# Project Report

| | |
|---|---|
| Date | 16 October 2022 |
| Team ID | PNT2022TMID35928 |
| Project Name | AI-POWERED NUTRITION ANALYZER FOR FITNESS ENTHUSIASTS |
| Maximum Marks | 4 Marks |

AI-Powered Nutrition Analyzer For Fitness Enthusiasts

Category: Artificial Intelligence

## 1. INTRODUCTION

### Project Overview

The main aim of the project is to building a model which is used for classifying the fruit depends on the different characteristics like colour, shape, texture etc. Here the user can capture the images of different fruits and then the image will be sent the trained model. The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.) and also display the current price in market.

### Purpose

Food is essential for human life and has been the concern of many healthcare conventions. Nowadays new dietary asses sment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and maintain a healthy diet. Nutritional analysis is the process of determining the nutritional content of food. It is a vital part of analytical chemistry that provides information about the chemical composition, processing, quality control and contamination of food.So it is important that we analyze nutrient content which is essential for ensuring our fitness and a healthy lifestyle.

## 2. LITERATURE SURVEY

- Fruit is one of the most popular products in the market. Automatic and accurate classification of fruit can bring great convenience to fruit sellers. However, there are great similarities between some apple varieties and pears and peaches, and these kinds of fruit are generally popular, which has increased the difficulty of this task. Aiming at this problem, this paper proposes a method of fruit automatic recognition and classification based on convolutional neural network. First, we obtained two color fruit image data set (public data set and self-made data set). The public data sets is composed of fruit images with simple background, while the fruit images in the self-made data set are taken in a complex environment. Then, on the basis of convolutional neural network, we conducted several research experiments through parameter adjustment, and achieved the highest average classification accuracy of 99.8% on the public data set. In the self-made data set, the classification accuracy is 90.2%. Finally, we improved the classification accuracy of the selfmade data set from the original 90.2% to 98.9% by adopting appropriate data enhancement techniques.

  **References:** Fruit Classification using Convolutional Neural Network via Adjust Parameter and Data Enhancement.

  **Authors:** Liuchen Wu; Hui Zhang; Ruibo Chen; Junfei Yi

  **Published in:** 2021 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)

- Deep Learning algorithms are designed in such a way that they mimic the function of the human cerebral cortex. These algorithms are representations of deep neural networks i.e. neural networks with many hidden layers. Convolutional neural networks are deep learning algorithms that can train large datasets with millions of parameters, in form of 2D images as input and convolve it with filters to produce the desired outputs. In this article, CNN models are built to evaluate its performance on image recognition and detection datasets. The algorithm is implemented on MNIST and CIFAR-10 dataset and its performance are evaluated. The accuracy of models on MNIST is 99.6 %, CIFAR-10 is using real-time data augmentation and dropout on CPU unit.

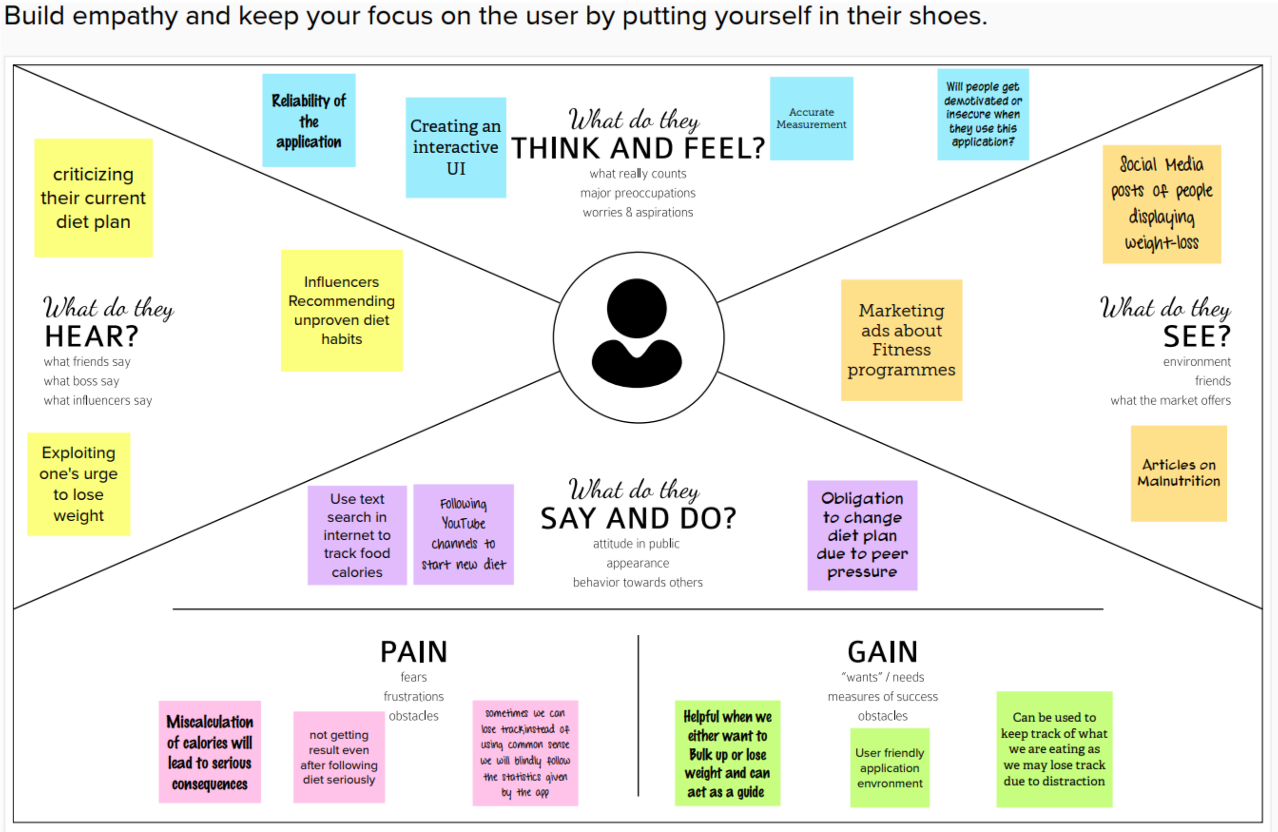
**References:** Convolutional Neural Network (CNN) for Image Detection and Recognition

**Authors:** Rahul Chauhan; Kamal Kumar Ghanshala; R.C Joshi

**Published in:** 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)

# 3. IDEATION & PROPOSED SOLUTION

- Empathy Map Canvas



Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they THINK AND FEEL?

what really counts
major preoccupations
worries & aspirations

Reliability of the application

Creating an interactive UI

Accurate Measurement

Will people get demotivated or insecure when they use this application?

What do they HEAR?

what friends say
what boss say
what influencers say

criticizing their current diet plan

Influencers Recommending unproven diet habits

Exploiting one's urge to lose weight

What do they SEE?

environment
friends
what the market offers

Social Media posts of people displaying weight-loss

Marketing ads about Fitness programmes

Articles on Malnutrition

What do they SAY AND DO?

attitude in public
appearance
behavior towards others

Use text search in internet to track food calories

Following YouTube channels to start new diet

Obligation to change diet plan due to peer pressure

PAIN

fears
frustrations
obstacles

Miscalculation of calories will lead to serious consequences

not getting result even after following diet seriously

sometimes we can lose track,instead of using common sense we will blindly follow the statistics given by the app

GAIN

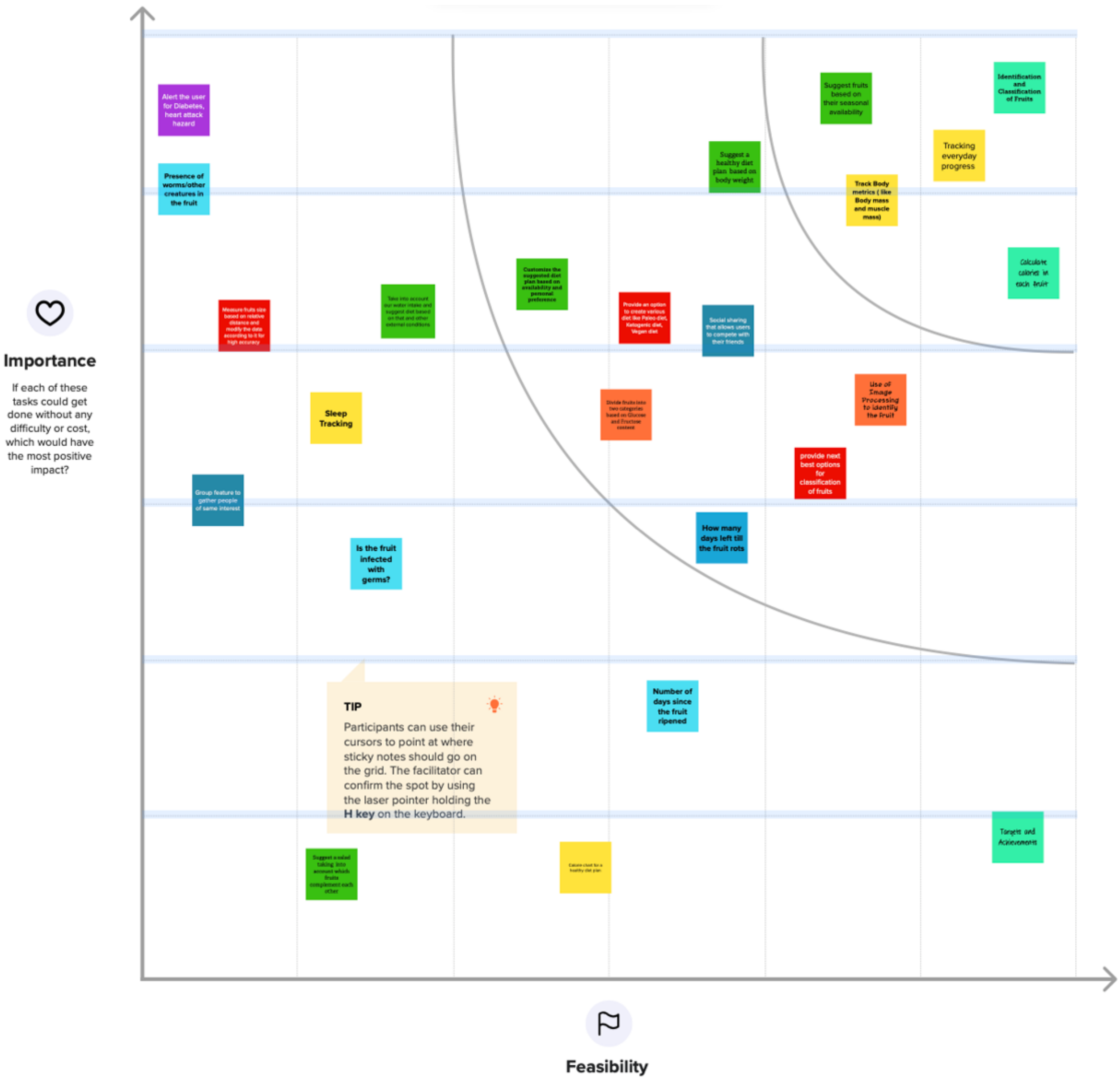
"wants" / needs
measures of success
obstacles

Helpful when we either want to Bulk up or lose weight and can act as a guide

User friendly application envronment

Can be used to keep track of what we are eating as we may lose track due to distraction

- Ideation & Brainstorming


Importance
Feasibility
Alert the user for Diabetes, heart attack hazard
Presence of worms/other creatures in the fruit
Measure fruits size based on relative distance and modify the data according to it for high accuracy
Take into account our water intake and suggest diet based on that and other external conditions
Customize the suggested diet plan based on availability and personal preference
Provide an option to create various diet like Paleo diet, Ketogenic diet, Vegan diet
Social sharing that allows users to compete with their friends
Suggest a healthy diet plan based on body weight
Suggest fruits based on their seasonal availability
Track Body metrics ( like Body mass and muscle mass)
Tracking everyday progress
Identification and Classification of Fruits
Calculate calories in each fruit
Sleep Tracking
Divide fruits into two categories based on Glucose and Fructose content
Use of Image Processing to identify the fruit
provide next best options for classification of fruits
Group feature to gather people of same interest
Is the fruit infected with germs?
How many days left till the fruit rots
Number of days since the fruit ripened
Suggest a salad taking into account which fruits complement each other
Calorie chart for a healthy diet plan
Targets and Achievements

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

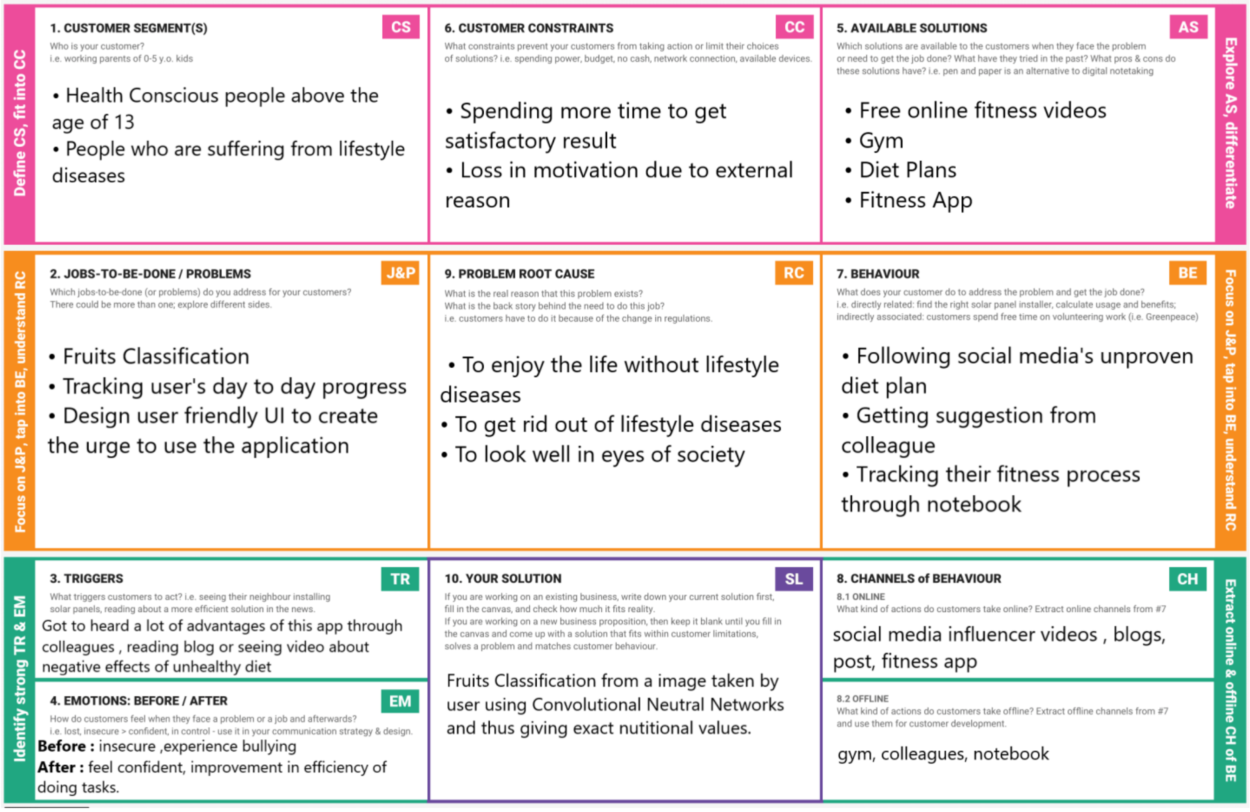
- Proposed Solution

| S.no. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | In recent years, the lifestyles of people have changed so much that has to the increasing demand for fast food culture which has led to the ignorance of personal diet and fitness. The lack of focus on these important aspects of life have led to deteriorating effects on the body such as obesity, diabetes and can also lead to heart attack. A multitude of stats present a disappointing reality. The harsh reality is that people start following various fitness programs on social media platforms but they eventually result in vain due to the time constraint and work pressure. However, there has been a new trend of following social media influencers' diet and fitness regimes cannot be verifiable, due to the questionable legitimacy of the claims made presented by them. The most common reason for people to fail in their diet routine is that they lose their confidence after planning different diet plans and failing to follow all of them. |
| 2. | Idea / Solution description | The objective of this project is to develop a fitness tracker which can motivate users to track their diet and stick to their diet without the eventual abatement. The classification of fruits is planned to be based on Convolutional Neural Network. Primarily, the model is trained using a training dataset of several fruits to be able to accurately measure the calories, sugar, fiber and proteins present in a particular fruit.<br>In addition to this, based on the image taken by the user, real time processing is done and uploaded to a custom-made website. |
| 3. | Novelty / Uniqueness | The differentiating factor for this project is the intuitiveness that keeps in mind the users' needs and requirements, allowing them to stick to their regimen strictly without any distractions. This is |

|  |  |  |
|---|---|---|
|  |  | possible with the help of a user-friendly UI along with useful progress tools that helps the user to monitor their daily activities, where only the carefully curated tools that assist people with their daily progress are provided.<br>This project aids fitness enthusiasts to create a flexible and plausible diet schedule that does not stifle their interests eventually. The fruit pointed by the user is further classified based on nutrients such as calories, sugar, fiber present and calories in each fruit is calculated.<br>The data entered by the user is sent to the cloud. Furthermore, the data stored in the cloud returns all necessary statistics which helps the user to track the progress of his/her regimen, thereby motivating them to continue the diet. This project tracks body metrics as well and Suggests fruits based on seasonal availability. |
| 4. | Social Impact / Customer Satisfaction | If a customer has achieved his short time goals through the app, he/she can be rewarded using badges, etc. Also, we can add a feature of progress to show where the customers stand on the path to achieving their goals. Giving compliments and rewards like these helps the customer to stay motivated. Rewards can be used to unlock various features which motivate customers to obtain the reward at any cost.<br>One benchmark you can set is a Net Promoter Score, which basically entails surveying members and seeing how many would recommend the membership to someone else |
| 5. | Business Model (Revenue Model) | The business model will be a freemium model with an add-on subscription. The Freemium model brings in customers who get used to basic services like tracking personnel diet which lures them to join subscriptions and gives valuable suggestions to users like dynamic food diet for next few days based on fruits availability, also intake of foods based on weather.<br>In order to attract more customers at the initial |

| | | |
|---|---|---|
| | | stage an affiliated-based business model will be<br>used by that user to get certain coupons for the first three customers, they invite to the platform. Also, money can be collected from certain fitness companies to promote their products like whey protein, fitness supplements. |
| 6. | Scalability of the Solution | New machine learning features like identification of rotten fruits, predicting the user's food interest based on food taste, appearance and price etc will be added as exclusive features etc which can be used through subscription.<br>To retain customers this application will be extended to fitness watches, and smartphones which will be helpful in recording important vitals thus the validity of data given to the application becomes normalized without false readings.<br>Social media features with many trusted fitness experts will be added to promote a healthy lifestyle. |

- Problem Solution Fit



| | | | | |
|---|---|---|---|---|
| Define CS, fit into CC | **1. CUSTOMER SEGMENT(S)** CS<br>Who is your customer?<br>i.e. working parents of 0-5 y.o. kids<br>• Health Conscious people above the age of 13<br>• People who are suffering from lifestyle diseases | **6. CUSTOMER CONSTRAINTS** CC<br>What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.<br>• Spending more time to get satisfactory result<br>• Loss in motivation due to external reason | **5. AVAILABLE SOLUTIONS** AS<br>Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking<br>• Free online fitness videos<br>• Gym<br>• Diet Plans<br>• Fitness App | Explore AS, differentiate |
| Focus on J&P, tap into BE, understand RC | **2. JOBS-TO-BE-DONE / PROBLEMS** J&P<br>Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.<br>• Fruits Classification<br>• Tracking user's day to day progress<br>• Design user friendly UI to create the urge to use the application | **9. PROBLEM ROOT CAUSE** RC<br>What is the real reason that this problem exists?<br>What is the back story behind the need to do this job?<br>i.e. customers have to do it because of the change in regulations.<br>• To enjoy the life without lifestyle diseases<br>• To get rid out of lifestyle diseases<br>• To look well in eyes of society | **7. BEHAVIOUR** BE<br>What does your customer do to address the problem and get the job done?<br>i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)<br>• Following social media's unproven diet plan<br>• Getting suggestion from colleague<br>• Tracking their fitness process through notebook | Focus on J&P, tap into BE, understand RC |
| Identify strong TR & EM | **3. TRIGGERS** TR<br>What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.<br>Got to heard a lot of advantages of this app through colleagues , reading blog or seeing video about negative effects of unhealthy diet<br>**4. EMOTIONS: BEFORE / AFTER** EM<br>How do customers feel when they face a problem or a job and afterwards?<br>i.e. lost, insecure > confident, in control - use it in your communication strategy & design.<br>**Before :** insecure ,experience bullying<br>**After :** feel confident, improvement in efficiency of doing tasks. | **10. YOUR SOLUTION** SL<br>If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.<br>If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.<br>Fruits Classification from a image taken by user using Convolutional Neutral Networks and thus giving exact nutitional values. | **8. CHANNELS of BEHAVIOUR** CH<br>8.1 ONLINE<br>What kind of actions do customers take online? Extract online channels from #7<br>social media influencer videos , blogs, post, fitness app<br>8.2 OFFLINE<br>What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.<br>gym, colleagues, notebook | Extract online & offline CH of BE |

# 4. Requirement Analysis

| | |
|---|---|
| Date | 16 October 2022 |
| Team ID | PNT2022TMID35928 |
| Project Name | AI POWERED NUTRITION ANALYZER FOR FITNESS ENTHUSIASTS |
| Maximum Marks | 4 Marks |

**Functional Requirements :**
Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration through Form<br>Registration through Gmail<br>Registration through LinkedIN |
| FR-2 | User Confirmation | Confirmation via Email<br>Confirmation via OTP |
| FR-3 | User Interface | Intuitive and easy to navigate |
| FR-4 | Business Model | Freemium Model |
| FR-4 | Dataset Collection | Daily intake of calories and food physical activities |
| FR-5 | Training and Testing | Providing necessary informationwith great accuracy |

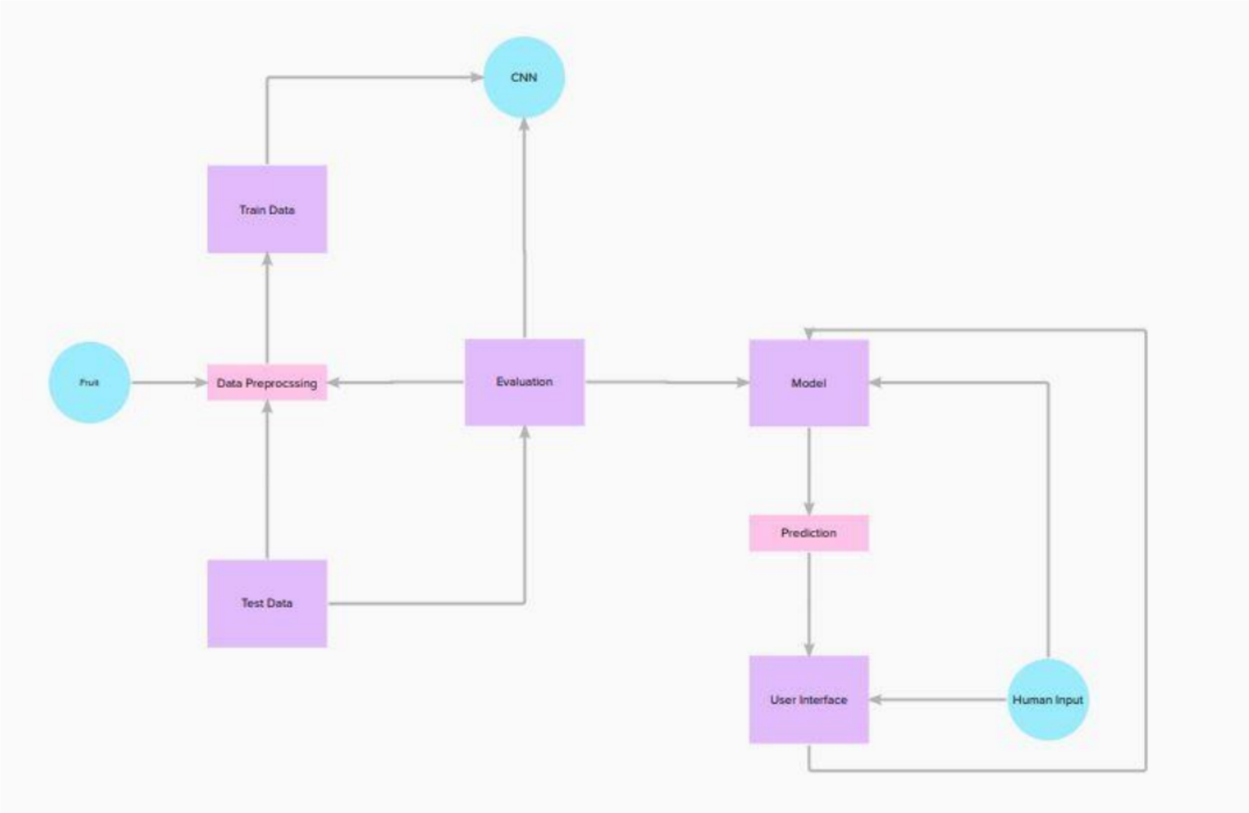
**Non-functional Requirements :**
Following are the non-functional requirements of the proposed solution.

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | Measure Calorie Intake |
| NFR-2 | **Security** | Minimum amount of food necessary for energy requirements will be considered and proper confidentiality standards will met to prevent dataleakage |
| NFR-3 | **Reliability** | The model should predict with great accuracy the number of calories present in a fruit |
| NFR-4 | **Performance** | The images captured should be fed into the model to provide the required information of amount ofcalories |
| NFR-5 | **Availability** | Compatible as both web apps and native apps |
| NFR-6 | **Scalability** | Extend to different types of nutrients |

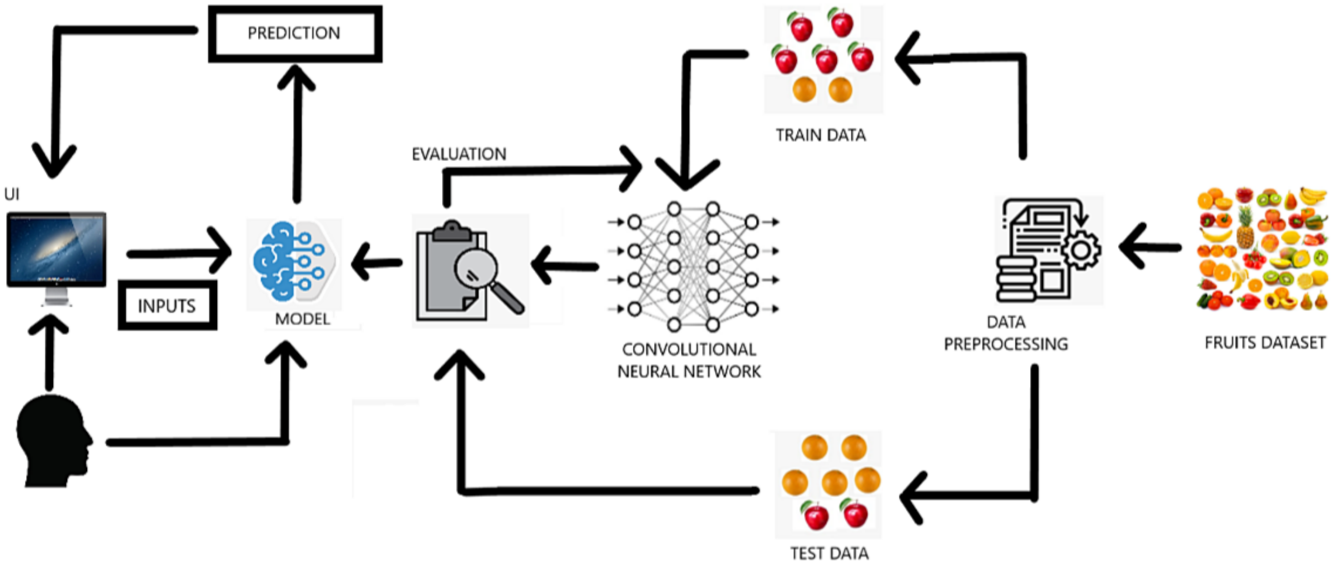
# 5.PROJECT DESIGN

- Data Flow Diagram

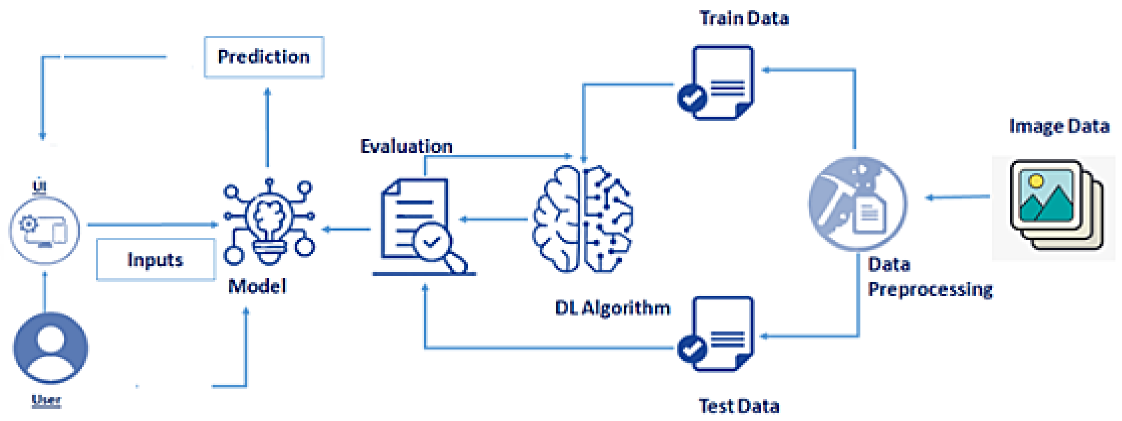
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored



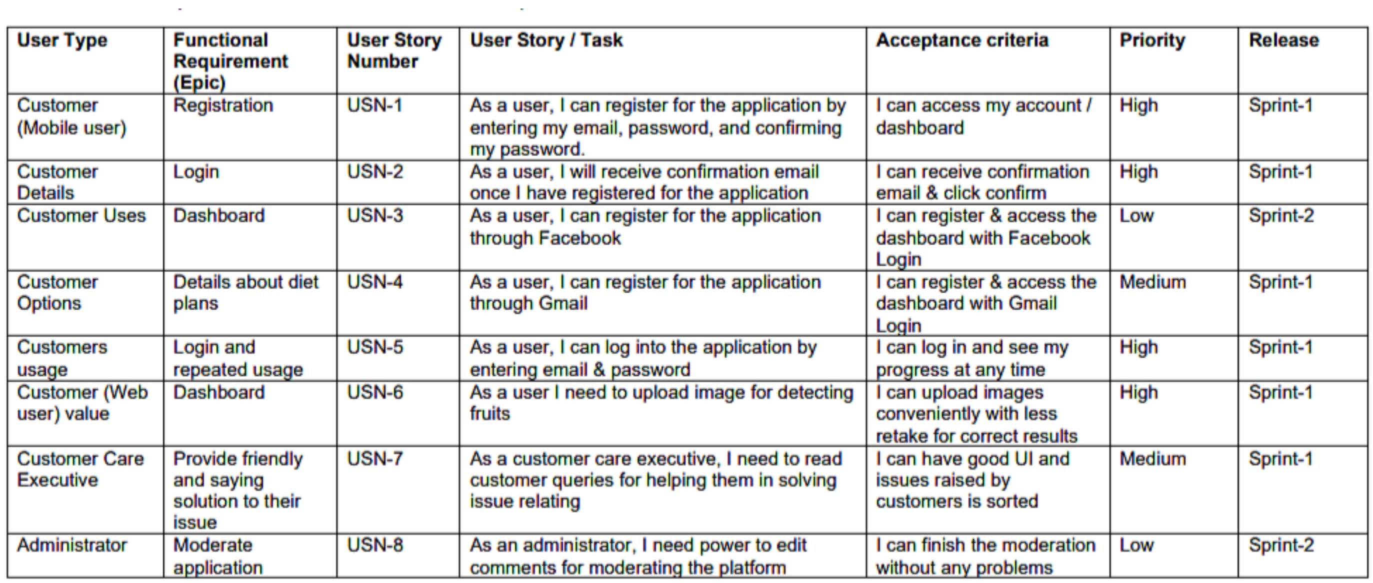
- Solution Architecture



- Technical Architecture



- User Stories



| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Customer (Mobile user) | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can access my account / dashboard | High | Sprint-1 |
| Customer Details | Login | USN-2 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmation email & click confirm | High | Sprint-1 |
| Customer Uses | Dashboard | USN-3 | As a user, I can register for the application through Facebook | I can register & access the dashboard with Facebook Login | Low | Sprint-2 |
| Customer Options | Details about diet plans | USN-4 | As a user, I can register for the application through Gmail | I can register & access the dashboard with Gmail Login | Medium | Sprint-1 |
| Customers usage | Login and repeated usage | USN-5 | As a user, I can log into the application by entering email & password | I can log in and see my progress at any time | High | Sprint-1 |
| Customer (Web user) value | Dashboard | USN-6 | As a user I need to upload image for detecting fruits | I can upload images conveniently with less retake for correct results | High | Sprint-1 |
| Customer Care Executive | Provide friendly and saying solution to their issue | USN-7 | As a customer care executive, I need to read customer queries for helping them in solving issue relating | I can have good UI and issues raised by customers is sorted | Medium | Sprint-1 |
| Administrator | Moderate application | USN-8 | As an administrator, I need power to edit comments for moderating the platform | I can finish the moderation without any problems | Low | Sprint-2 |

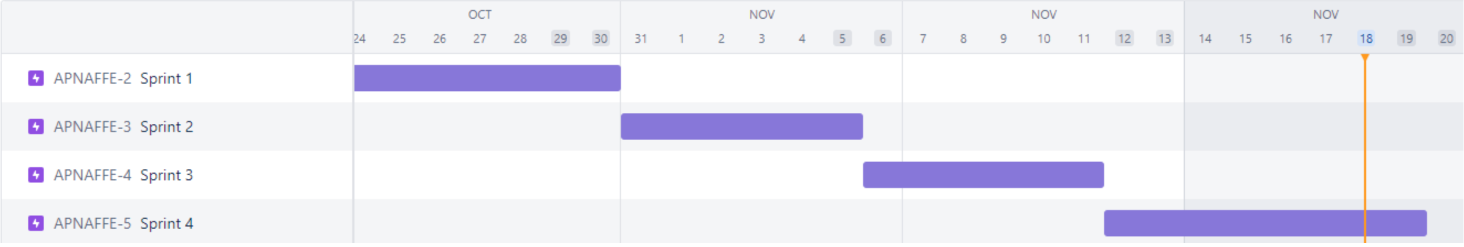
## 6.PROJECT PLANNING & SCHEDULING

- Sprint Planning & Estimation

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Dataset collection | USN-1 | Image collection of food items especially fruits | 3 | Medium | Suriya N<br>Palaniappan S |
| Sprint-1 | Image Preprocessing | USN-2 | Image augmentation | 5 | High | Suriya N<br>Palaniappan S |
| Sprint-1 | Development phase | USN-7 | Creation of the home page | 4 | Low | Pranav Sreenivas R<br>Enesh Naren A |
| Sprint-1 | Application Phase | USN-11 | Importing the flask module | 2 | Medium | Pranav Sreenivas R Enesh Naren A |
| Sprint-2 | Image Preprocessing | USN-3 | Applying image augmentation to the train set and test set | 4 | Low | Suriya N<br>Palaniappan S |
| Sprint-2 | Modeling | USN-4 | Defining the model architecture | 5 | Medium | Suriya N<br>Palaniappan S |
| Sprint-2 | Modeling | USN-6 | Create database | 2 | Medium | Prana Sreenivas R<br>Enesh Naren A |
| Sprint-2 | Development phase | USN-8 | Login and registration page creation | 3 | Medium | Pranav Sreenivas R<br>Enesh Naren A |
| Sprint-3 | Modeling | USN-5 | Adding CNN, dense layers and testing, saving the model | 8 | High | Suriya N<br>Palaniappan S |
| Sprint-3 | Development phase | USN-9 | User input page creation | 4 | Low | Pranav Sreenivas R Enesh Naren A |
| Sprint-3 | Development phase | USN-10 | Creation of rating and feedback page | 3 | Medium | Pranav Sreenivas R Enesh Naren A |
| Sprint-4 | Application Phase | USN-12 | Loading the model by creating flask | 4 | High | Pranav Sreenivas R Enesh Naren A |
| Sprint-4 | Application | USN-13 | API-integration |  | High | Suriya N |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Phase | | | 5 | | Palaniappan S<br>Pranav Sreenivas R<br>Enesh Naren A |
| Sprint-4 | Deployment Phase | USN-14 | Cloud deployment | 3 | Low | Suriya N<br>Palaniappan S<br>Pranav Sreenivas R<br>Enesh Naren A |
| Sprint-4 | Deployment Phase | USN-15 | Check scalability, usability and performance | 5 | Medium | Suriya N<br>Palaniappan S<br>Pranav Sreenivas R Enesh Naren A |

- Sprint Roadmap:

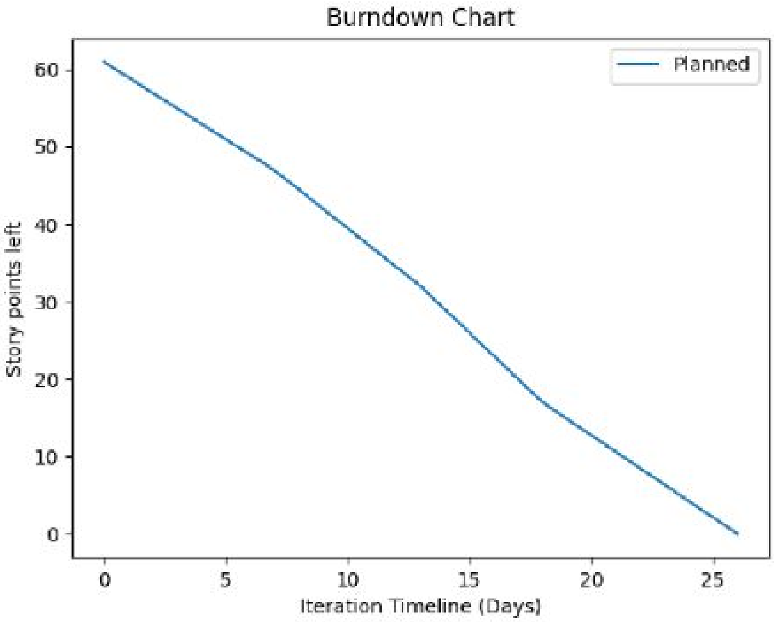


APNAFFE-2 Sprint 1

APNAFFE-3 Sprint 2

APNAFFE-4 Sprint 3

APNAFFE-5 Sprint 4



## 7.CODING & SOLUTIONING:

The main aim for this project is to identify the fruit and detect the number of calories present in it. Along with it, we have added some features that enhance the user experience.

### Feature 1: -

Along with the number of calories, there are several other nutrient parameters. All the parameters are: - name, serving, calories, fiber, fat, protein, water, sugar, carbohydrates, food type.

```
import csv
def index_to_information (index_of_pred_class : int,files_name_list: list) -> list:
    '''
    This function takes the value of predicted class index from CNN model file
    And returns information about that file in files_name_list (csv)'s as Dictionary '''

    #initialize empty dictonary
    info_dict = {}

    # index to food name
                food_name_of_pred_class = csv_extractor (files_name_list[0],
str(index_of_pred_class), row_number=1)["COMMON TERM"]
    print(food_name_of_pred_class)

    # food name to extract information in list of other files
    for file_name in files_name_list[1:]:
        info_dict.update(csv_extractor (file_name, food_name_of_pred_class))
    return info_dict



def csv_extractor (csv_file_name : str, food_name : int, row_number :int = 0):
    '''
    This function takes in a csv file name, a food name, and a row number.
    The row number is optional and defaults to 0.
    The function returns a dictionary with the food name as the key and the
    corresponding row as the value.
    '''

    with open(csv_file_name, 'r') as file:
        reader = csv.reader(file)
        head = next(reader)
        for row in reader:
            if row[row_number].lower() == food_name.lower():
                return dict(zip(head,row))
        else:
            return {}
```

## Feature 2: -

Some additional information are also provided on the various market prices of the fruit available. The considered market prices are: - WHOLESALE PRICE, RETAIL PRICE, SHOPPING MALL PRICE.

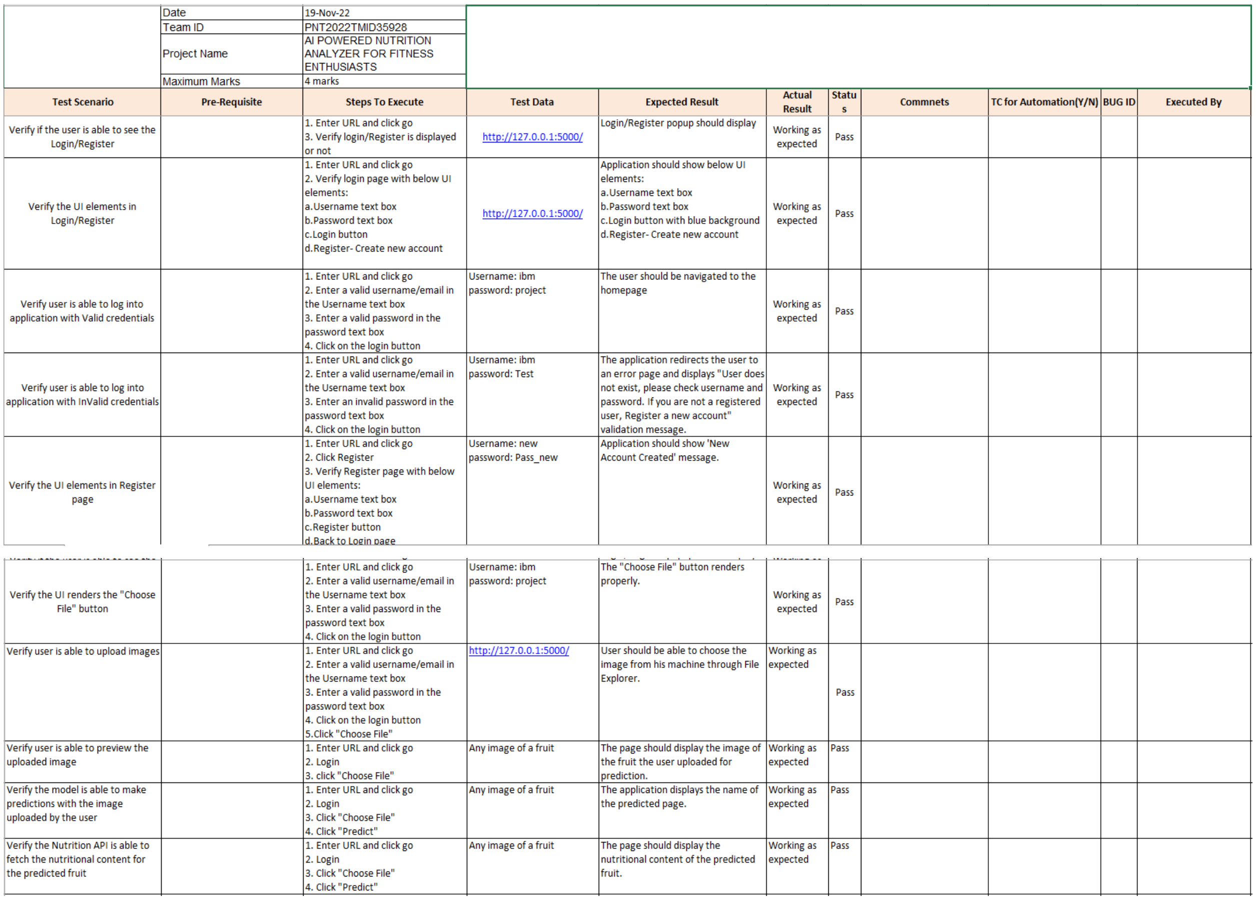
CODE:-

```
#libraries
import requests
from bs4 import BeautifulSoup
import pandas as pd

# predefined lists
veg_list =
["Beetroot","Cauliflower","Corn","Cucumber","Brinjal","Ginger","Lemon","Mango
Raw","Onion Big","Potato","Sweet Potato","Tomato"]
fruits_list = ["Apple
Shimla","Banana","Cantaloupe","Guava","Orange","Papaya","Pineapple","Pomegran
ate Kabul","Sapota","Watermelon"]
database = []
save_address = "../csv/price_list.csv"

#functions to webscrap to list
web_scrapper("https://vegetablemarketprice.com/market/chennai/today",
      veg_list,
      database,
      food_type = "vegetable")
```

```
web_scrapper("https://vegetablemarketprice.com/fruits/tamilNadu/today",
       fruits_list,
       database,
       food_type = "fruits")

#Converting list to Pandas DataFrame and saving the file as csv
df = pd.DataFrame(data = database)
df.columns = ["NAME","WHOLESALE PRICE", "RETAIL PRICE","SHOPPING MALL
PRICE","QUANTITY","TYPE"]
df.to_csv(save_address, header = True, index=False)

def web_scrapper(web_link: str, target_row_name_list : list, database_arr : list,
food_type : str = "NA" ):
    """Function which is used web_scrapper required content from web_link
       Select appropriate rows from target_row_name_list
       and appends it to database_arr list with food_type (default value = NA) """

    request_reply = requests.get(web_link)
    soup = BeautifulSoup(request_reply.content, 'html5lib')
    table = soup.find('table', attrs={"class":"table"})
    table_rows = table.find_all("tr")
    for tr in table_rows:
        td = tr.find_all("td")
        row = [i.text for i in td]
        if (len(row)):
            if (row[1].split("(")[0].strip() in target_row_name_list):
                database_arr.append(format_rows(row,food_type = food_type))
    return 0

def name_formatter (message : str):
    """
    This function to used to format string to remove redudant information
    """
        return message.replace(message[message.rfind("(") : message.rfind(")") + 1],
"").strip()

def price_list_extractor (message : str):
    """
    Extracts the price range from a message.
     :param message: The message to extract the price range from which is in the for
"₹low_price - ₹high_price".
    :return: A list of two integers, the low and high price."""

    low , high = message[1:].split(" - ")
    return [int(low), int(high)]

def format_rows (row_content : list, food_type = "NA" ):
    """
    This function takes in a list of strings and returns a list of strings.
    The input list is a row of data from the webscrapped file.
    The output list is a row of data that is formatted for the database.
    The function does the following:
        1. Formats the name of the fruit.
        2. Formats the wholesale price.
        3. Formats the retail price.
        4. Formats the shopping mall price.
        5. Formats the quantity. """
    name_of_foodtype = name_formatter(row_content[1])
    wholesale_price = int(row_content[2][1:])
    retail_price = price_list_extractor (row_content[3])
    shopping_mall_price = price_list_extractor (row_content[4])
    quantity = row_content[5]

    return [name_of_foodtype,
            wholesale_price,
            retail_price,
            shopping_mall_price,
            quantity,
            food_type]
```

# 8. Testing

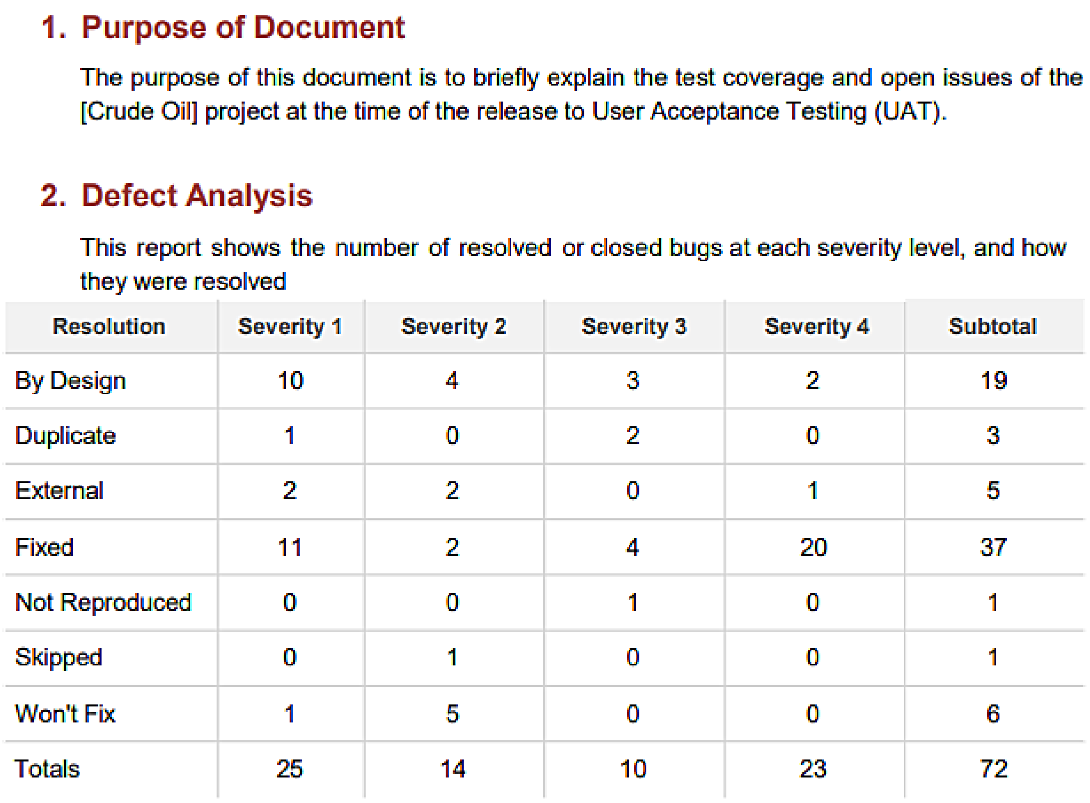
- Test Cases Report



| | |
|---|---|
| Date | 19-Nov-22 |
| Team ID | PNT2022TMID35928 |
| Project Name | AI POWERED NUTRITION ANALYZER FOR FITNESS ENTHUSIASTS |
| Maximum Marks | 4 marks |

| Test Scenario | Pre-Requisite | Steps To Execute | Test Data | Expected Result | Actual Result | Status | Commnets | TC for Automation(Y/N) | BUG ID | Executed By |
|---|---|---|---|---|---|---|---|---|---|---|
| Verify if the user is able to see the Login/Register | | 1. Enter URL and click go<br>3. Verify login/Register is displayed or not | http://127.0.0.1:5000/ | Login/Register popup should display | Working as expected | Pass | | | | |
| Verify the UI elements in Login/Register | | 1. Enter URL and click go<br>2. Verify login page with below UI elements:<br>a.Username text box<br>b.Password text box<br>c.Login button<br>d.Register- Create new account | http://127.0.0.1:5000/ | Application should show below UI elements:<br>a.Username text box<br>b.Password text box<br>c.Login button with blue background<br>d.Register- Create new account | Working as expected | Pass | | | | |
| Verify user is able to log into application with Valid credentials | | 1. Enter URL and click go<br>2. Enter a valid username/email in the Username text box<br>3. Enter a valid password in the password text box<br>4. Click on the login button | Username: ibm<br>password: project | The user should be navigated to the homepage | Working as expected | Pass | | | | |
| Verify user is able to log into application with InValid credentials | | 1. Enter URL and click go<br>2. Enter a valid username/email in the Username text box<br>3. Enter an invalid password in the password text box<br>4. Click on the login button | Username: ibm<br>password: Test | The application redirects the user to an error page and displays "User does not exist, please check username and password. If you are not a registered user, Register a new account" validation message. | Working as expected | Pass | | | | |
| Verify the UI elements in Register page | | 1. Enter URL and click go<br>2. Click Register<br>3. Verify Register page with below UI elements:<br>a.Username text box<br>b.Password text box<br>c.Register button<br>d.Back to Login page | Username: new<br>password: Pass_new | Application should show 'New Account Created' message. | Working as expected | Pass | | | | |
| Verify the UI renders the "Choose File" button | | 1. Enter URL and click go<br>2. Enter a valid username/email in the Username text box<br>3. Enter a valid password in the password text box<br>4. Click on the login button | Username: ibm<br>password: project | The "Choose File" button renders properly. | Working as expected | Pass | | | | |
| Verify user is able to upload images | | 1. Enter URL and click go<br>2. Enter a valid username/email in the Username text box<br>3. Enter a valid password in the password text box<br>4. Click on the login button<br>5.Click "Choose File" | http://127.0.0.1:5000/ | User should be able to choose the image from his machine through File Explorer. | Working as expected | Pass | | | | |
| Verify user is able to preview the uploaded image | | 1. Enter URL and click go<br>2. Login<br>3. click "Choose File" | Any image of a fruit | The page should display the image of the fruit the user uploaded for prediction. | Working as expected | Pass | | | | |
| Verify the model is able to make predictions with the image uploaded by the user | | 1. Enter URL and click go<br>2. Login<br>3. Click "Choose File"<br>4. Click "Predict" | Any image of a fruit | The application displays the name of the predicted page. | Working as expected | Pass | | | | |
| Verify the Nutrition API is able to fetch the nutritional content for the predicted fruit | | 1. Enter URL and click go<br>2. Login<br>3. Click "Choose File"<br>4. Click "Predict" | Any image of a fruit | The page should display the nutritional content of the predicted fruit. | Working as expected | Pass | | | | |

- User Acceptance Testing



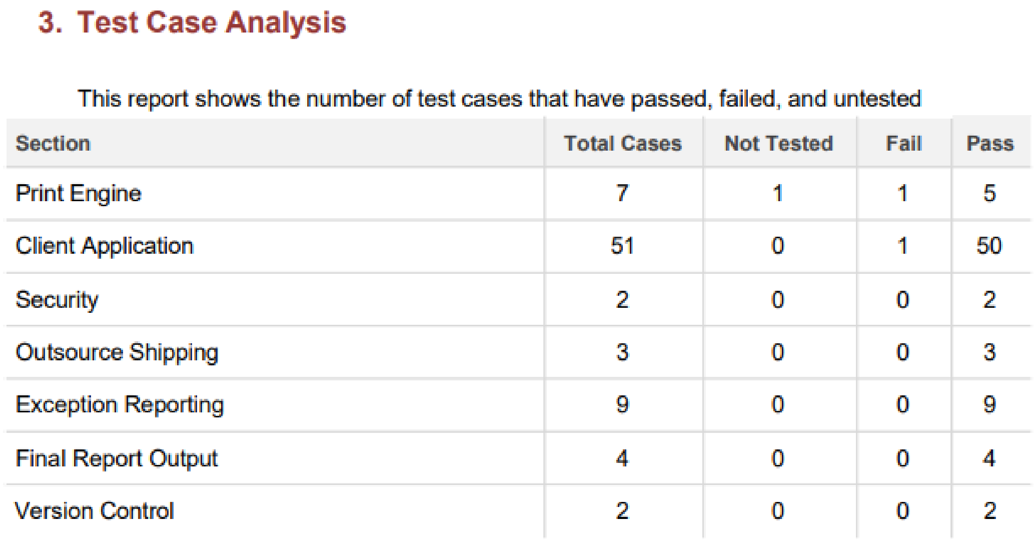
## 1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Crude Oil] project at the time of the release to User Acceptance Testing (UAT).

## 2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

| Resolution | Severity 1 | Severity 2 | Severity 3 | Severity 4 | Subtotal |
|---|---|---|---|---|---|
| By Design | 10 | 4 | 3 | 2 | 19 |
| Duplicate | 1 | 0 | 2 | 0 | 3 |
| External | 2 | 2 | 0 | 1 | 5 |
| Fixed | 11 | 2 | 4 | 20 | 37 |
| Not Reproduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 0 | 1 | 0 | 0 | 1 |
| Won't Fix | 1 | 5 | 0 | 0 | 6 |
| Totals | 25 | 14 | 10 | 23 | 72 |



## 3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

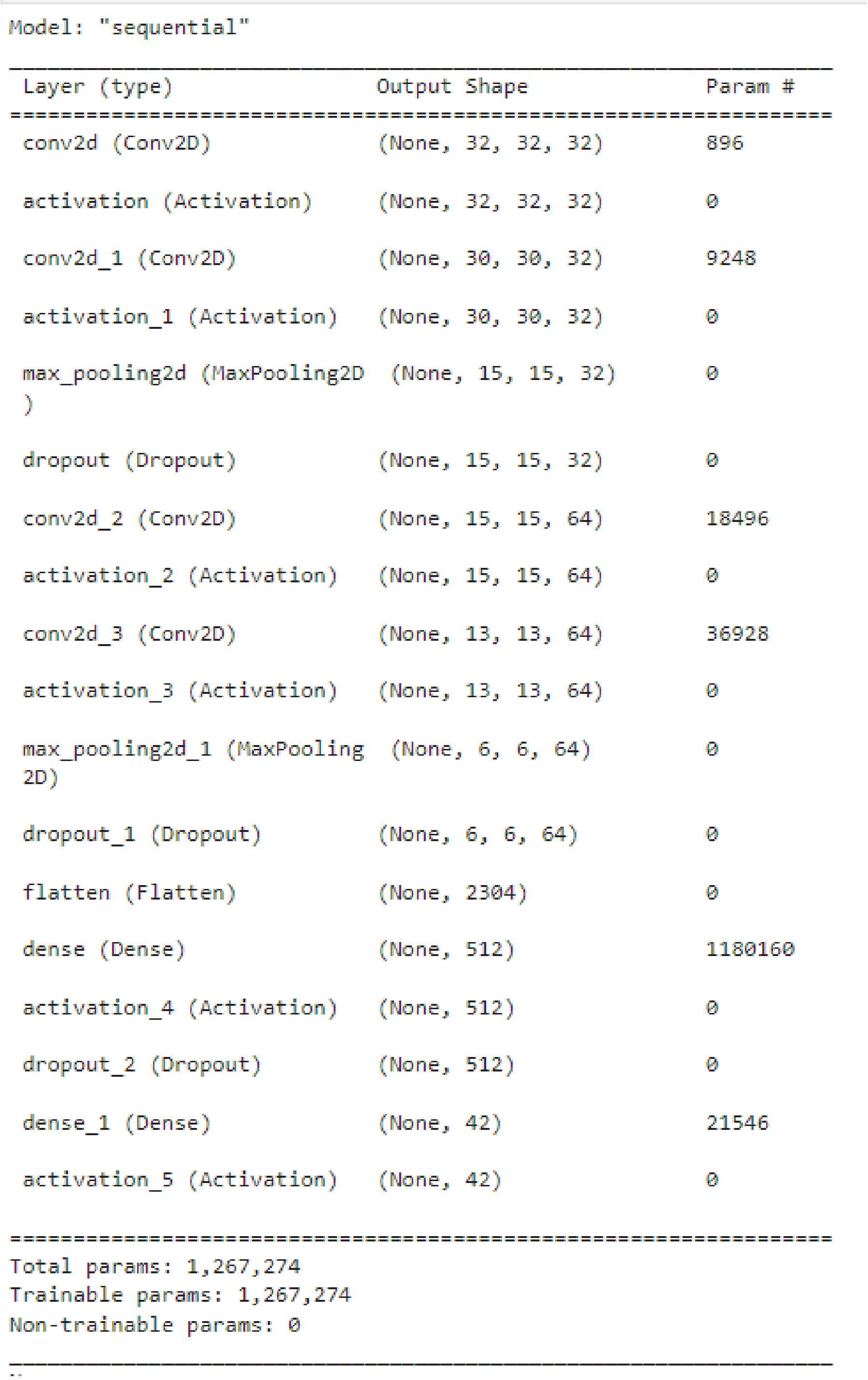
| Section | Total Cases | Not Tested | Fail | Pass |
|---|---|---|---|---|
| Print Engine | 7 | 1 | 1 | 5 |
| Client Application | 51 | 0 | 1 | 50 |
| Security | 2 | 0 | 0 | 2 |
| Outsource Shipping | 3 | 0 | 0 | 3 |
| Exception Reporting | 9 | 0 | 0 | 9 |
| Final Report Output | 4 | 0 | 0 | 4 |
| Version Control | 2 | 0 | 0 | 2 |

# 9.RESULTS

- Performance Testing

Model Summary :

Screenshot:



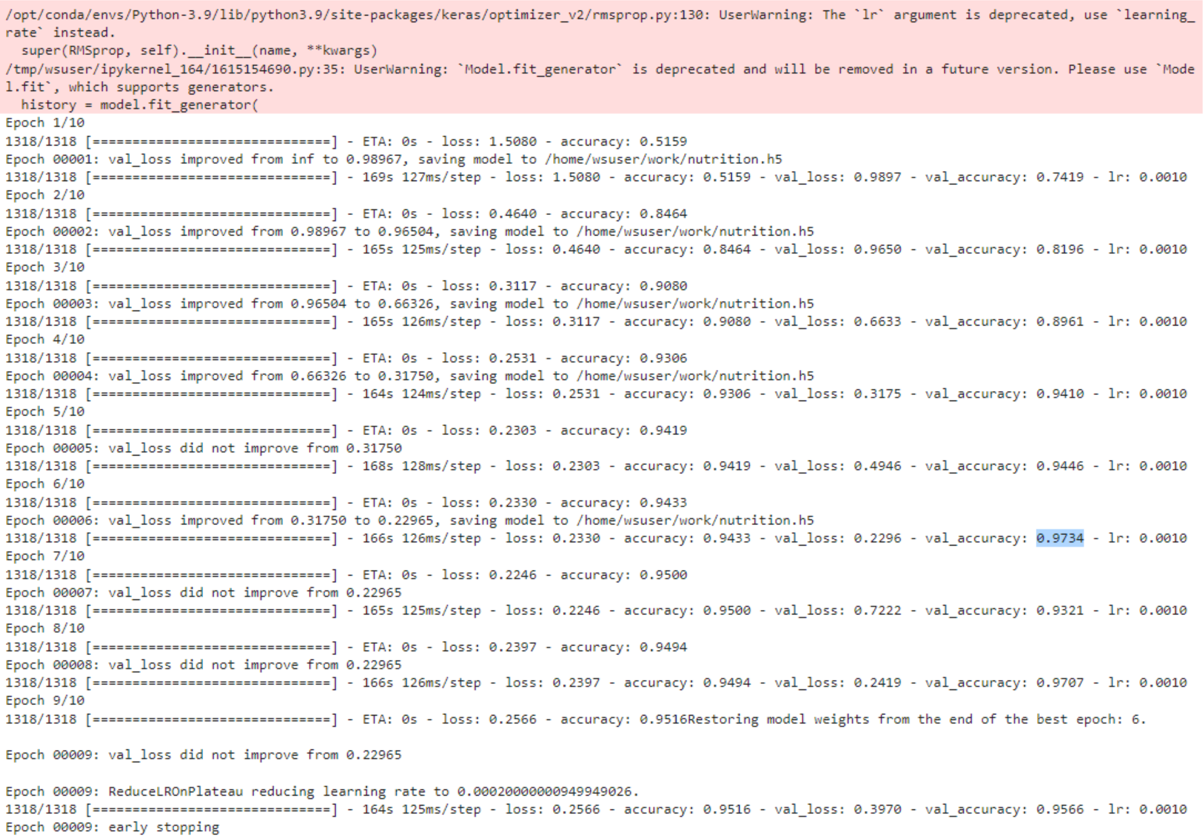
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 32)        896

 activation (Activation)     (None, 32, 32, 32)        0

 conv2d_1 (Conv2D)           (None, 30, 30, 32)        9248

 activation_1 (Activation)   (None, 30, 30, 32)        0

 max_pooling2d (MaxPooling2D  (None, 15, 15, 32)       0
 )

 dropout (Dropout)           (None, 15, 15, 32)        0

 conv2d_2 (Conv2D)           (None, 15, 15, 64)        18496

 activation_2 (Activation)   (None, 15, 15, 64)        0

 conv2d_3 (Conv2D)           (None, 13, 13, 64)        36928

 activation_3 (Activation)   (None, 13, 13, 64)        0

 max_pooling2d_1 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 dropout_1 (Dropout)         (None, 6, 6, 64)          0

 flatten (Flatten)           (None, 2304)              0

 dense (Dense)               (None, 512)               1180160

 activation_4 (Activation)   (None, 512)               0

 dropout_2 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 42)                21546

 activation_5 (Activation)   (None, 42)                0

=================================================================
Total params: 1,267,274
Trainable params: 1,267,274
Non-trainable params: 0
_________________________________________________________________
```

## Accuracy :

Training Accuracy : 94.33 %

Validation Accuracy : 97.34 %

## Screenshot :



```
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/keras/optimizer_v2/rmsprop.py:130: UserWarning: The `lr` argument is deprecated, use `learning_
rate` instead.
  super(RMSprop, self).__init__(name, **kwargs)
/tmp/wsuser/ipykernel_164/1615154690.py:35: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Mode
l.fit`, which supports generators.
  history = model.fit_generator(
Epoch 1/10
1318/1318 [==============================] - ETA: 0s - loss: 1.5080 - accuracy: 0.5159
Epoch 00001: val_loss improved from inf to 0.98967, saving model to /home/wsuser/work/nutrition.h5
1318/1318 [==============================] - 169s 127ms/step - loss: 1.5080 - accuracy: 0.5159 - val_loss: 0.9897 - val_accuracy: 0.7419 - lr: 0.0010
Epoch 2/10
1318/1318 [==============================] - ETA: 0s - loss: 0.4640 - accuracy: 0.8464
Epoch 00002: val_loss improved from 0.98967 to 0.96504, saving model to /home/wsuser/work/nutrition.h5
1318/1318 [==============================] - 165s 125ms/step - loss: 0.4640 - accuracy: 0.8464 - val_loss: 0.9650 - val_accuracy: 0.8196 - lr: 0.0010
Epoch 3/10
1318/1318 [==============================] - ETA: 0s - loss: 0.3117 - accuracy: 0.9080
Epoch 00003: val_loss improved from 0.96504 to 0.66326, saving model to /home/wsuser/work/nutrition.h5
1318/1318 [==============================] - 165s 126ms/step - loss: 0.3117 - accuracy: 0.9080 - val_loss: 0.6633 - val_accuracy: 0.8961 - lr: 0.0010
Epoch 4/10
1318/1318 [==============================] - ETA: 0s - loss: 0.2531 - accuracy: 0.9306
Epoch 00004: val_loss improved from 0.66326 to 0.31750, saving model to /home/wsuser/work/nutrition.h5
1318/1318 [==============================] - 164s 124ms/step - loss: 0.2531 - accuracy: 0.9306 - val_loss: 0.3175 - val_accuracy: 0.9410 - lr: 0.0010
Epoch 5/10
1318/1318 [==============================] - ETA: 0s - loss: 0.2303 - accuracy: 0.9419
Epoch 00005: val_loss did not improve from 0.31750
1318/1318 [==============================] - 168s 128ms/step - loss: 0.2303 - accuracy: 0.9419 - val_loss: 0.4946 - val_accuracy: 0.9446 - lr: 0.0010
Epoch 6/10
1318/1318 [==============================] - ETA: 0s - loss: 0.2330 - accuracy: 0.9433
Epoch 00006: val_loss improved from 0.31750 to 0.22965, saving model to /home/wsuser/work/nutrition.h5
1318/1318 [==============================] - 166s 126ms/step - loss: 0.2330 - accuracy: 0.9433 - val_loss: 0.2296 - val_accuracy: 0.9734 - lr: 0.0010
Epoch 7/10
1318/1318 [==============================] - ETA: 0s - loss: 0.2246 - accuracy: 0.9500
Epoch 00007: val_loss did not improve from 0.22965
1318/1318 [==============================] - 165s 125ms/step - loss: 0.2246 - accuracy: 0.9500 - val_loss: 0.7222 - val_accuracy: 0.9321 - lr: 0.0010
Epoch 8/10
1318/1318 [==============================] - ETA: 0s - loss: 0.2397 - accuracy: 0.9494
Epoch 00008: val_loss did not improve from 0.22965
1318/1318 [==============================] - 166s 126ms/step - loss: 0.2397 - accuracy: 0.9494 - val_loss: 0.2419 - val_accuracy: 0.9707 - lr: 0.0010
Epoch 9/10
1318/1318 [==============================] - ETA: 0s - loss: 0.2566 - accuracy: 0.9516Restoring model weights from the end of the best epoch: 6.

Epoch 00009: val_loss did not improve from 0.22965

Epoch 00009: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
1318/1318 [==============================] - 164s 125ms/step - loss: 0.2566 - accuracy: 0.9516 - val_loss: 0.3970 - val_accuracy: 0.9566 - lr: 0.0010
Epoch 00009: early stopping
```

# 10.ADVANTAGES & DISADVANTAGES

## Advantages:

Nowadays new dietary assessment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and maintain a healthy diet.

Nutritional analysis is the process of determining the nutritional content of food.

AI algorithms may help better understand and predict the complex and non-linear interactions between nutrition-related data and health outcomes, particularly when large amounts of data need to be structured and integrated, such as in metabolomics.

AI-based approaches, including image recognition, may also improve dietary assessment by maximizing efficiency and addressing systematic and random errors associated with self-reported measurements of dietary intakes.

Finally, AI applications can extract, structure and analyze large amounts of data from social media platforms to better understand dietary behaviours and perceptions among the population.

In summary, AI-based approaches will likely improve and advance nutrition research as well as help explore new applications

## Disadvantages:

Cannot further classify the main classes of fruits into sub categories.

Cannot distinguish between rotten or fresh fruits.

Heavy reliance on AI,Technology which leads to increased screentime.

## 11.CONCLUSION:

Thus we have built a model which is used for classifying the fruit depending on the different characteristics like colour, shape, texture etc. . The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).

## 12. FUTURE SCOPE:-

We can add datasets of rotten fruits and predict whether the fruit

TCreate customized diet plan for user based on their preference, allegeries, cost and ordering foods through grocery store API .

# 13.APPENDIX:

SOURCE CODE :-

HTML FILES:-

index.html :-

```
<html lang="en">


<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <title>AI-Powered Nutrition Analyzer For Fitness Enthusiasts</title>
   <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
   <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
   <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
   <script
src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
   <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
            <style>


            .bg-dark {
                        background-color:          green!important;
            }
            #result {
                        color: #ffffff;
```

```
                }
                body
{
    background-image: url("https://wallpaperaccess.com/thumb/826450.jpg");
    background-size: cover;
}


                </style>
</head>


<body>


    <nav class="navbar navbar-dark bg-dark">
        <div class="container">
            <a class="navbar-brand" href="#">AI-Powered Nutrition Analyzer For Fitness Enthusiasts</a>
        </div>
    </nav>
    <div class="container">
        <div id="content" style="margin-top:2em">
                                <div class="container">
                                  <div class="row">
                                                <div class="col-sm-6 bd">
                                                  <h3>Fruits Classification: </h3>
                                                  <br>
                                                  <p>Project Description : This Project uses Convolution Neural Network to predict the name of the fruit, whose photo has been uploaded.</p>
                <img src="https://media.istockphoto.com/photos/fruit-background-picture-id529664572?s=612x612" width="300" height="300">
            </div>
            <div class="col-sm-6">
                <div>
                    <h4>Upload Image</h4>
            <form action = "http://localhost:5000/" id="upload-file" method="post" enctype="multipart/form-data">
                <label for="imageUpload" class="upload-label">
                    Choose...
                </label>
                <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">
            </form>


            <div class="image-section" style="display:none;">
                <div class="img-preview">
                    <div id="imagePreview">
                    </div>
                </div>
                <div>
                    <button type="button" class="btn btn-info btn-lg " id="btn-predict">Predict</button>
                </div>
            </div>
```

```
                                                <div class="loader" style="display:none;"></div>


                                                <h3>
                                                                <span id="result"> </span>
                                                </h3>


                                <div>
                                                </div>


                                  </div>
                                </div>
                                </div>
    </div>
</body>


<footer>
    <script src="{{ url_for('static', filename='js/main.js') }}"
type="text/javascript"></script>
</footer>


</html>
```

error.html :-

```
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
   <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <title>AI-Powered Nutrition Analyzer For Fitness Enthusiasts</title>
   <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
   <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
   <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
   <script
src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
   <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
            <style>

            .bg-dark {
                        background-color:         gray!important;
            }
            #result {
                        color: #ffffff;
            }
            body
{
   background-image: url("https://wallpaperaccess.com/thumb/826450.jpg");
   background-size: cover;
}

            </style>
</head>

<body>
```

```
    <nav class="navbar navbar-dark bg-success">
        <div class="container">
            <a class="navbar-brand" href="#">AI-Powered Nutrition Analyzer For Fitness
Enthusiasts</a>
        </div>
    </nav>
    <div class="container">
        <div id="content" style="margin-top:3em"
                                <div class="container">
                                  <div class="row">
                                                <div class="col-sm-6 bd">
                                                  <h3>Error 404: </h3>
                                                  <br>
                                                  <p>

User does not exist, please check username and password. If you are not a
registered user, Register a new account.</p>
                                                                <img
src="https://cdni.iconscout.com/illustration/premium/thumb/404-error-message-
3702341-3119133.png" width="500" height="300">
                                                </div>
                                                <div class="col-sm-6">
                                                                <div>




                                                <div class="image-section" style="display:none;">
                                <div class="img-preview">
                                        <div id="imagePreview">
                                        </div>
                                </div>
                                <div>
                                        <button type="button" class="btn btn-info btn-lg " id="btn-predict">Predict!</button>
                                </div>
                        </div>

                        <div class="loader" style="display:none;"></div>

                        <h3>
                                <span id="Result"> </span>
                        </h3>

                </div>
                        </div>

                  </div>
                </div>
                </div>
    </div>
</body>

<footer>
    <script src="{{ url_for('static', filename='js/main.js') }}"
```

```
type="text/javascript"></script>
</footer>

</html>
```

-

-

-

-

login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

<style>
```

```
body {font-family: Arial, Helvetica, sans-serif;     background-image:
url("https://wallpaperaccess.com/thumb/826450.jpg");
    background-size: cover;}
form {border: 0px solid #f1f1f1;}

input[type=text], input[type=password] {
  width: 20%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

button {
  background-color: #048caa;
  color: white;
  padding: 14px 20px;
  margin: 0 0 10px 100px;
  border: none;
  cursor: pointer;
  width: 20%;
}

button:hover {
  opacity: 0.8;
}
```

```
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

.imgcontainer {
  text-align: center;
  margin: 45px 0 60px 0;
}

img.avatar {


  border-radius: 10%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
```

```
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 20%;
  }
}

</style>
</head>
<body>

    <nav class="navbar navbar-dark bg-success">
        <div class="container">
            <a class="navbar-brand" href="#">AI-Powered Nutrition Analyzer For Fitness Enthusiasts - Register</a>
        </div>
<a href="/">Back to login page</a>
    </nav>

<form method="post">
  <div class="imgcontainer">
    <img src="https://media.istockphoto.com/vectors/user-icon-flat-isolated-on-white-background-user-symbol-vector-vector-id1300845620?k=20&m=1300845620&s=612x612&w=0&h=f4XTZDAv7NPuZbG0habSpU0sNgECM0X7nbKzTUta3n8=" alt="Avatar" class="avatar" width="135"
```

```
height="175">
  </div>

  <div class="container">
    <label for="uname"><b>Username</b></label>
    <input type="text" placeholder="Enter Username" name="user_name" required>

    <label for="psw"><b>Password</b></label>
    <input type="password" placeholder="Enter Password" name="password"
required>

    <button type="submit">Register</button>
    <!--<label>
      <input type="checkbox" checked="checked" name="remember"> Remember me
    </label>
  </div>

  <div class="container" >
    <button type="button" class="cancelbtn">Cancel</button>
    <span class="psw">Forgot <a href="#">password?</a></span>
  </div>-->
</form>

</body>
</html>
```

-

-

-

-

register.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

<style>



body {font-family: Arial, Helvetica, sans-serif;     background-image: url("https://wallpaperaccess.com/thumb/826450.jpg");
    background-size: cover;}
form {border: 0px solid #f1f1f1;}

input[type=text], input[type=password] {
  width: 20%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

button {
  background-color: #048caa;
  color: white;
  padding: 14px 20px;
  margin: 0 0 10px 100px;
  border: none;
  cursor: pointer;
  width: 20%;
}

button:hover {
  opacity: 0.8;
}

.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}
```

```
.imgcontainer {
  text-align: center;
  margin: 45px 0 60px 0;
}

img.avatar {


  border-radius: 10%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 20%;
  }
}

</style>
</head>
<body>

    <nav class="navbar navbar-dark bg-success">
        <div class="container">
            <a class="navbar-brand" href="#">AI-Powered Nutrition Analyzer For Fitness Enthusiasts - Register</a>
        </div>
<a href="/">Back to login page</a>
    </nav>

<form method="post">
  <div class="imgcontainer">
    <img src="https://media.istockphoto.com/vectors/user-icon-flat-isolated-on-white-background-user-symbol-vector-vector-id1300845620?k=20&m=1300845620&s=612x612&w=0&h=f4XTZDAv7NPuZbG0habSpU0sNgECM0X7nbKzTUta3n8=" alt="Avatar" class="avatar" width="135" height="175">
  </div>

  <div class="container">
    <label for="uname"><b>Username</b></label>
    <input type="text" placeholder="Enter Username" name="user_name" required>
    <label for="psw"><b>Password</b></label>
    <input type="password" placeholder="Enter Password" name="password" required>


    <button type="submit">Register</button>
    <!--<label>
      <input type="checkbox" checked="checked" name="remember"> Remember me
    </label>
  </div>


  <div class="container" >
    <button type="button" class="cancelbtn">Cancel</button>
    <span class="psw">Forgot <a href="#">password?</a></span>
  </div>-->
</form>


</body>
</html>
```

-

-

-

## PYTHON FILES:

File_Extractor.py

```
import csv


def index_to_information (index_of_pred_class : int,files_name_list: list) -> list:
    '''
    This function takes the value of predicted class index from CNN model file
    And returns information about that file in files_name_list (csv)'s as Dictionary '''

    #initialize empty dictonary
    info_dict = {}

    # index to food name
    food_name_of_pred_class = csv_extractor (files_name_list[0],
str(index_of_pred_class), row_number=1)["COMMON TERM"]
    print(food_name_of_pred_class)

    # food name to extract information in list of other files
    for file_name in files_name_list[1:]:
        info_dict.update(csv_extractor (file_name, food_name_of_pred_class))
    return info_dict



def csv_extractor (csv_file_name : str, food_name : int, row_number :int = 0):
    '''
    This function takes in a csv file name, a food name, and a row number.
    The row number is optional and defaults to 0.
    The function returns a dictionary with the food name as the key and the
    corresponding row as the value.
    '''
    with open(csv_file_name, 'r') as file:
        reader = csv.reader(file)
        head = next(reader)
        for row in reader:
            if row[row_number].lower() == food_name.lower():
                return dict(zip(head,row))
        else:
            return {}
```

WebScraping.py

```
#libraries
import requests
from bs4 import BeautifulSoup
import pandas as pd
import os
import time

SECONDS_IN_DAY = 86400

def last_modified_csv(filename) -> int:
    if (os.path.exists(filename)):
        modified_time = os.path.getmtime(filename)
        now = time.time()
        modified_interval = now - modified_time
        decision = modified_interval > SECONDS_IN_DAY
    else:
        decision = 1
    return decision
```

```
def web_scrapper(web_link: str, target_row_name_list : list, database_arr : list,
food_type : str = "NA" ):
    """Function which is used web_scrapper required content from web_link
       Select appropriate rows from target_row_name_list
       and appends it to database_arr list with food_type (default value = NA) """


    request_reply = requests.get(web_link)
    soup = BeautifulSoup(request_reply.content, 'html5lib')
    table = soup.find('table', attrs={"class":"table"})
    table_rows = table.find_all("tr")
    for tr in table_rows:
        td = tr.find_all("td")
        row = [i.text for i in td]
        if (len(row)):
            if (row[1].split("(")[0].strip() in target_row_name_list):
                database_arr.append(format_rows(row,food_type = food_type))
    return 0


def name_formatter (message : str):
    """
    This function to used to format string to remove redudant information
    """
    return message.replace(message[message.rfind("(") : message.rfind(")") + 1],
"").strip()
```

```
def price_list_extractor (message : str):
    """
    Extracts the price range from a message.
    :param message: The message to extract the price range from which is in the for "₹low_price - ₹high_price".
    :return: A list of two integers, the low and high price."""

    low , high = message[1:].split(" - ")
    return [int(low), int(high)]


def format_rows (row_content : list, food_type = "NA" ):
    """
    This function takes in a list of strings and returns a list of strings.
    The input list is a row of data from the webscrapped file.
    The output list is a row of data that is formatted for the database.
    The function does the following:
        1. Formats the name of the fruit.
        2. Formats the wholesale price.
        3. Formats the retail price.
        4. Formats the shopping mall price.
        5. Formats the quantity. """
    name_of_foodtype = name_formatter(row_content[1])
    wholesale_price = int(row_content[2][1:])
    retail_price = price_list_extractor (row_content[3])
    shopping_mall_price = price_list_extractor (row_content[4])
    quantity = row_content[5]
```

```
  return [name_of_foodtype,
      wholesale_price,
      retail_price,
      shopping_mall_price,
      quantity,
      food_type]


def exec_webscrapper() -> int:
        '''this function execute webscrapper function wheneven modified file
exceeds'''
        if (last_modified_csv("../csv/price_list.csv")):
                 # predefined lists
                veg_list =
["Beetroot","Cauliflower","Corn","Cucumber","Brinjal","Ginger","Lemon","Mango
Raw","Onion Big","Potato","Sweet Potato","Tomato"]
                fruits_list = ["Apple
Shimla","Banana","Cantaloupe","Guava","Orange","Papaya","Pineapple","Pomegran
ate Kabul","Sapota","Watermelon"]
                database = []
                save_address = "../csv/price_list.csv"


                #functions to webscrap to list

web_scrapper("https://vegetablemarketprice.com/market/chennai/today",
               veg_list,
               database,
               food_type = "vegetable")
```

```
web_scrapper("https://vegetablemarketprice.com/fruits/tamilNadu/today",
        fruits_list,
        database,
        food_type = "fruits")

        #Converting list to Pandas DataFrame and saving the file as csv
        df = pd.DataFrame(data = database)
        df.columns = ["NAME","WHOLESALE PRICE", "RETAIL
PRICE","SHOPPING MALL PRICE","QUANTITY","TYPE"]
        df.to_csv(save_address, header = True, index=False)
        return 1
    else:
        return 0
```

app2.py

```
import numpy as np
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask,render_template,request
from scripts.File_Extractor import index_to_information

from flask_sqlalchemy import SQLAlchemy
```

```
import sqlalchemy
from sqlalchemy import create_engine
from sqlalchemy import Table,Column,Integer,String,ForeignKey
from sqlalchemy import select
from sqlalchemy.orm import Session
from sqlalchemy.orm import declarative_base
from sqlalchemy.orm import relationship
import numpy as np
from keras.utils import load_img, img_to_array



current_dir=os.path.abspath(os.path.dirname(__file__))
app=Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI']='sqlite:///'+os.path.join(current_dir,'authenticate.sqlite3')
db=SQLAlchemy()
db.init_app(app)
app.app_context().push()

model=load_model("nutrition.h5")

class Users(db.Model):
    _tablename_='users'
    user_name=db.Column(db.String,primary_key=True,nullable=False)
    password=db.Column(db.String,nullable=False)
```

```
def print_info (info_dict : dict) -> str:
    info = []
    for i in info_dict.keys():
        info.append(f"{i} : {info_dict[i]}")
    return " ".join(info)


@app.route('/', methods=["GET","POST"])
def hello_world():
    if request.method=="GET":
        return render_template("login.html")
    elif request.method=="POST":
        username=request.form["user_name"]
        #print(username)
        delt=Users.query.filter(Users.user_name==username)
        try:
            print(delt[0])
        except:
            print('hi')
            return render_template('error.html')
        password=request.form["password"]
        user=Users.query.all()
        delt1=Users.query.filter(Users.password==password)
        try:
            print(delt1[0])
        except:
            print('hey')
            return render_template('error.html')
        #print(user)
        return render_template('index.html',user=user)
engine=create_engine("sqlite:///./authenticate.sqlite3")

@app.route('/register', methods=["GET","POST"])

def register():
    if request.method=="GET":
        return render_template("register.html")
    elif request.method=="POST":
#username=request.form["user_name"]
        #password=request.form["password"]
        with Session(engine,autoflush=False) as session:
            session.begin()

user=Users(user_name=request.form['user_name'],password=request.form['password'])
            session.add(user)
            session.flush()
            session.commit()
        return render_template("register.html")

@app.route('/index')
def index():
                return render_template("index.html")
```

```
@app.route('/predict',methods=['GET','POST'])
def upload():
        if request.method=='POST':
                f=request.files['image']
                basepath=os.path.dirname(__file__)
                filepath=os.path.join(basepath,'uploads',f.filename)
                f.save(filepath)
                img=image.load_img(filepath,target_size=(32,32))
                x = img_to_array(img)
                x = np.expand_dims (x,axis = 0)
                predictions = (model.predict(x) > 0.5).astype("int32")
                print(predictions)
                index = predictions[0].nonzero()[0][0]
                predicted_class_info =
index_to_information(index,[r"csv/index_list.csv",r"csv/nutrients_list.csv",r"csv/price
_list.csv"] )
                text="The Fruit is : " + predicted_class_info['NAME']
        return print_info(predicted_class_info )
if __name__=='__main__':
  app.run(debug=True)
```

## GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-618-1658310273

## Demo Link:

https://www.youtube.com/embed/YhP4sL2gDs0