing so, the model aims to enhance user satisfaction and trust in the MoodBite platform.

**Methodology**

1. Data Collection

The dataset consists of recipe details with the following features:

- Recipe Attributes such as views, favorites, spoonacular scores, average ratings, total ratings, and rating standard deviations.

- Engagement Metrics such as total likes, dislikes, and recommendation likes.

- Mood Tags associated with each recipe, such as "Relaxing" or "Boosting Energy."

The dataset was extended to include 50 recipes, ensuring initial diversity and sufficient data for training and predictions. However, in the long-term, as user interaction increases, the dataset will be increasing as well.

2. Risk Mitigation Through ML Model

Step 1: Training the Model

A Random Forest Classifier was trained on historical data, where recipes were labeled as "popular" (high quality) or "not popular" based on a threshold of views (>2). The model used the following steps:

1. Preprocessing data to handle missing values and ensure consistency.

2. Balancing class distribution using SMOTE (Synthetic Minority Oversampling Technique) to address imbalanced data.

3. Training the Random Forest model and tuning it for accuracy.

Step 2: Predicting Recipe Quality

The trained model predicts whether a recipe is "good" or "bad" based on its features. Recipes predicted as "good" are saved with a `predicted\_quality` label, allowing for further analysis and filtering.

Step 3: Filtering High-Quality Recipes

The model identifies high-quality recipes (`predicted\_quality = 1`), which are then saved separately for recommendation to users. This ensures that only recipes meeting quality standards are presented.

**Risk Mitigation Impact**

1. Reducing Low-Quality Recommendations

The model evaluates each recipe's quality using a combination of features like user engagement and ratings. This eliminates the manual biases or oversight that may result in low-quality recipes being recommended.

2. Automating the Quality Assessment Process

By automating the process of assessing recipe quality, the ML model ensures scalability and consistency. This is particularly important as the database grows, allowing MoodBite to scale operations without compromising quality.

3. Enhancing User Satisfaction

Users are more likely to engage with and trust the platform if the recommended recipes consistently meet their expectations. The ML model's ability to filter out low-quality recipes directly contributes to user retention and satisfaction.

4. Increasing Business Credibility

A consistent, high-quality user experience strengthens MoodBite's reputation as a reliable source for mood-based recipe recommendations, encouraging new users to adopt the platform.

### **Model Evaluation and Results**

The Random Forest Classifier trained for recipe quality prediction demonstrated the following performance metrics and feature importance insights:

#### **Model Performance Metrics**

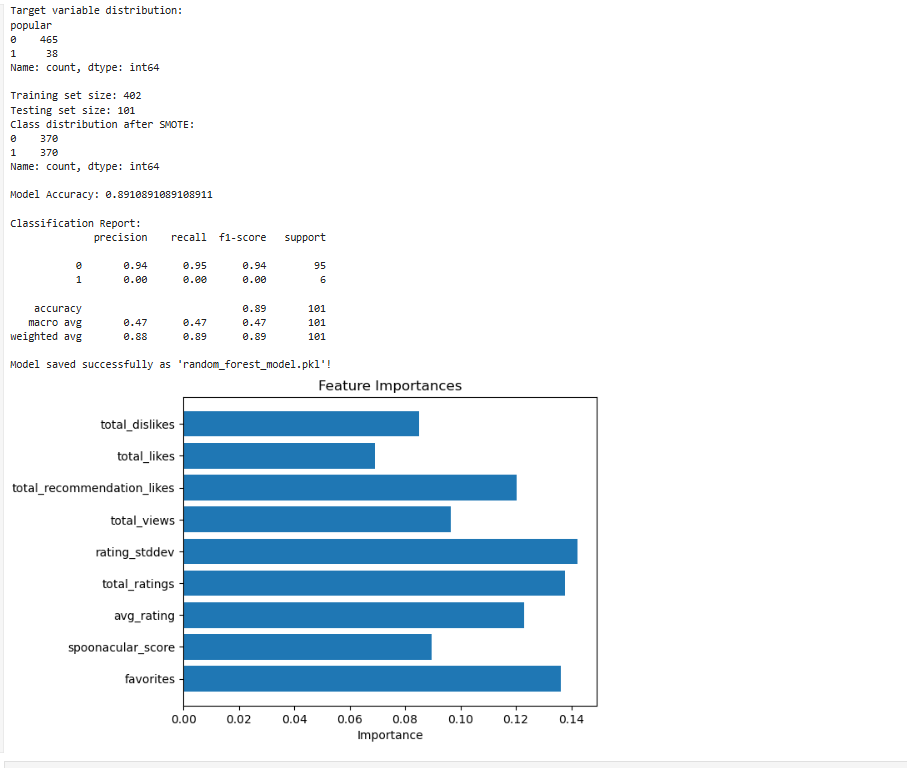
* **Accuracy**: The model achieved an accuracy of approximately 89% after addressing class imbalance with SMOTE.
* **Classification Report**:
  + Precision for the "popular" class (label 1) was **0.00**, indicating the need for additional data or fine-tuning to improve predictions for rare classes.
  + Precision and recall for the "not popular" class (label 0) were **0.94** and **0.95**, respectively, showcasing high reliability in identifying low-quality recipes.
  + The **macro average F1-score** was **0.47**, highlighting the challenges in predicting underrepresented classes.

#### **Feature Importance Analysis**

The following features significantly influenced the model's predictions:

1. **Favorites**: Most impactful, highlighting user engagement as a critical factor.
2. **Spoonacular Score**: Indicates the importance of recipe ratings in determining quality.
3. **Average Rating**: Reflects user sentiment toward a recipe.
4. **Total Ratings and Views**: Metrics showcasing recipe popularity and reach.

The feature importance chart below illustrates the relative contribution of each feature:

**

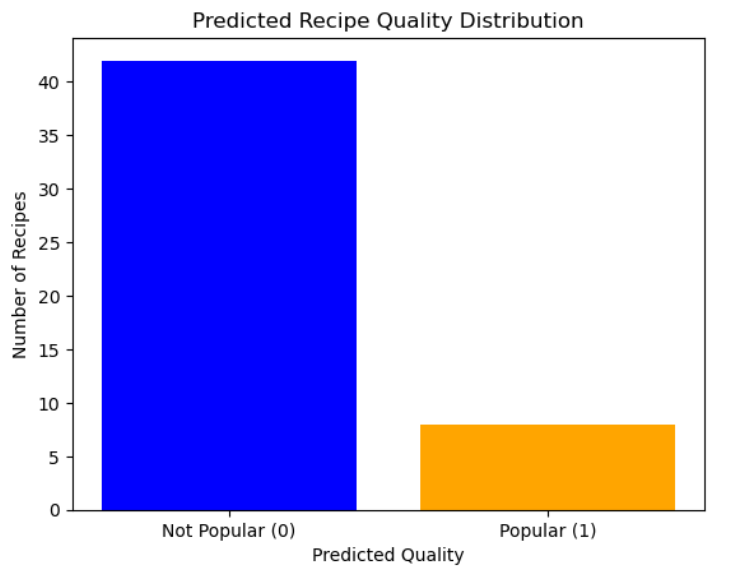
*Figure-Feature Importance.*

#### **Observations**

* Recipes with higher engagement metrics (e.g., favorites, likes) are more likely to be classified as high quality.
* Low precision for the "popular" class suggests that additional data collection and rebalancing may enhance the model's ability to identify high-quality recipes effectively.

#### **Model Deployment**

The trained model has been saved as random\_forest\_model.pkl for deployment. High-quality recipes identified by the model are stored in filtered\_good\_recipes.csv for use in recommendations.  
  
The trained Random Forest Classifier was applied to the dataset2 to predict the quality of each recipe. The model, previously saved as random\_forest\_model.pkl, was loaded using the joblib library and used to classify recipes as either "Popular" (represented by predicted\_quality = 1) or "Not Popular" (represented by predicted\_quality = 0). The predictions were then appended to the dataset as a new column, predicted\_quality, and the extended dataset, including these predictions, was saved to a new file, predicted\_recipes.csv. This output allows for a clear differentiation between high-quality and low-quality recipes based on the model’s evaluation.



*Figure- Recipe popularity.*

The primary goal of utilizing this dataset was to apply the trained model to an extended collection of recipes, ensuring that only high-quality recipes are recommended to users while also incorporating mood-related attributes into the decision-making process. This approach not only increases the relevance of recommendations but also enhances user satisfaction by offering suggestions that align with both their emotional needs and quality expectations. Additionally, this dataset bridges the gap between recipe quality evaluation and mood-specific recommendation capabilities, making it a vital part of the MoodBite platform. By leveraging this dataset, the system can identify recipes that align with user preferences and moods, ensuring that the platform delivers reliable, high-quality, and personalized recommendations.

**Results**

1. Model Accuracy: The Random Forest model achieved an accuracy of ~89% after applying SMOTE, demonstrating its ability to classify recipes effectively.

2. Feature Importance: Key features influencing recipe quality include `spoonacular\_score`, `avg\_rating`, and `total\_ratings`. These insights can guide future recipe curation and feature selection.

3. Filtered Recipes: High-quality recipes have been cached in a separate file (`filtered\_good\_recipes.csv`) for immediate use in recommendations.

**Additional Machine Learning Models for Future Implementation:**

**1. Recommendation Model Based on History of Preferences of Individual User**A more personalized recommendation model can be developed by analyzing a user’s past interactions with the system. By tracking the recipes a user has rated positively or frequently interacted with (viewed, liked, saved, etc.), the system can build a unique profile for each user. This model can use collaborative filtering techniques to predict which recipes a user would likely enjoy based on their previous choices, similar users' behavior, and recipe similarities. This would enhance the recommendation system by considering individual preferences beyond general ratings and ensuring that suggestions align more closely with the user's evolving tastes.  
**How it Works:**

* + **Data Collection**: Gather historical data on user interactions with recipes, such as ratings, likes, and frequency of recipe views. Since, there is not historic data when a user first registers, a gamification approach can be used to prompt users swipe right or left different recipes from different clusters one-by-one and therefore gain some data.
  + **Feature Engineering**: Create features that describe each user’s behavior, such as recipe types (e.g., vegetarian, keto), ingredient preferences, meal categories (e.g., breakfast, dinner), and dietary restrictions.
  + **Collaborative Filtering**: Use collaborative filtering techniques like matrix factorization (e.g., Singular Value Decomposition) or K-Nearest Neighbors (KNN) to find similarities between users and recommend recipes that similar users have liked.
  + **Content-Based Filtering**: Combine this with content-based filtering by recommending recipes similar to those that the user has previously liked (using ingredients, categories, and nutritional values as features).
* **Benefits:**
  + More tailored recommendations that adapt over time.
  + Enhanced user engagement by suggesting recipes the user is more likely to enjoy based on their personal history.

**2. Clustering System Based on User Preferences and Dietary Habits**A clustering algorithm can be used to group users with similar preferences, dietary restrictions, and health goals. By categorizing users into clusters, the system can recommend recipes that align with the general preferences of each cluster. For example, a user with a specific dietary requirement, such as gluten-free or low-carb, can be grouped with others who share the same needs, leading to more precise recipe suggestions. This model can also identify different user segments, such as those who prioritize weight loss, muscle gain, or general health, further personalizing meal suggestions.  
**How it Works:**

* + **Data Collection**: Gather data on user demographics, dietary preferences, restrictions, and health goals.
  + **Clustering Algorithm**: Implement clustering algorithms like K-Means or DBSCAN to group users with similar characteristics. These algorithms can cluster users based on their dietary preferences (e.g., vegan, gluten-free) and goals (e.g., weight loss, muscle gain).
  + **Recommendation Adjustment**: Based on a user’s cluster, the system can recommend recipes tailored to the typical preferences of that group. For instance, users in the "high-protein" cluster would receive more protein-rich recipe recommendations.
* **Benefits:**
  + More efficient grouping of users based on common needs, ensuring that recommendations align well with group preferences.
  + The ability to identify emerging trends in user behavior (e.g., new dietary trends) that could inform future recipe suggestions.

**3. Nutritional Optimization Model Based on User Goals**This model could recommend recipes that align with specific user health goals such as weight loss, muscle gain, or maintaining general health. It would analyze users’ nutritional needs based on their health metrics (e.g., BMI, activity level) and dietary preferences, providing them with balanced meals that help achieve their goals.  
**How it Works:**

* + **Data Collection**: Collect user metrics like weight, height, activity level, and health goals.
  + **Optimization Algorithm**: Use linear programming or optimization algorithms to suggest recipes that meet the user’s nutritional targets (e.g., protein, carb, fat, and calorie intake).
  + **Model Feedback**: As users interact with the system and update their goals or feedback, the model can adjust recommendations based on progress or changing targets.
* **Benefits:**
  + Tailored meal plans that not only match taste preferences but also optimize nutritional intake for health goals.
  + Flexibility in adjusting recommendations as users' goals evolve.

**4. Sentiment Analysis on Recipe Reviews for Enhanced Recommendations**Sentiment analysis can be performed on user-generated feedback (ratings and textual reviews) to better understand the emotional tone behind the feedback. For instance, even if a recipe receives a moderate rating, sentiment analysis could reveal whether users felt positively or negatively about specific ingredients or preparation methods. This will allow the system to recommend recipes not just based on numerical ratings but also on the emotional sentiment expressed by users.  
**How it Works:**

* + **Data Collection**: Gather textual feedback from users along with their numerical ratings.
  + **Sentiment Analysis**: Use Natural Language Processing (NLP) techniques, such as VADER or BERT, to analyze the sentiment of each review.
  + **Incorporation into Recommendation**: Integrate sentiment scores with the overall recipe rating to adjust recommendations. For example, a recipe with a low rating but positive sentiment in reviews could still be recommended if users express excitement or satisfaction despite the low score.
* **Benefits:**
  + Adds an emotional layer to recommendations, understanding not just if a recipe is liked, but how users truly feel about it.
  + Improves user satisfaction by considering both rational and emotional feedback

**Conclusion**

By leveraging a machine learning model, MoodBite mitigates the risks of recommending low-quality recipes while automating the quality assessment process. This ensures that users receive reliable, high-quality recipe recommendations tailored to their moods, enhancing user satisfaction, and driving business growth. As the platform evolves, the model can be further refined to maintain its competitive edge in personalized recipe recommendations.