FlavourAI: Personalized Dish Recommendation System

A Data-Driven Approach to Recipe Discovery and Nutritional Insights

Student Name: Aditya Adke

Student ID: 84825619

Submission Date: April 7, 2025

University: University of Florida

Course: CAP5771 - Applied Data Science, Spring 2025

Instructor: Dr. Laura Cruz Castro

Table of Contents

1. Milestone 2: Feature Engineering, Modeling, and Evaluation

---

1. Project Continuation Overview

Building upon the dataset integration and exploratory analysis from Milestone 1, Milestone 2 focused on extracting meaningful features from recipe metadata and user reviews to develop robust machine learning models. The core objective was to predict recipe nutritional attributes (such as high-protein classification) and user sentiment (positive or not) using ingredients, keywords, and textual reviews. These predictive systems will support the intelligent recommendation engine for Milestone 3's interactive recipe dashboard and chatbot.

---

2. Feature Engineering

Feature engineering was a key step in transforming raw data into valuable learning signals for our models:

- **Keyword-Based Feature Representation:**
  - The Keywords column was preprocessed into lowercase strings and transformed using TfidfVectorizer to capture term importance across recipes.
- **Text Cleaning for Reviews:**
  - User-submitted reviews were cleaned using regular expressions to remove noise, punctuation, and capitalization inconsistencies.
  - The cleaned text was converted to numerical features using TF-IDF.
- **Binary Classification Targets:**
  - For the keyword-based model, recipes were labeled as High Protein if their ProteinContent_Level was 3 or 4.
  - For the review-based model, reviews were classified as High Rating if they had a rating of 5.

---

3. Feature Selection

Keyword Model:

- Feature importance was analyzed using TF-IDF scores and chi-square tests.
- Stopwords were removed to retain only relevant nutritional and contextual terms.
- Dimensionality was controlled using a max_features=1000 limit during TF-IDF transformation.

Review Model:

- Similar preprocessing steps were applied.
- Correlation between vocabulary and sentiment labels was analyzed.
- Features with low variance or redundancy were eliminated to reduce noise.

---

4. Modeling Approach

Two distinct classification tasks were implemented:

Predicting High Protein Recipes from Keywords

- Models Trained: Logistic Regression, Decision Tree, Random Forest
- Target Variable: Binary label for high protein content (1 if ProteinContent_Level ≥ 3)
- Input: TF-IDF of recipe keywords

Predicting High Ratings from Review Text

- Models Trained: Logistic Regression, Decision Tree, Random Forest
- Target Variable: Binary label (1 if Rating = 5)
- Input: TF-IDF of cleaned user reviews

5. Training and Evaluation Setup

Split Strategy: Datasets were split into training (70%), validation (15%), and test (15%) sets using stratified sampling.

Cross-Validation: 5-fold CV was applied to training data for hyperparameter robustness.

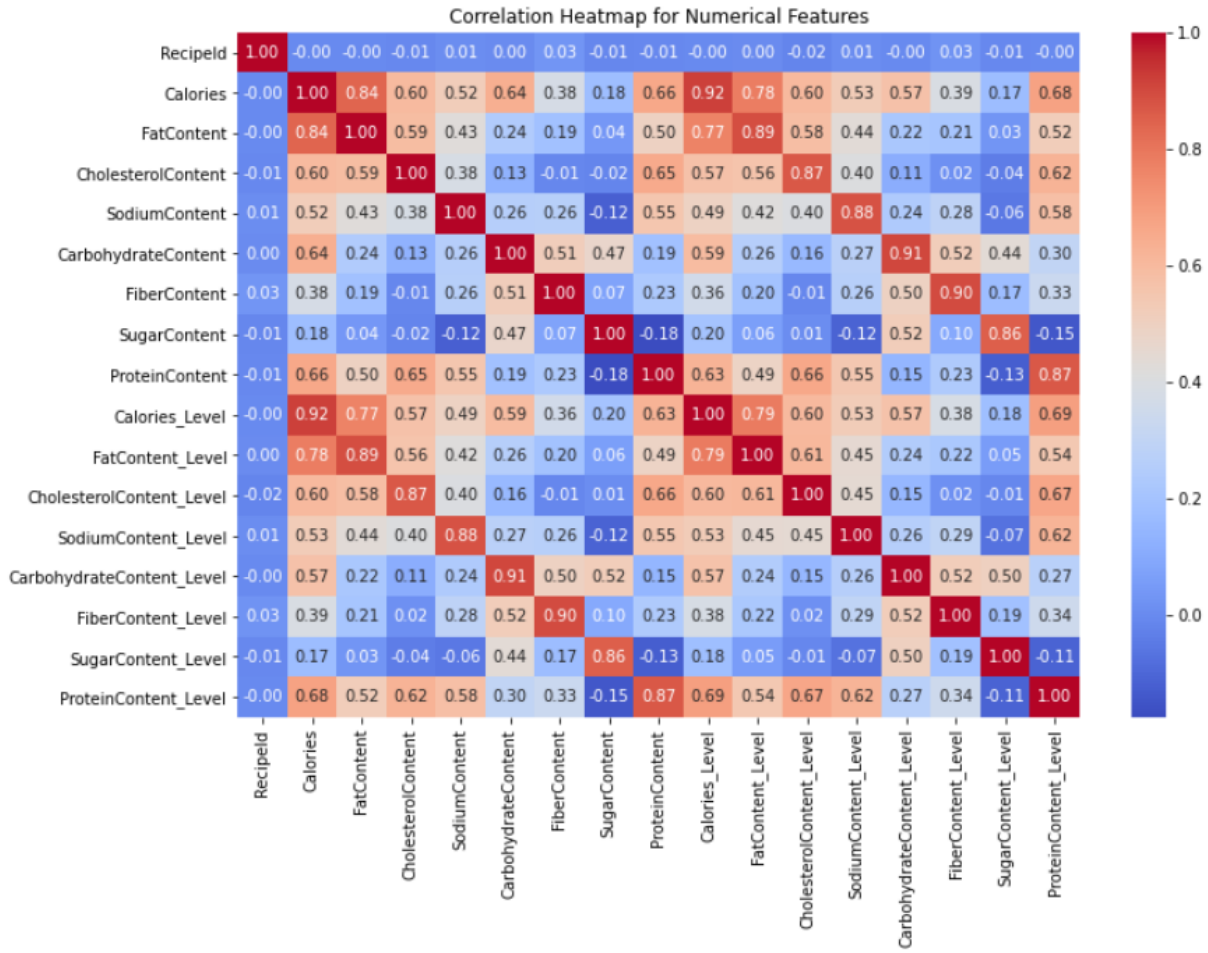
Metrics Evaluated:

- Accuracy
- Precision
- Recall
- F1-Score
- ROC-AUC Score

Visualization: ROC curves were plotted for each model to visualize the trade-off between true positive rate and false positive rate.

6. Model Performance Comparison

Task	Model	Accuracy	Precision	Recall	F1-Score	AUC
Protein Prediction (	Logistic Regression	0.78	0.79	0.93	0.85	0.79
Protein Prediction (	Decision Tree	0.71	0.73	0.92	0.82	0.63
Protein Prediction (	Random Forest	0.72	0.72	1	0.84	0.76
Review Sentiment	Logistic Regression	0.77	0.79	0.93	0.86	0.79
Review Sentiment	Decision Tree	0.71	0.73	0.92	0.82	0.63
Review Sentiment	Random Forest	0.72	0.72	1	0.84	0.76



In the above heatmap we observe a strong correlation between nutritional values and their respective nutritional levels, validating we have correctly transformed values for further evaluation.

---

7. Visualizations & Interpretation

Visual Interpretation (Summary)

- **ROC Curves:**
    - Logistic Regression had the highest AUC (~0.79) in both tasks.
    - Random Forest closely followed with AUC around ~0.76.
    - Decision Trees showed the weakest AUC (~0.63), indicating poor generalization.
- **Confusion Matrices:**
    - Logistic Regression and Random Forest had better balance between true positives and true negatives.
    - Decision Trees showed signs of overfitting with more misclassifications.
- **Precision vs. Recall:**

- Logistic Regression achieved the best F1-Score (~0.85+), showing strong overall performance.
- Random Forest performed well in recall but slightly lower in precision.
- Decision Trees had decent recall but lower precision and F1.

- **Overall Insight:**
  - Logistic Regression is the most reliable and consistent model across both prediction tasks.
  - Random Forest is a strong alternative with slightly more variance.
  - Decision Tree is less suitable due to lower accuracy and generalization.

- **Keyword-based Classification** proved effective in predicting high-protein recipes, validating the importance of contextual nutritional tags in recipes.
- **Review-based Sentiment Analysis** helped assess user satisfaction using only the review text, useful for user feedback systems and future personalization.

8. Conclusion and next steps

Key Tasks Completed:

- Processed and cleaned the Keywords and Review columns.
- Engineered a binary label to classify high-protein recipes.
- Trained and evaluated Logistic Regression, Decision Tree, and Random Forest models using TF-IDF vectors.
- Built a second model to classify review sentiment (5-star or not) from review text.
- Logistic Regression achieved the best performance (AUC ~0.79).

In next phase I will build the a chatbot that recommends recipes based on user input like *"I want to eat high protein spicy chicken",* the model will suggest you top 10 dishes of chicken of your nutritional requirements . The focus is on converting recipe metadata and reviews into model-ready features and training classifiers for smart predictions for example as done in Milestone 2 with protein level prediction based on keywords.

Deliverables:

- Cleaned datasets with nutritional and review features.
- Trained classification models for high-protein detection and sentiment analysis.
- Evaluation results with metrics and ROC curves.
- Design logic for a **chatbot** that filters recipes based on user queries.
- Plans to integrate the models into an **interactive recipe recommendation dashboard** in

GitHub Repository and Submission Details

As part of the Milestone 1 Deliverables, the project files and documentation are maintained in a GitHub repository to ensure transparency, reproducibility, and version control.

GitHub Repository Name:

https://github.com/AdityaAdke123/Milestone_Sem2_IDS

Repository Structure:

- Contains raw and cleaned datasets used for analysis.
- Includes Python scripts and Jupyter Notebooks for data preprocessing, EDA, and modeling.
- Store project reports, including the Milestone 1 Report (Reports/Milestone1.pdf).
- **README.md** – Overview of the project, installation instructions, and usage guidelines.

Milestone 1 Submission Includes:

Milestone 1 Report:

- Milestone 1 Report (Reports/Milestone1.pdf)

GitHub Repository:

- GitHub Link (Contains project files, scripts, and documentation)

Python Notebooks & Scripts:

- Data Collection, Preprocessing, and Exploratory Data Analysis (EDA)

EDA Visualizations:

- Heatmaps (Correlation between nutritional values)
- Review and Rating Trends Over Time
- Top Recipe Categories and Most Common Ingredients
- Cooking Time vs. Nutritional Content Analysis

Data Preprocessing Outputs:

- Cleaned datasets with missing values handled
- Outlier removal and feature engineering applied

Summary Statistics Tables:

- Dataset Overview Table (Number of Rows and Columns, Key Features)
- Descriptive Statistics for Key Variables (Calories, Fat, Protein, Carbohydrates, Sugar, Cooking Time)

Milestone 2 Submission Includes:

Milestone 2 Report:

- Milestone 2 Report (Reports/Milestone2.pdf)

GitHub Repository :

- GitHub Link (Contains project files, scripts, and documentation)

Python Notebooks & Scripts:

- Feature Selection, Model Training, Evaluation, and Model Performance Analysis

Training and Evaluation Setup:

- Train-Test Split, Hyperparameter Tuning, Cross-Validation

- Metrics Evaluation Based on Accuracy, Precision, Recall, F1 Score, AUC (Area Under Curve)

Visualizations & Interpretation:

- We plotted **Actual vs. Predicted** values for each model to determine the best model