A Mini Project Report on

Predictive Modeling using Personalized Fitness data for Nutrition Recommendations

Submitted By:

Vishal

(2K21/CO/521)

Vishal Kumar Gupta

(2K21/CO/525)

Under the Guidance of

Dr. Anil Singh Parihar

(Department of Computer Science & Engineering)

Delhi Technological University

(Formerly Delhi College of Engineering)
Bawana Road, Shahbad Daulatpur
Delhi-110042

May-2024

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have contributed to the successful completion of this mini-project. Firstly, we would like to thank our faculty advisor, Dr. Anil Singh Parihar, for providing us with valuable guidance and support throughout the project.

We would also like to thank our peers for their encouragement and assistance, which helped us to overcome various obstacles during the project.

Finally, we would like to express our thanks to the authorities of Delhi Technological University for providing us with the resources and infrastructure necessary to complete this mini-project.

Place: Delhi Vishal (2K21/CO/521)

Date: 03/06/24 Vishal Kumar Gupta(2K21/CO/521)

Candidate Declaration

We hereby declare that the Mini Project Work Report entitled "Predictive Modelling using Personalised Fitness Data for Nutrition Recommendations" is being submitted to the Delhi Technological University, New Delhi. The material contained in this Mini Project Report has not been submitted to any University or Institution for the award of any degree but, to the best of our knowledge.

Name of the Student (Roll Number) with Signature:

(1) Vishal (2K21/CO/521)

(2) Vishal Kumar Gupta (2K21/CO/525)

Department of Computer Science & Engineering
Place: DTU, New Delhi
Date: 03/06/24

CERTIFICATE

This is to certify that the Minor Project Work Report entitled " Predictive Modelling using Personalised Fitness Data for Nutrition Recommendations" submitted by:

Name of the Student (Roll Number):

(1) Vishal (2K21/CO/521)

(2) Vishal Kumar Gupta (2K21/CO/525)

as the record of the work carried out by them, is accepted as the B.Tech. Mini Project work report submission in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in the Department of Computer Science & Engineering, DTU Rohini.

Dr. Anil Singh Parihar (Professor)
Department of Computer Science & Engineering
DTU Rohini- 110042

Abstract

In the health and fitness industry, accurately predicting individual nutritional needs is crucial for achieving personalised health outcomes. Traditional methods often struggle with the high dimensionality of fitness data and the dynamic nature of individual health metrics. This project addresses two critical areas in health analysis: optimizing feature reduction for personalised nutrition recommendations using fitness data, and classifying obesity levels to enhance nutrition recommendations.

The first part of the project involves developing a novel model using machine learning techniques to optimally reduce features while maintaining high predictive accuracy. Specifically, we employ a Random Forest ensemble technique to classify obesity levels based on personalised fitness data. The second part focuses on using the classified obesity levels to recommend suitable food and nutrition plans. By integrating feature reduction and obesity classification models, the project aims to create a comprehensive tool for personalised health assessment.

The expected outcomes include an optimised feature set that improves predictive accuracy and efficiency, enhanced data processing and interpretation methods, a robust obesity classification tool, and improved decision-making capabilities for individuals and healthcare providers. Additionally, the project aims to contribute to better health outcomes by enabling earlier and more accurate identification of nutritional needs, thus allowing proactive measures to optimise health and wellness. The methodologies developed in this project also offer valuable contributions to both academic research and practical applications in health and fitness analysis.

Contents

Chapter-1

Introduction

1.1 Problem Statement: Nowadays, numerous individuals face challenges in discovering suitable nutrition and effectively controlling their weight. Conventional approaches to providing nutritional guidance frequently fail since they struggle to manage the intricate and constantly evolving characteristics of individual health and fitness information. Furthermore, in light of the escalating issue of obesity, it is essential to develop more accurate and tailored nutrition plans.

The goal of this project is to address these problems by leveraging machine learning to generate improved and customized nutrition suggestions. This project consists of two primary components:

1.1.1 Extracting Features from Data: We require a more efficient method to Extract features from this data without compromising the accurate prediction of an individual's nutritional requirements.

1.1.2 Classifying Obesity Levels: To get maximum accuracy we can classify data into different obesity levels.

To meet these objectives, the project will utilize a machine learning method known as Random Forest to categorize individuals according to their obesity levels using their own fitness information. Based on these categories, we can suggest appropriate dietary and nutritional programs that are customized for each individual's requirements.

1.2 Objectives:-

- Feature reduction from Fitness Data[14]: Develop a method to make sense of complex fitness data while keeping it accurate for predicting nutrition needs.
- Classify Obesity Levels: Use Random Forest[10] to accurately determine obesity levels from personal fitness data.
- Recommend Personalized Nutrition Plans: Based on obesity levels, suggest the best food and nutrition plans for individuals.

- Building A Web Application: After getting a nutrition plan we have to create a Python-based web application that predicts healthy food.

1.3 Expected Outcomes:-

- A simpler and more accurate way to understand fitness data.
- A reliable tool for classifying obesity levels that helps in making better nutrition decisions.
- A software product to tailor personalized nutrition plans that lead to better health and wellness.
- Contributions to practical applications in health and fitness.

By working on these goals, this project aims to provide a comprehensive and effective tool for better health assessment and personalized nutrition planning, ultimately helping people lead healthier lives.

Chapter - 2

Software Requirement Specifications

2.1 Introduction

The purpose of this SRS is to create a design of a full-scale web application that uses Flask as frontend, RESTapi for routing.

2.1.1 Purpose

The purpose of this model is to classify obesity levels using the Random Forest Ensemble technique and recommend personalized levels.

2.1.2 Scope

The personalized dietary recommendation model is designed to help individuals. and healthcare providers make informed nutritional decisions. The system processes individual data from users and makes the healthy food prediction/

2.1.3 Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification
- Random Forest: Machine Learning Algorithm for Classification.
- JSON: JavaScript Object Notation
- API: Application Programming Interface

2.1.4 Overview

This document provides a comprehensive overview of functional and non-functional requirements, system characteristics, and design constraints. This is a guide for developers and project stakeholders.

2.2.2 Overall Description

2.2.1 Product Perspective

The Personal Nutrition Recommender Model relies on user-supplied exercise data; it is a stand-alone application that includes a web-based data entry interface for food records and prediction.

2.2.2 Product Functions

- Data Processing: Accept JSON files containing health data.
- Feature Reduction: Reduce the dimensionality of a dataset using machine learning techniques.

- Obesity Classification: Use Random Forest to classify obesity.
- Nutrition Recommendation: Provide personalized nutrition plans based on classified obesity levels or recommend healthy food.

2.2.3 User Classes and Characteristics

- Individuals: Require personalized nutrition plans to improve the health condition of the user.
- Healthcare Providers: require insights into patient's health data to predict obesity levels and based on this recommend healthy nutrition plans.

2.2.4 Operating Environment

- A web server running a suitable web framework like Flask or Django.
- Require modern web browsers like Chrome, Firefox, and Safari.
- Backend processing on Python version 3.

2.2.5 Design and Implementation Constraints

- The system must handle large JSON files efficiently.
- Real-time processing and visualization should have minimal latency.
- The system must be secure to prevent unauthorized access to sensitive fitness data.

2.2.6 User Documentation

User documentation will be provided in the form of an online help guide accessible from the web application. It will include entering the user data and based on these data our application predicts the food.

2.2.7 Assumptions and Dependencies

- The user provides correctly formatted JSON files.
- The web server has sufficient resources to handle large datasets[14].
- The application relies on external libraries such as Flask and matplotlib, which must be maintained.

2.3. Specific Requirements

2.3.1 Functional Requirements

2.3.1.1 Data Ingestion

- The system shall accept JSON files or user data in the form of a dictionary on the web.
- The system shall validate the content of the JSON files before processing.

2.3.1.2 Feature Reduction

- The system shall reduce the number of features in the dataset using various techniques.
- The system shall maintain or improve predictive accuracy after feature reduction.

2.3.1.3 Obesity Level Classification

- The tool shall use the random forest ensemble[10] technique for model training.
- The tool uses fitness data to improve classification accuracy.

2.3.1.4 Nutrition Recommendation

- The application tool shall provide personalized nutrition plans based on classified obesity levels.
- The application tool shall offer actionable nutrition advice to individual needs.

2.3.1.5 Real-Time Visualization

- The application tool shall display real-time output on the web interface

2.3.2 Non-Functional Requirements

2.3.2.1 Performance

- It processes large datasets with minimal latency.
- The application processes multiple data[9] at a time.

2.3.2.2 Usability

- The interface should be easy to navigate.
- The system shall provide clear insights from the predictions.

2.3.2.3 Reliability

- The system shall be available 99.9% of the time.
- The system shall handle errors and exceptions easily.

2.3.2.4 Security

- It ensures data privacy and security.

2.3.2.5 Maintainability

- The system shall be modular and follow best coding practices.
- The maintenance cost of the system is very low.

2.3.3 Interface Requirements

2.3.3.1 User Interfaces

- The system shall provide a web-based interface for data input and result visualization.
- The system shall use a responsive design to support different devices and screen sizes.

2.3.3.2 Hardware Interfaces

- The system shall not have any specific hardware interface requirements beyond standard web server hardware.

2.3.3.3 Software Interfaces

- The application uses Flask[12] as the web framework.
- The system shall use Matplotlib for data visualization.
- The application shall interact with JSON files for data[9] input.

2.3.3.4 Communication Interfaces

- The application uses HTTP/HTTPS[11] for communication between the client and server.

2.4. System Requirements

2.4.1 Hardware:

- Processor: Pentium Chip or better
- RAM: 2 GB or more
- Hard Disk Space: 500MB or more

2.4.2 Software:

- Operating System: Windows XP or above, Linux, MacOS
- Web Browser: Google Chrome, Mozilla Firefox, Safari
- Languages: Python

Chapter - 3

Design of the Product

we are developing a software system that asks questions to end users by providing the users answers to the questions based on their dietary habits and lifestyle preferences, we predict where they stand with respect to the obese category. The algorithm uses these obesity levels to suggest meal options for achieving a healthy lifestyle. This chapter explains the methodology and structure for developing such a tool.

3.1 Building a Web Application

3.1.1 Front-End Designing: The front end provides an interface between the user and the software. It helps the user understand the functionality and features of the product without getting into technical details. Here are the tools used to create the front-end part of this software:

- **HTML:** Web pages rely on HTML (Hypertext Markup Language) to organize and present their content. It forms the fundamental building blocks of online information, allowing developers to give attributes such as paragraphs, headers, links, and images for elements. It also creates the layout structure in conjunction with other UI/UX software products. Furthermore, it forms the form at a high level for whole fields or data objects.

- **CSS:** Cascading Style Sheets--CSS for short--is a style sheet language that helps control the fonts, colors, and page layouts while protecting the overall look of a page. Alternatively stated, it disregards all questions such as how you produced font data into a given technology by doing so.

- **Javascript[13]:** It is a programming language that gives functionality to HTML and CSS buttons and elements. for example, JavaScript[13] provides the required functionality for navigation buttons to navigate from one web page to another.

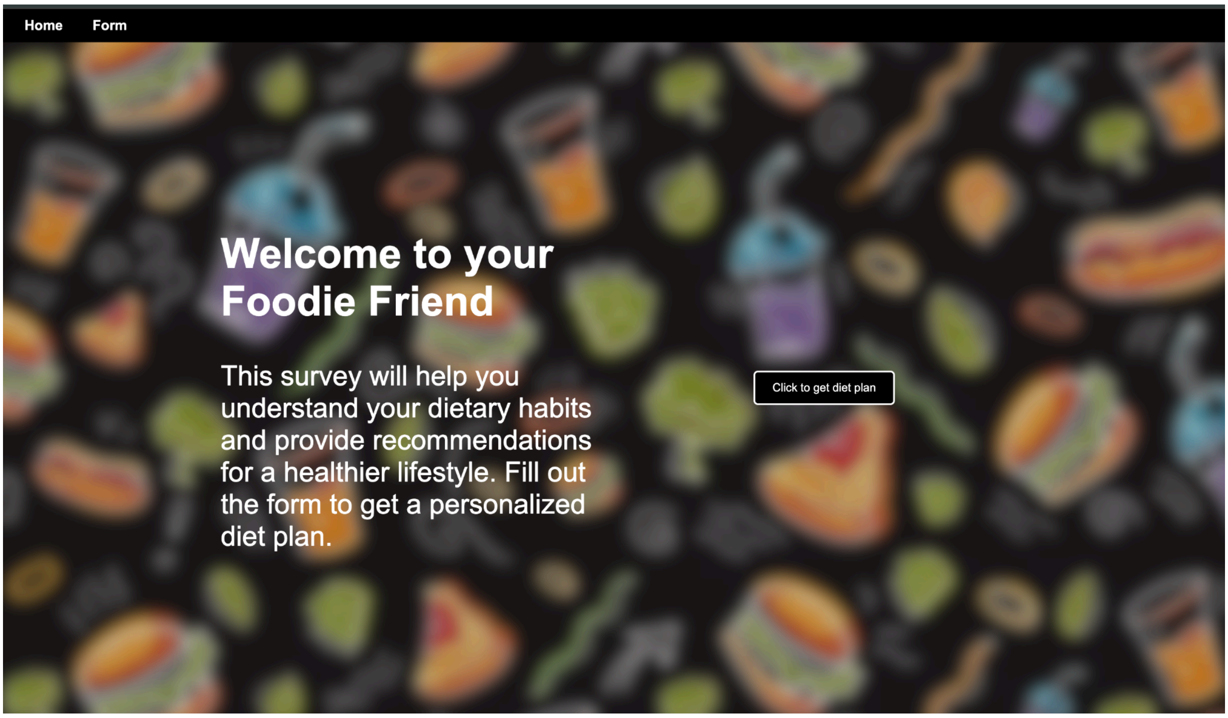

Figure 3.1: Front-end design of product

3.1.2 Back-End Designing: Here backend processes user responses, performs the prediction of the machine learning model, and displays the result using Flask [12].

- **Flask[12]:** Flask is a microweb framework based on Python language that efficiently manages routes, handles form submissions, and renders[12] models. User inputs are collected from the form, transformed by the model, and predictions are generated. Using HTML templates[12], the results are processed and presented. The machine learning model and user interface are well-synchronized[12] thanks to this setup.

3.2 Methodology

3.2.1 Requirement Gathering: The first phase of our software development process is requirement gathering. we needed datasets to train our model so we analyzed several datasets available on Kaggle and selected the most suitable dataset[14] for the following substantial reasons:

- Availability of Huge amounts of fitness data[14].

- Abundant features for model training.
- Availability of Lifestyle preferences, such as screen time (in hours) and smoking habit(e.g., Yes, No).

3.2.2 Feature Reduction from Dataset: The Second phase of our software development process is extracting essential features from the dataset to increase the efficiency of our product. Here The Pearson correlation plotting technique is used to do so.

- Pearson correlation plotting technique: this relation measures the linear correlation between two sets of data.

$$r = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}} \qquad (3.1)$$

where,

r = correlation coefficient
x_i = values of the x-variable in a sample
$\bar{x}$ = mean of the values of the x-variable
y_i = values of the y-variable in a sample
$\bar{y}$ = mean of the values of the y-variable

3.2.3 Model Pipeline: Using a pipeline, you may preprocess the data by applying a series of transformers one after the other. It consists of various steps that can handle missing values, encoding the categorical value, standardizing the numerical value, and many more.

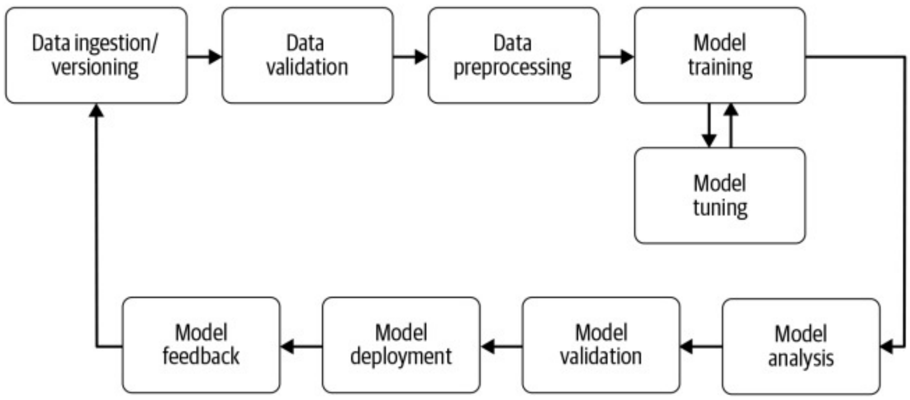

Figure 3.2: Model Pipeline Architecture

This diagram consists of two pipelines numerical and categorical for categorical data. After that, the column transformer joins the two pipelines.

3.2.4 Model Training: we used various machine learning algorithms like KNN[3,4] (k- nearest neighbor) classifier, Decision tree classifier, and Random forest classifier to classify obesity levels. The accuracy of the Random forest classifier outperforms the other two accuracies of the other two algorithms.

3.2.5 Product Deployment: This model uses a cloud environment so that is why we deployed our final product on vercel.

3.3 Algorithm

We used multiple machine learning techniques to build our model like the k-nearest neighbor classifier[4], decision tree classifier, and random forest classifier. Out of these algorithms random forest gives the maximum accuracy.

3.3.1 k-nearest neighbors classifier: it is the supervised machine learning technique that uses various distance metrics like Euclidean distance, Manhattan distance, etc for classification. After using this algorithm, the accuracy of our model is 82%.

```
Accuracy Score = 0.82
{'algorithm': 'auto', 'metric': 'manhattan', 'n_neighbors': 4, 'p': 1, 'weights': 'distance'}

Classification Report:
                     precision    recall  f1-score   support

Insufficient_Weight       0.83      0.84      0.83        92
      Normal_Weight       0.79      0.53      0.64        77
     Obesity_Type_I       0.83      0.83      0.83       114
    Obesity_Type_II       0.84      0.94      0.89        85
   Obesity_Type_III       0.91      1.00      0.95        92
 Overweight_Level_I       0.78      0.78      0.78        89
Overweight_Level_II       0.70      0.74      0.72        85

           accuracy                           0.82       634
          macro avg       0.81      0.81      0.81       634
       weighted avg       0.81      0.82      0.81       634
```

Figure 3.3: KNN Accuracy of the model

3.3.2 Decision tree classifier: it is the supervised machine learning technique[4] that uses various metrics for splitting the tree which are gini impurity, entropy, and information gain. After reducing the features it classifies the obesity level. The accuracy of our model through this technique is 76%.

- **Entropy:** A node's entropy is a measurement of its disorder or impurity. Entropy would therefore be greater for a node with a more varied composition for example, 2Pass and 2 Fail than for a node with just pass or only fail.

$$H(X) = -\sum_{x \in X} p(x) log_b p(x) \qquad (3.2)$$

where,

H(X) = Node's entropy

x = features

p(x) = probability.

- **Information Gain Function:** The information gain function calculates the mutual information for each feature in the selected subset, helping to assess the relevance of features.

$$IG(T, a) = H(T) - H(T|a) \qquad (3.3)$$

where,

IG = information gain.

H(x) = entropy.

T = feature.

a = child feature.

```
Decision Tree:
--------------------------------------------------------------
Accuracy: 0.75237
Accuracy w/Scaled Data (ss): 0.76025
Accuracy w/Scaled Data (mm): 0.76814

Classification Report (mm):
                      precision    recall  f1-score   support

 Insufficient_Weight       0.84      0.84      0.84        92
       Normal_Weight       0.52      0.49      0.51        77
      Obesity_Type_I       0.81      0.72      0.76       114
     Obesity_Type_II       0.87      0.87      0.87        85
    Obesity_Type_III       0.98      0.99      0.98        92
  Overweight_Level_I       0.71      0.69      0.70        89
 Overweight_Level_II       0.62      0.75      0.68        85

            accuracy                           0.77       634
           macro avg       0.76      0.76      0.76       634
        weighted avg       0.77      0.77      0.77       634

                             ----------------------------------
```

Figure 3.4: Decision Tree Accuracy of the model

3.3.3 Random Forest Ensemble[10]: Random[10] Forest works by constructing multiple decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees. It's effective for this task because it can handle a large number of input variables and is resistant to overfitting(the model performs better on the training set and performs poor on the testing set), which is crucial when dealing with complex datasets[14] like those related to health and lifestyle.

The random forest ensemble uses the bagging technique[5] which consists of multiple decision trees. Out of those trees, it selects the best accuracy model. The important point of the bagging technique[5] is that all models are independent of each other

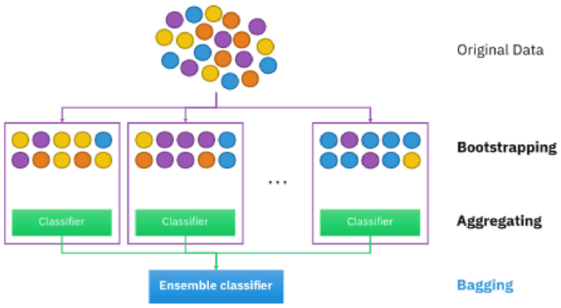

Figure 3.5 : Bagging

3.3.3.1 Hyperparameters in Random Forest [10]:

- **n_estimators**: Number of trees the algorithm builds before the predictions.
- **max_features**: Maximum number of features random forest considers splitting a node.
- **mini_sample_leaf**: minimum leaf is required to split the node.
- **criterion**: the splitting criteria is Entropy or Gini impurity.
- **max_leaf_nodes**: maximum number of leaves in each tree.

Through this technique, the accuracy of our model is 95%.

```
Random Forest :
Classification Repot
                     precision    recall  f1-score   support

Insufficient_Weight       0.97      0.95      0.96        78
      Normal_Weight       0.81      0.96      0.88        82
     Obesity_Type_I       0.99      0.98      0.99       115
    Obesity_Type_II       1.00      1.00      1.00        96
   Obesity_Type_III       1.00      1.00      1.00        91
 Overweight_Level_I       0.96      0.86      0.91        88
Overweight_Level_II       0.94      0.89      0.91        84

           accuracy                           0.95       634
          macro avg       0.95      0.95      0.95       634
       weighted avg       0.96      0.95      0.95       634
```

Figure 3.6: Random Forest Accuracy of the Model

3.4 System Layout

The software is designed using a three-tier architecture[11] method containing the Presentation layer, Data layer, and Application layer.

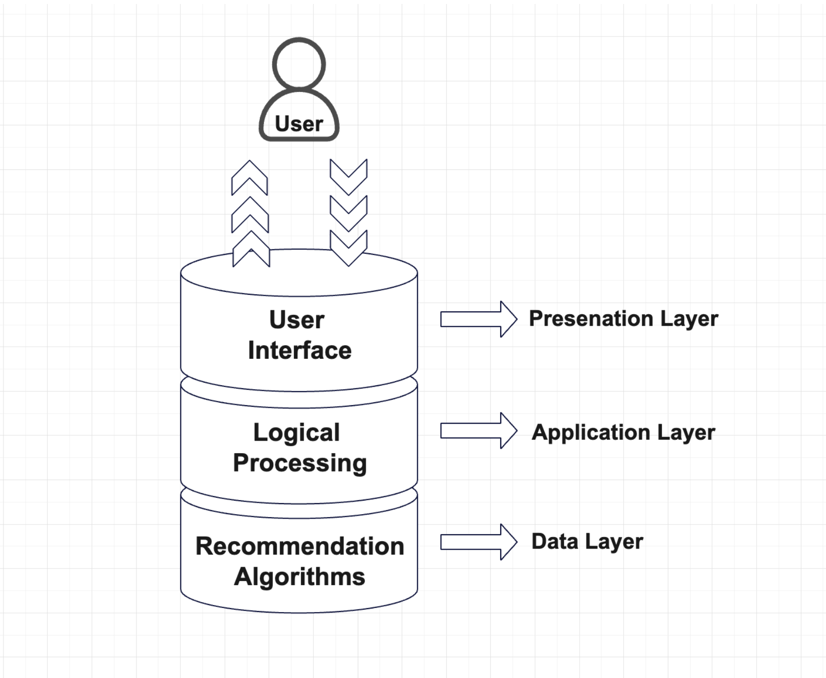

Figure 3.7: System Layout Design

3.4.1 Presentation Layer :

- Home Section: The main software product interface from where users start using the product.
- Form Section: it is a secondary interface where for instance voting takes place in the health questionnaire (eating habits, health metrics)
- Secondary Output Section: The tertiary interface with interactive food recommendations.

3.4.2 Application Layer :

- Processing User Responses: This subpart gathers and processes the input entered by the user through the form on the application[11,12].
- Generating Recommendations: Based on the responses and processed information, this layer interacts with the data layer to create personalized food recommendations[2].

3.4.3 Data Layer :

- Food Recommendation Algorithms: Contains the necessary algorithms and data for generating food recommendations using the user inputs.

3.4.3 Flow of Data [11]:

- User Interaction: Users, through the form, send replies to the residing secondary presentation layer.
- The Responses Processing:- At the application layer entered user responses are processed by the application.
- Generating Meal Recommendation: The application layer generates food recommendations using the algorithms that the data layer has.

Chapter 4

System Manual

The system manual is a document aiming to guide users to use the application at full efficiency. This contains every feature and their contribution with effect on the software product.

4.1 Introduction

This software product is created to provide food recommendations based on user responses. This system contains three main parts: Home Page, Form, and Output screen. The Home page briefs about the software and contains a button that navigates to the Form section. The form section contains a bunch of questions that collect categorical and numerical information from the user. Finally, the Output screen shows the recommended food[3] based on user responses.

4.2 Components of Application

4.2.1 Home

The Home Page acts as the initial interface of the software[12]. It includes a brief introduction to the software and a navigation button.

- Introduction: Briefly describe the objective and functionality of the tool
- Navigation Button: A button labeled “ click to get diet plan” that takes the user to the form section.

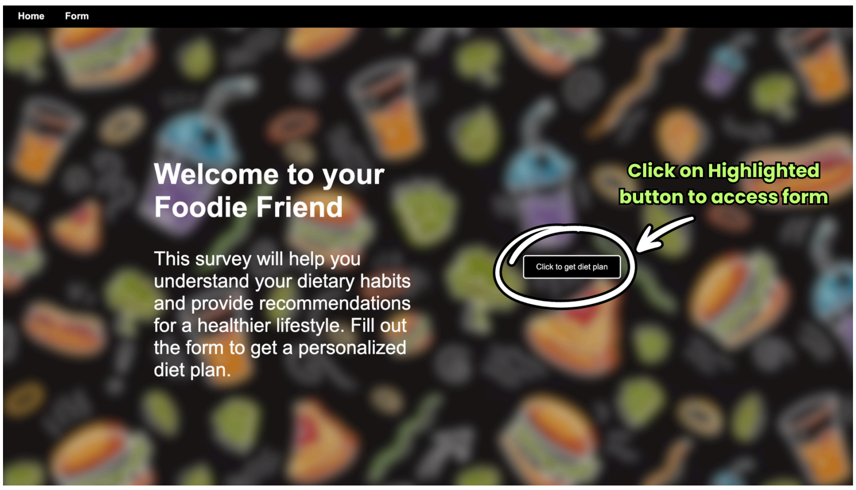

Figure 4.1: Home page section of the Application

4.2.2 Form

A 10-question, user-data gathering questionnaire is housed in this section After the User completes the form, they can submit the answers to go to the Output screen.

- Categorical Questions: These are the questions that accept the predefined inputs Eg: Gender (Male, Female).
- Numerical Questions: Questions that take numeric inputs, for example, Time spent on Technology devices daily in hours.

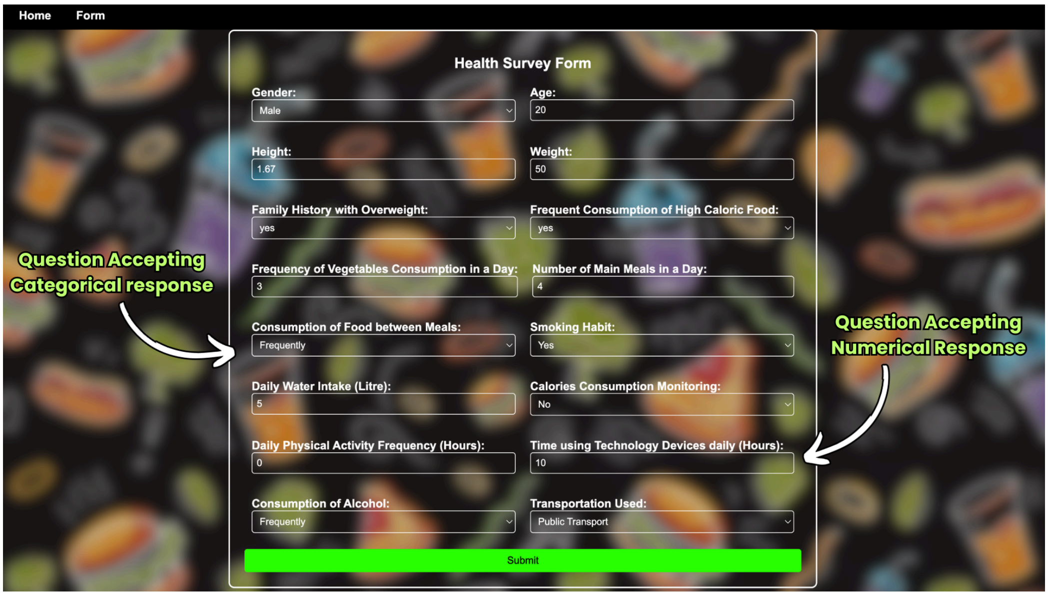

Figure 4.2: Types of Questions in The Form

- Submit button: A button that submits the user's response and redirects to the output section.

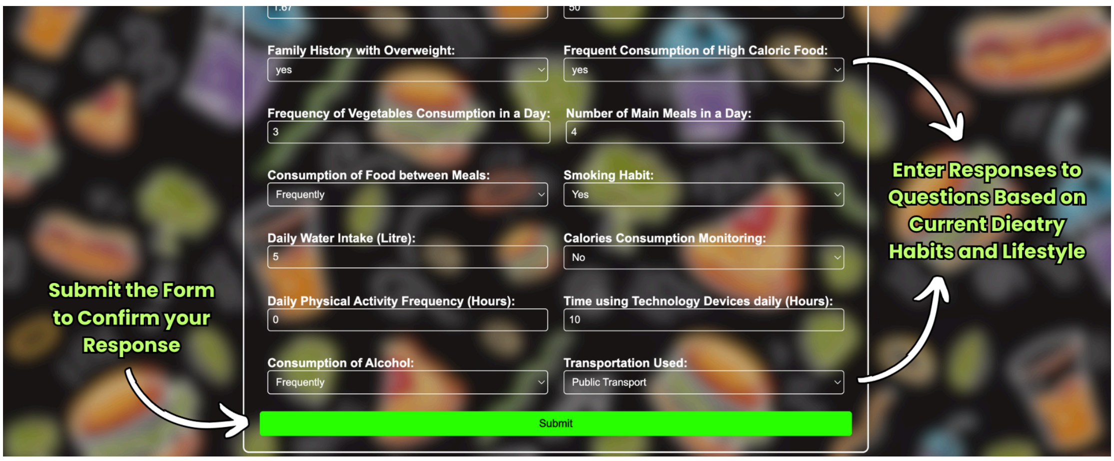

Figure 4.3: Functioning of the Submit button

4.2.3 Output

Depending on the survey results, the output screen will display food recommendations[3] of the user.

- Suggested food plan: A recipe list for breakfast, lunch, and dinner based on user preferences
- Clickable Button: A button of " Go back to Home page" where a user can be directed to the Home section

Figure 4.4: Result Screen of Application

4.3 Technical Specifications

4.3.1 System Requirements

- Operating Systems Supported: macOS, Windows, Linux
- Browser Preferences: Chrome, Firefox, Safari or Edge
- Internet connection: Required for accessing the software.

Chapter 5

Results

Our Application successfully analyses user responses and based on the model training from the dataset available we classify the obesity level[3] of the user. Based on the obesity level we provided regionally available meal options for breakfast, Lunch, and Dinner.

After all calculations were made to obtain the mass body index for each individual, the results were compared with the data provided by WHO and the Mexican Normativity [7].

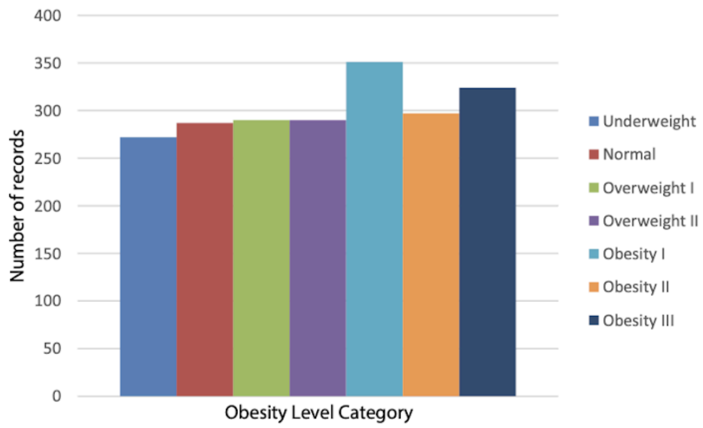

Figure 5.1: Prediction Sample

5.1. Application Result Screen

For a specific user response, our application recommended the below meal options to the user according to his current physical status and lifestyle preferences to improve the user's dietary habits. they are based on the obesity level calculated using their inputs.

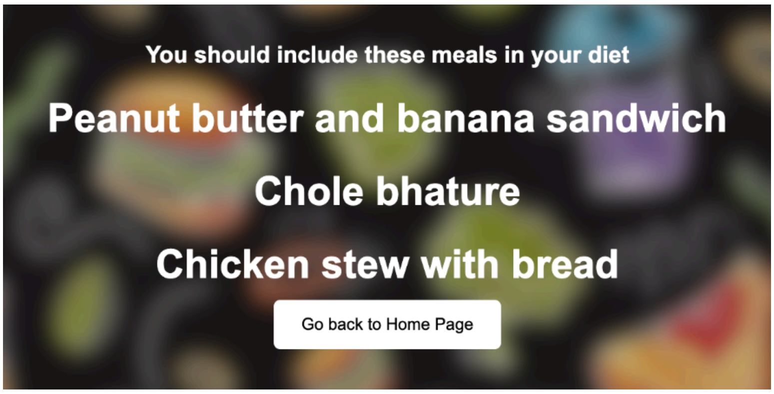

Figure 5.2: Application Result Sample

Chapter 6

Conclusion

The objective of this project was to build software to predict the level of obesity of an individual and suggest suitable meals according to their tastes. It uses a random forest classifier, based on which predictor variable importance values, dietary habits, and lifestyle preferences of the population are consistently able to predict obesity levels with a 95% accuracy rate. The lessons learned from this project have shown that the use of good data and feature selection by, for example, plotting a correlation matrix with the Pearson correlation statistic can improve the ability to predict. The tool guarantees both access and scalability by having an understandable interface and publishing the tool on the Vercel cloud platform. An extension for the future could be the incorporation of real-time monitoring of health statistics for a more in-depth approach to health.

Chapter – 7

Future Work

The form of this project is a chapter that focuses on components of the project that are worthy of further improvement and expansion for more effective results.

6.1 Including Supplementary Data Sources:

- Biometrics that are tailored-specific: Incorporate other biometric data such as genetic information, sleep patterns, and even heart rate to render dietary recommendations further personalized and exact.
- Environmental and Contextual Information: Incorporate things like location and weather and the social context to give more adaptive and relevant recommendations[3, 4].

6.2 Advanced Modeling Approaches:

- Machine and Deep Learning Fusion: Step Step Integration of deep learning with machine learning to build better more accurate recommendation personal recommended solution
- Explainable AI Models: Create AI models with deductive reasoning for the purpose of enhanced consumer trust and comprehension

References

[1] M.V. Olmedo, La obesidad: un problema de salud pública. Revista de divulgacio científica y tecnologica de la Universidad Veracruzana, 2011. Recuperado de: https://www.uv.mx/cienciahombre/revistae/vol24num3/articulos/obesidad/.

[2]C.Davila-Payan,M.DeGuzman,K.Johnson,N.Serban,J.Swann,Estimatingprevalenceofoverweightorobesechildrenand adolescents in small geographic areas using publicly available data, Prev. Chronic Dis. 12 (2015).

[3] S. Manna, A.M. Jewkes, Understanding early childhood obesity risks: an empirical study using fuzzy signatures, in: Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on, IEEE, 2014, July, pp. 1333e1339.

[4] M.H.B.M. Adnan, W. Husain, A hybrid approach using Naïve Bayes and Genetic Algorithm for childhood obesity prediction, in: Computer & Information Science (ICCIS), 2012 International Conference on vol. 1, IEEE, 2012, June, pp. 281e285.

[5] T.M. Dugan, S. Mukhopadhyay, A. Carroll, S. Downs, Machine learning techniques for prediction of early childhood obesity,
Appl. Clin. Inf. 6 (3) (2015) 506e520.

[6] Eduardo De-La-Hoz-Correa, Fabio E. Mendoza-Palechor, Alexis De-La-Hoz-Manotas, Roberto C. Morales-Ortega, Beatriz
Adriana Sanchez Hernandez, Obesity level estimation software based on decision Trees, J. Comput. Sci. 15 (Issue 1) (2019)
67e77, https://doi.org/10.3844/jcssp.2019.67.77.

[7] DO, NORMA Oficial Mexicana NOM-008-SSA3-2010, Para el tratamiento integral del sobrepeso y la obesidad, Diario Oficial,
2010.

[8] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell.
Res. 16 (2002) 321e357.

[9] A. Zadeh and S. Khoshroo, "A Food Recommender System Considering Nutritional Information and User Preferences," 2019. [Online]. Available: https://www.researchgate.net/publication/334529528_A_Food_Recommender_System_Considering_Nutritional_Information_and_User_Preferences.

[10] T. K. Ho, "Random Decision Forests," in Proc. 3rd Int. Conf. Document Anal. Recognit., Montreal, QC, Canada, Aug. 14-16, 1995, pp. 278-282.

.[11] J. J. Donovan, "Three-tier architecture," Wikipedia, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Multitier_architecture#Three-tier_architecture. [Accessed: 16-Jun-2024].

[12] "Quickstart — Flask Documentation (3.0.x)," Pallets Projects, 2023. [Online]. Available: https://flask.palletsprojects.com/en/3.0.x/quickstart/. [Accessed: 19-Nov-2023].

[13] "JavaScript," Mozilla Developer Network, 2023. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript. [Accessed: 11-Nov-2023].

[14] "J. Jayita Bhattacharyya, "Estimation of Obesity Levels UCI Dataset," Kaggle, 2024. [Online]. Available:https://www.kaggle.com/datasets/jayitabhattacharyya/estimation-of-obesity-levels-uci-dataset. [Accessed: 17-Sep-2023].