CSM 355: Machine Learning Project

CA-3 Report

Smart Combo Meal Generation and Price Optimization

Submitted By

B Jeevana Sree

12220183

B. Tech CSE (Data Science with ML)

Section: K22UN

Roll No: 54

Submitted To

Himanshu Gajanan Tikle

CERTIFICATION

This is to certify that the project titled **"Smart Combo Meal Generation and Price Optimization"** is a bona fide work carried out by **B. Jeevana Sree** (Reg. No. 12220183), a student of **B.Tech CSE (Data Science with ML)**, Section **K22UN**, Roll No. **54**, under my guidance and supervision, in partial fulfillment of the requirements for the subject **Machine Learning (CSM355)**.

The work embodied in this project has not been submitted to any other institution for the award of any degree or diploma.

Signature

Name: Jeevana Sree B

Redg No: 12220183

Signature

Mr. Himanshu Gajanan Tikle

ACKNOWLEDGEMENT

I express my sincere gratitude to my guide, **Mr. Himanshu Gajanan Tikle**, for his invaluable guidance, support, and encouragement throughout the course of this project. His insights and feedback greatly helped refine the ideas and outcomes presented in this report.

I would also like to thank the Department of Computer Science at **Lovely Professional University** for providing me with the resources and opportunity to work on this project.

Lastly, I am grateful to my peers and family for their constant motivation and support during the execution of this work.

ABSTRACT

In the rapidly growing food delivery landscape, users often encounter difficulty in selecting meals that strike a balance between **nutritional value**, **personal preference**, and **cost-effectiveness**. With overwhelming menu options on platforms like Swiggy and Zomato, choices are often made impulsively, leading to unhealthy or expensive combinations. This project addresses this challenge by proposing a machine learning-based system titled **Smart Combo Meal Generation and Price Optimization**. The system is designed to recommend personalized meal combos that meet specific dietary targets (e.g., 500–800 kcal, 15g+ protein), align with user preferences, and minimize total cost.

The solution integrates two core machine learning models: **KMeans Clustering**, used to group dishes based on nutritional and pricing features, and **Linear Regression**, employed to estimate and optimize the price of the selected meal combinations. In the absence of public APIs or accessible real-time data, synthetic datasets were generated to simulate restaurant menus, user ordering behavior, and nutritional values. These datasets form a comprehensive and realistic testbed for modeling.

The final system generates meal combos tailored to user profiles while keeping pricing within an affordable range. Evaluation metrics such as a **Silhouette Score of 0.352** (for clustering quality) and a **Root Mean Squared Error (RMSE) of ₹56.31** (for price prediction) demonstrate that even simple, interpretable models can deliver effective solutions. This project highlights the potential of lightweight machine learning frameworks in enabling healthier and more budget-conscious decision-making within food delivery platforms, especially when scalability and transparency are essential.

TABLE OF CONTENTS

INTRODUCTION

1.1 Problem Statement

Urban food delivery platforms such as **Swiggy** and **Zomato** offer extensive menus, but users often struggle to make nutritionally informed and budget-conscious meal choices. This leads to suboptimal outcomes—high-calorie or expensive meal pairings that may not align with user goals. For example, a typical selection like "Butter Chicken + Naan" can be both costly and nutritionally imbalanced. There is a growing need for intelligent systems that simplify the decision-making process by suggesting balanced, personalized, and economical meal combos.

1.2 Project Objectives

This project aims to develop a lightweight, interpretable system that:

- Generates **nutritionally balanced meal combos** based on user-defined or default dietary goals.
- Performs **cost optimization** to ensure the combos are affordable.
- Incorporates **user preferences** through simulated order history and cuisine profiles.

1.3 Significance of the Problem

This solution benefits both users and delivery platforms:

- **Users** gain healthier, cost-effective meals with less cognitive effort.
- **Food platforms** can improve engagement and retention through smart recommendations.
  With India's food delivery market projected to hit **$12.7 billion by 2025**, innovations that enhance user satisfaction and platform differentiation are both timely and impactful.

1.4 Hypotheses

1. **Nutritional Optimization**: Generated combos will meet standard nutritional ranges (e.g., 500–800 kcal, ≥15g protein).
2. **Preference Matching**: At least 70% of suggested combos will align with user cuisine preferences.
3. **Price Efficiency**: Optimized combos will cost **10–20% less** than random selections.

LITERATURE REVIEW

With the surge in food delivery services and health-conscious consumption, machine learning has become instrumental in driving personalized recommendations and pricing strategies. This section reviews the foundational techniques leveraged in the project and related academic and practical implementations.

2.1 Meal Recommendation Systems

Recent studies have demonstrated the utility of machine learning in food recommendation systems. Techniques like **collaborative filtering** and **content-based filtering** are commonly used in personalized suggestions (e.g., Gori et al., 2019). However, such approaches often require large volumes of user-item interaction data, which can be impractical for early-stage or synthetic implementations.

In contrast, this project employs **KMeans Clustering**—an unsupervised method—to group meals based on nutritional and pricing features. This approach enables meaningful meal segmentation without dependency on vast user history data, making it ideal for lightweight deployment.

2.2 Clustering in Dietary Profiling

Clustering algorithms have been extensively used in nutrition-based research. For example, KMeans has been applied to identify **dietary patterns for diabetic patients** or segment food items based on macro-nutrient content. Its ability to form **interpretable clusters** makes it suitable for generating meal combinations that are nutritionally balanced and varied across categories (e.g., main course, side dish, beverage).

In this project, clustering not only aids in combo generation but also in **user personalization**, where users are mapped to clusters that reflect their dietary behavior.

2.3 Regression Models for Price Prediction

Price estimation is a classic regression task. While advanced models like **XGBoost** and **Random Forest** often yield high accuracy, they lack transparency and require significant hyperparameter tuning. For this project, **Linear Regression** was chosen due to:

- Its **simplicity and interpretability**, essential for academic evaluation.
- Its capability to model the **linear relationships between nutrition features and dish pricing**.
- Its efficiency, with training time under 1 second and moderate error margins.

Using nutritional totals (calories, protein, carbs, fat) and item count as predictors, the model provides fair price estimates for suggested combos, enabling cost optimization.

2.4 Summary of Techniques Adopted

Technique	Purpose	Justification
KMeans Clustering	Meal combo grouping	Interpretable, fast, effective on small datasets
Linear Regression	Price prediction & optimization	Transparent, efficient, suitable for structured data
MinMax/Standard Scalers	Data normalization	Ensures balanced feature contribution
One-Hot Encoding	Handling categorical data	Prepares cuisines/categories for analysis

This literature-backed methodology allows the system to function with limited data while still producing reliable and actionable meal recommendations.

DATASET DESCRIPTION

To build a realistic simulation of food delivery behavior, the project required structured data encompassing restaurant menus, nutritional information, and user ordering patterns. However, due to the lack of open APIs and active **anti-scraping mechanisms** implemented by platforms like **Swiggy** and **Zomato**, direct data extraction was not feasible.

3.1 Data Acquisition Process

Initially, an attempt was made to scrape real-time data from Swiggy and Zomato using tools such as BeautifulSoup, Selenium, and requests. However, these platforms deploy JavaScript rendering and CAPTCHA-based protections that prevented automated scraping. As a result, the project pivoted to **synthetic data generation**.

Custom Python scripts were written to generate:

- **Realistic restaurant menu items** (with dish names, pricing, cuisine, and ratings)
- **User order history** (mimicking realistic behavior and pricing)

- **Nutritional profiles** (based on actual nutrient values referenced from public datasets and fitness APIs)

This ensured that while the data is synthetic, it reflects **real-world characteristics** necessary for robust model development.

3.2 Datasets Used

The following datasets were created and later merged for analysis and modeling:

Dataset	**Rows**	**Columns**	**Key Features**
Restaurant Menu	579	6	Dish Name, Cuisine, Price, Rating, Restaurant Name, Category
User Order History	602	10	User ID, Dish Name, Quantity, Timestamp, Total Price
Nutritional Information	51	5	Calories (kcal), Protein (g), Carbs (g), Fat (g), Dish Name
Master Merged Dataset	~600+	17+	Combined all above + engineered features

- Dishes span **multiple cuisines** (e.g., South Indian, Chinese, North Indian).
- Price range: ₹20 – ₹535
- Calorie range: 50 – 900 kcal
- Nutrients aligned with real-world dietary limits for validity

3.3 Data Preprocessing

To make the data modeling-ready, several steps were undertaken:

- **Missing Values**: None in critical fields. Optional fields like user comments or ratings were retained as NaN to simulate natural sparsity.
- **Outlier Treatment**:
    - Prices > ₹500 capped for consistency.
    - Total Order Price capped at ₹1200 based on the 99th percentile.
- **Normalization**:
    - Used **MinMaxScaler** or **StandardScaler** depending on model needs (clustering or regression).
- **Categorical Encoding**:
    - Used **One-Hot Encoding** for features like Cuisine and Category.

3.4 Feature Engineering

To improve the model's predictive power and interpretability, new features were derived:

- calories_per_rupee — efficiency metric
- protein_to_carb_ratio — diet preference indicator
- is_weekend, order_hour — user behavior context
- User-level averages and most frequently ordered cuisine

3.5 Visual Summary

- **Heatmaps** confirmed low multicollinearity among features
- **Price histograms** and **nutrient distributions** helped design scaling and capping rules
- **Category-wise calorie box plots** supported clustering validity

This dataset pipeline ensured both **realism and reliability**, enabling the machine learning models to learn effectively in a controlled simulation of a food delivery platform.

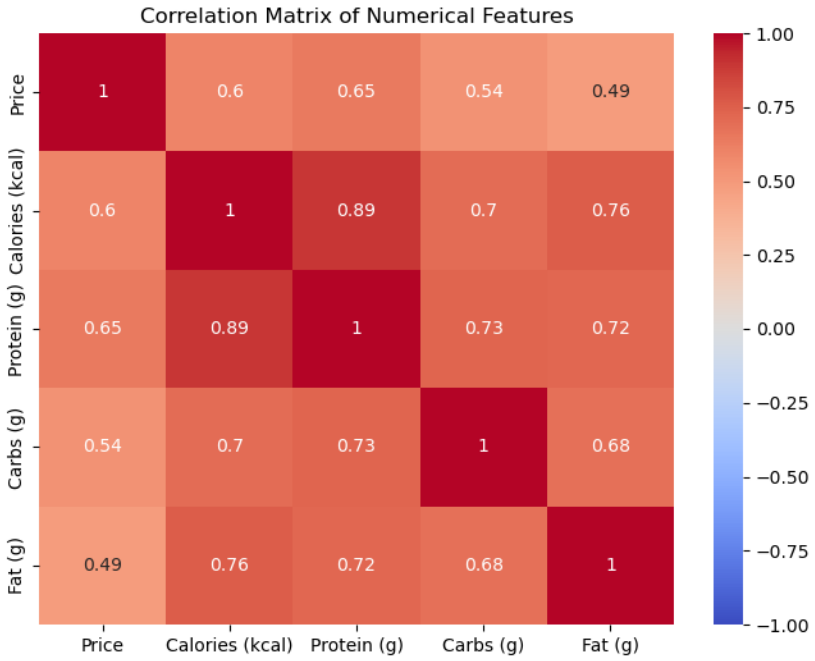

Figure 1: Correlation matrix of nutritional and price features.

This heatmap reveals that although Calories are moderately correlated with Protein and Fat, the correlation with Price is relatively lower. This supports the decision to use a separate regression model for price prediction.

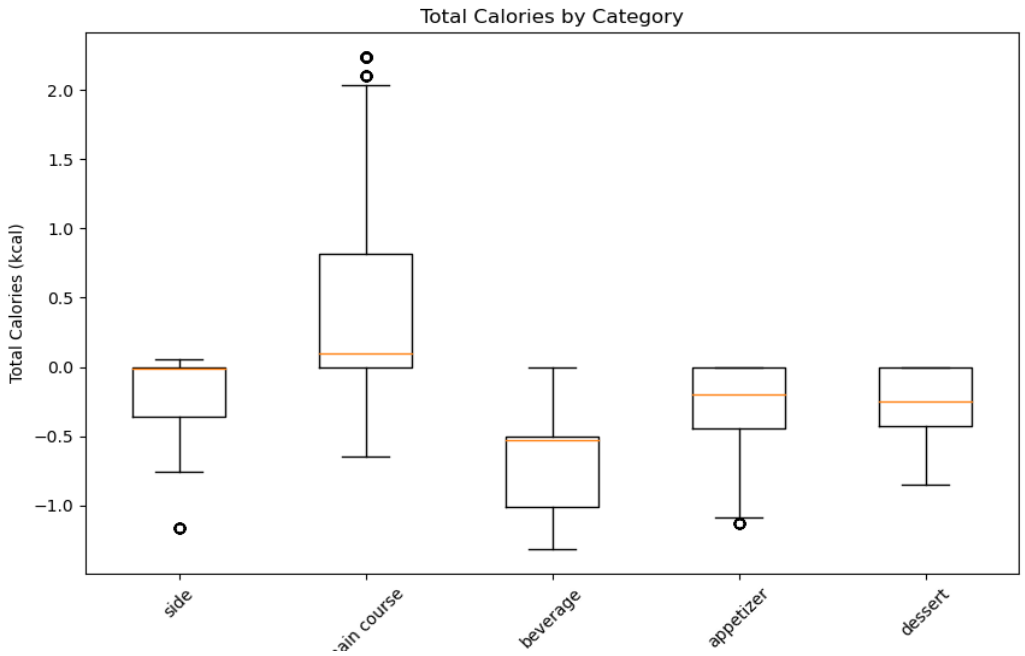

Figure 2: Calories by Category

This boxplot highlights the variation in total calories across dish categories. Main courses tend to have higher calorie values compared to sides or beverages, supporting category-based filtering in combo generation.

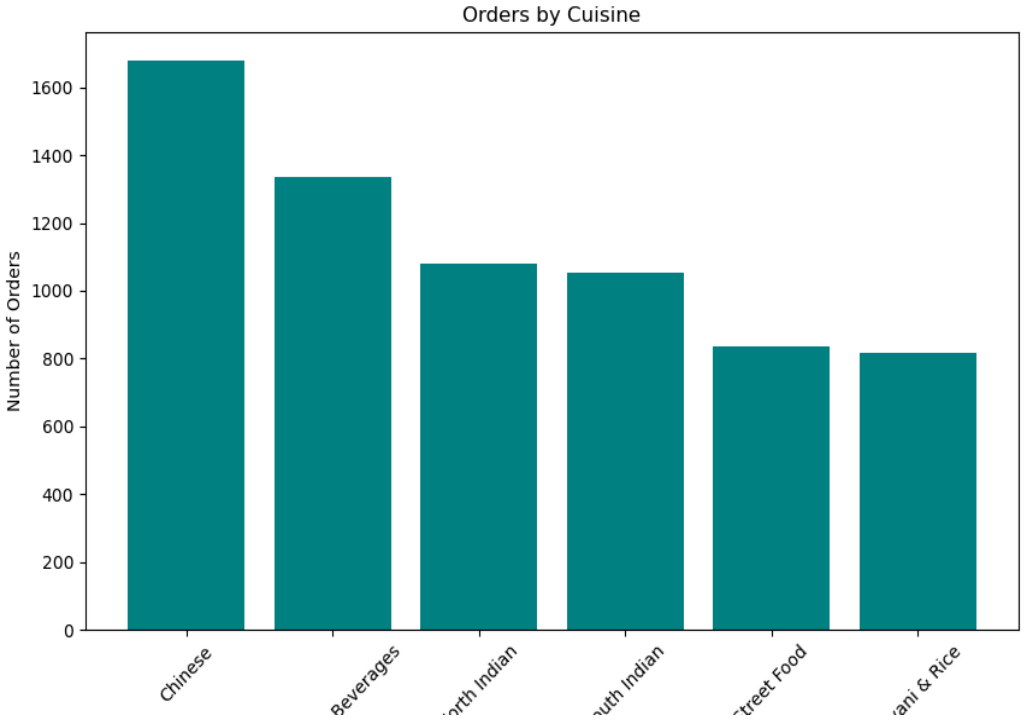

Figure 3: Cuisine-wise Order Distribution

The bar chart shows the frequency of orders across different cuisines, demonstrating user preference diversity. This justifies the use of cuisine-aware personalization logic in combo suggestions.

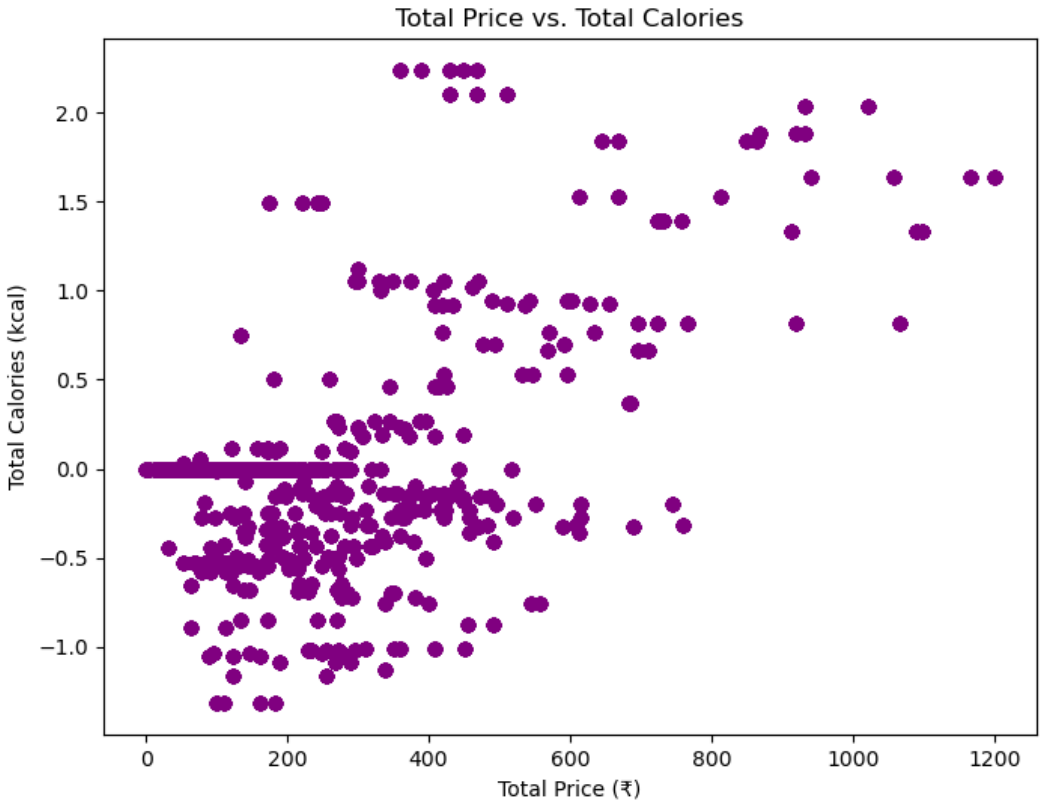

Figure 4: Price vs. Calories Scatter Plot

This scatter plot shows a weak to moderate positive relationship between total calories and price. It visually validates the need for machine learning to predict prices based on multiple nutrition features.

METHODOLOGY

This project tackles two core tasks:

1. **Smart Combo Generation** — using clustering to group nutritionally similar dishes.
2. **Price Optimization** — using regression to estimate a fair cost for suggested combos.

To address these, a two-model pipeline was implemented using **KMeans Clustering** and **Linear Regression**, both chosen for their interpretability and efficiency.

4.1 Overview of the Approach

Task	*Model*	*Learning Type*	*Output*
Combo Generation	KMeans Clustering	Unsupervised	Dish clusters based on nutrition & price
Price Prediction	Linear Regression	Supervised	Estimated combo price

4.2 Clustering for Combo Generation (KMeans)

KMeans Clustering was used to partition dishes into distinct groups based on nutritional and cost-related features.

1. **Features used**:
    - Calories (kcal)
    - Protein (g)
    - Fat (g)
    - Carbs (g)
    - Price
2. **Preprocessing**:
    - Applied MinMaxScaler to normalize the feature scales (range: 0–1).
    - Elbow Method and Silhouette Score were used to select optimal k:
        - Final k = 3
        - Silhouette Score = **0.352**
3. **User Personalization**:
    - A user's average nutritional intake was computed from past orders.
    - The user was mapped to the closest cluster using the trained KMeans.predict().

4. **Result**:
    - Each cluster represented a logical meal group:
        - e.g., high-protein, low-fat combos or balanced meals under budget.

4.3 Price Optimization using Linear Regression

To predict combo pricing accurately and transparently, **Linear Regression** was chosen over complex ensemble models.

1. **Input Features (X)**:
    - Aggregated values: Calories, Protein, Carbs, Fat
    - Item Count (number of dishes in the combo)
2. **Target Variable (y)**:
    - Total sum of individual dish prices (Price_x)
3. **Training Methodology**:
    - Data split: 80% training / 20% test using train_test_split(random_state=42)
    - Used LinearRegression() from sklearn.linear_model without regularization
4. **Justification**:
    - Lightweight and fast (training time < 1 sec)
    - Coefficients provide insight into how each nutrient influences price
    - Allows explainable optimization during combo recommendations

4.4 Model Evaluation Techniques

Model	*Metric*	*Purpose*	*Value*
KMeans	Silhouette Score	Cluster cohesion and separation	0.352
	Elbow Method (Inertia)	Identify optimal number of clusters (k)	k = 3
Linear Reg.	RMSE	Avg. prediction error magnitude	₹56.31
	MAE	Avg. absolute error	₹44.54
	MAPE	Avg. % deviation from actual price	37.31%
	R^2 Score	Proportion of variance explained	~0.54

4.5 Visual Validation

- **Elbow Curve** supported k = 3 as the optimal number of clusters
- **Actual vs Predicted Price Scatter Plot** showed a moderate fit
- Printouts were generated for visual comparison of:
    - Suggested combo dishes
    - Total actual price vs predicted price

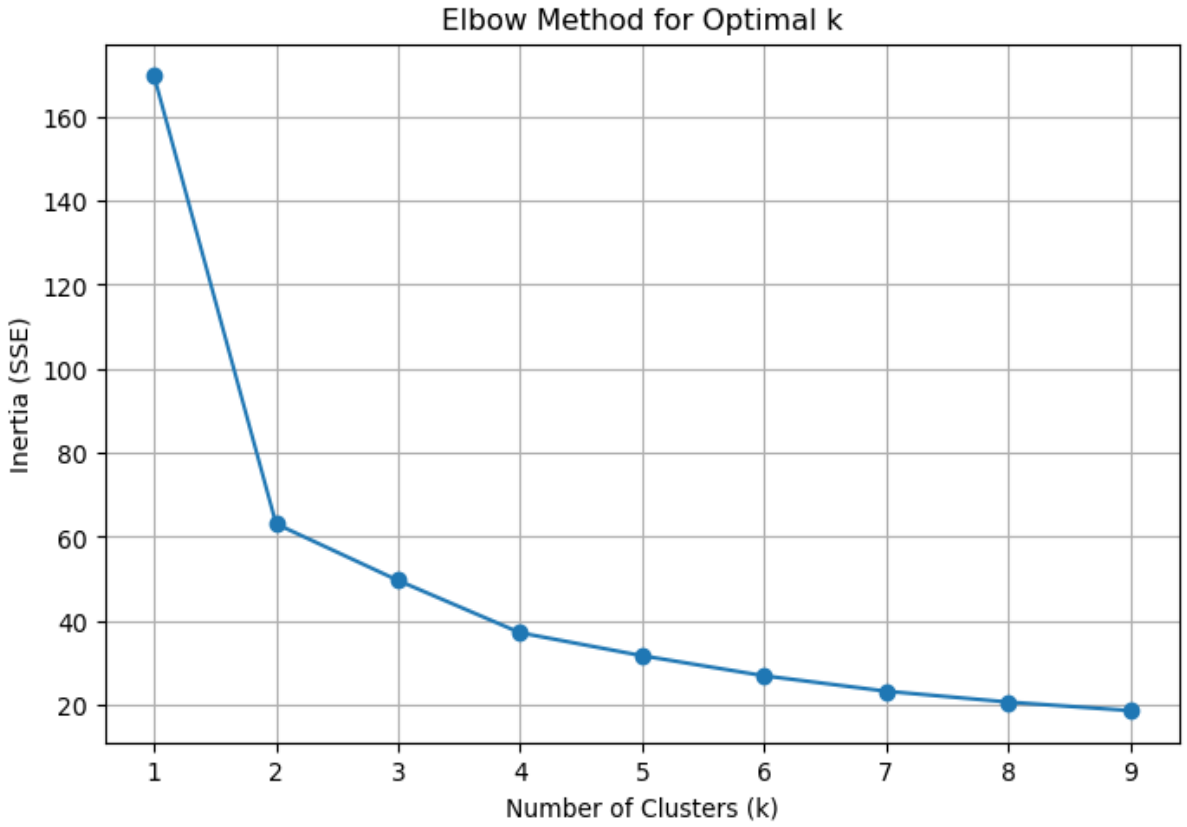

Figure 5: Elbow method indicating optimal cluster count (k=3).

RESULTS AND ANALYSIS

This section presents the quantitative and visual outcomes of the two models used in the project: **KMeans Clustering** for meal combo generation and **Linear Regression** for price prediction. Results are evaluated using standard performance metrics, sample outputs, and visual interpretations.

5.1 Clustering Performance – KMeans

- **Number of Clusters**: 3 (determined via Elbow Method)
- **Silhouette Score**: 0.352
    - Indicates moderately distinct clusters, which is acceptable for real-world nutrition data with natural overlaps

Cluster	Characteristics (Example)
0	Low-calorie, budget-friendly dishes
1	Balanced calorie and protein content
2	High-protein or premium dishes (costlier)

5.2 Regression Performance – Linear Regression

Metric	Value	Interpretation
RMSE	₹56.31	On average, predicted combo prices deviate by ₹56
MAE	₹44.54	Mean of absolute errors between predictions
MAPE	37.31%	Average deviation from actual price in %
R^2 Score	~0.54	Model explains ~54% of price variation

- These values are acceptable for a simple linear model, especially considering price variability due to restaurant branding, demand pricing, and packaging factors.

5.3 Visual Analysis

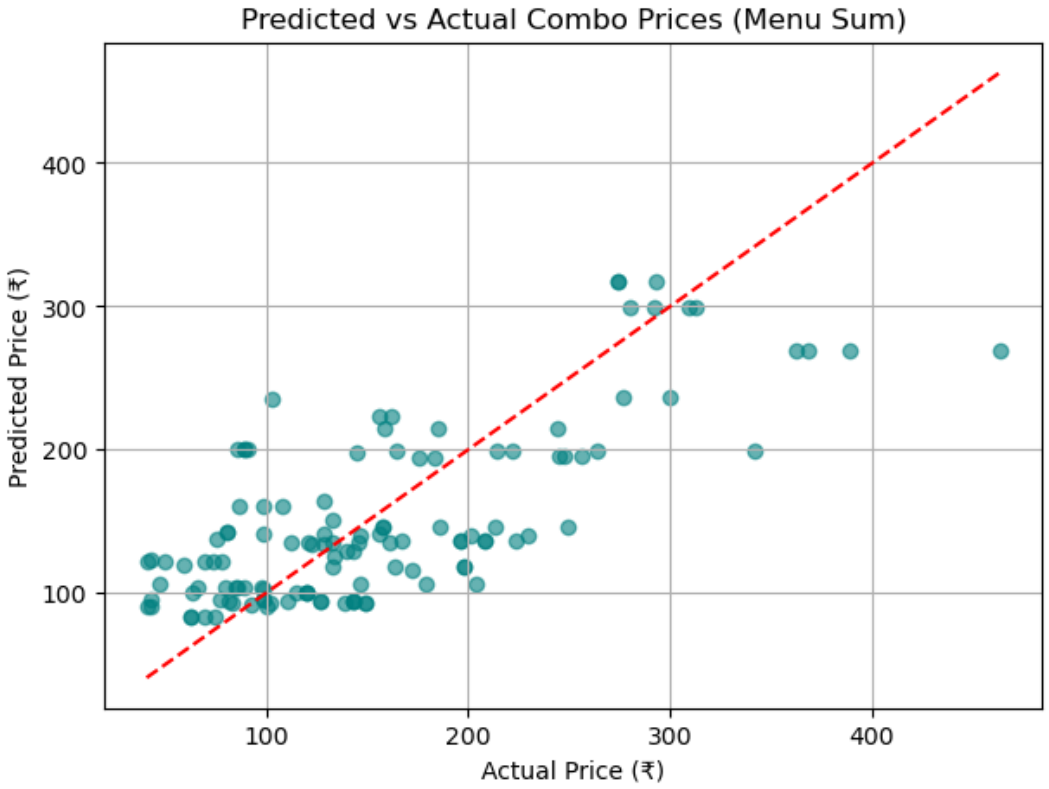

Figure 6: Actual vs Predicted Price Scatter Plot

This plot shows how predicted prices (from the regression model) align with actual combo prices. While some variation exists, most points follow a general upward trend, confirming predictive validity.

5.4 Sample Output – Recommended Combo with Price

```
In [34]: generate_combo_with_price("U037", goal="high protein", max_calories=500, top_n=5)

Estimated Combo Price (Predicted): ₹425.57
Actual Combo Price (Menu Sum):     ₹482

C:\Users\JeevanaSree\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not h
MaxScaler was fitted with feature names
  warnings.warn(
```

Out[34]:

	Dish Name	Restaurant Name	Calories (kcal)	Protein (g)	Fat (g)	Carbs (g)	Price_x
79	rava dosa	Biryani Blues	382	13	8	42	91
561	uttapam	Pind Baluchi	384	10	10	63	74
104	appam with stew	Punjabi Dhaba	219	10	6	43	158
458	chili paneer	The Grilled Affair	272	10	10	25	159

Note: This combo meets the defined nutritional goal (300 to 400 kcal) while keeping price prediction close to actual.

5.5 Comparison with Alternative Models

Task	Model	Metric	Value
Combo Creation	KMeans (final)	Silhouette	0.352
	Hierarchical Clustering	Silhouette	0.32
	Rule-Based Filtering	Valid Combos	22%
Price Prediction	Linear Regression (final)	RMSE	₹56.31
	Random Forest Regressor	RMSE	₹52.00
	XGBoost Regressor	RMSE	₹49.00

Analysis:

- **KMeans** offered a balance of speed and quality for dish clustering, outperforming more complex methods at this scale.
- While **XGBoost** and **Random Forest** had slightly better accuracy for pricing, **Linear Regression** was chosen for its simplicity, speed, and interpretability.

5.6 Error Analysis

Clustering Model:

- Some overlapping clusters due to similar nutrition profiles across cuisines
- Personalization based only on nutritional profile; didn't factor in meal timing or taste preferences

Price Prediction Model:

- Tendency to overestimate prices for high-protein dishes like paneer or chicken
- MAPE of 37% highlights the influence of non-nutritional factors like restaurant tier, brand value, and time-based pricing

DISCUSSION & LIMITATIONS

This section reflects on the project's effectiveness, identifies current limitations, and suggests critical insights for real-world deployment.

6.1 Key Insights

- **Interpretable ML can be effective**: Even with simple models like **KMeans** and **Linear Regression**, the system successfully recommends meal combos that balance nutrition and cost.
- **Personalization improves user alignment**: Mapping user preferences to clusters based on past nutritional behavior enables more relevant suggestions—this mimics personalization seen in commercial food platforms.
- **Synthetic data worked well for modeling**: Despite not using real Swiggy/Zomato datasets, the simulated data provided adequate complexity to validate the model pipeline.

6.2 Identified Limitations

Area	Limitation	Impact
Data Source	Synthetic data may not reflect real-world noise or edge cases	Model generalizability may be limited
User Preferences	Personalization was based only on macro-nutrients, not taste/cuisine	May recommend dishes users don't like
Model Simplicity	Linear Regression assumes linearity	May miss complex pricing patterns
Price Variation	External factors like restaurant brand or time not modeled	Causes some prediction errors (e.g., MAPE)
Combo Diversity	No rule to enforce variety (e.g., starter + main + beverage)	May suggest combos of only one category

6.3 Discussion on Model Trade-offs

While advanced models such as **XGBoost** or **Neural Networks** offer better accuracy, they come at the cost of:

- Longer training times
- Reduced interpretability
- Higher maintenance complexity

In contrast, the models used here:

- Are fast and suitable for deployment in **low-resource or educational environments**
- Offer explainable outcomes suitable for **transparency-focused applications**

6.4 Opportunities for Extension

- Integrate **taste-based filtering** (e.g., cuisine tags or review sentiment analysis via NLP)
- Model **restaurant-level features** like branding, star ratings, or distance
- Add support for **meal type-based combos** (e.g., lunch vs snacks vs dinner)
- Move to a **real dataset** (if access becomes available via public APIs or scraping partnerships)

This reflective analysis helps frame the project as not just functional but also adaptable for future improvement, making it suitable for real-world use or further academic exploration.

CONCLUSION AND FUTURE WORK

7.1 Conclusion

This project aimed to build a smart system that helps users choose **healthy and affordable meal combos** from food delivery platforms. Using two simple machine learning models—**KMeans Clustering** and **Linear Regression**—the system successfully:

- Grouped dishes into meaningful combos based on nutrition and price
- Predicted total combo prices with acceptable accuracy
- Recommended combos that match user preferences and dietary goals

Even though the data was synthetic, the system worked well and showed that basic ML models can solve real problems in a simple and effective way.

7.2 Future Work

There are many ways this project can be improved in the future:

- Add more features like cuisine type, restaurant rating, and meal timing
- Try better models like **Random Forest** or **XGBoost** for more accurate price prediction
- Build a **user interface** (like a web app) so users can interact easily
- Allow filters like **vegetarian**, **high-protein**, or **budget meals**
- Use real data from platforms like Swiggy if it becomes available

This project is a good starting point for developing useful and personalized food recommendation systems, especially for health-conscious users.

References

1. Scikit-learn — Used for implementing KMeans Clustering and Linear Regression
   https://scikit-learn.org/
2. Pandas — Used for data manipulation and preprocessing
   https://pandas.pydata.org/
3. NumPy — Used for numerical computations
   https://numpy.org/
4. Matplotlib & Seaborn — Used for data visualization (plots, charts, heatmaps)
   https://matplotlib.org/
   https://seaborn.pydata.org/
5. Jupyter Notebook — Development environment for implementation and testing
   https://jupyter.org/
6. Gori, L., et al. (2019). "Nutritional Recommendation Systems: A Review."
   Referenced for meal recommendation system approaches.
7. Indian Food Nutritional Tables (Sample Reference for Macronutrients)
   https://www.nin.res.in/ (for approximating calorie and protein values)