RECIPE PREDICTION USING IBM WATSON

A UG PROJECT PHASE-1 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

Submitted by

MOHAMMAD TAJUDDIN	**19UK1A0588**
KONDAPALLI SUBHASH VARDHAN	**19UK1A05C8**
YAGGADI VAMSHI	**19UK1A05C2**
BURRA SHIVA KUMAR	**19UK1A0592**

Under the esteemed guidance of

Mr. P.Ilanna

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
(Affiliated to JNTUH, Hyderabad)
Bollikunta, Warangal – 506005

2019– 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
BOLLIKUNTA, WARANGAL – 506005 2019 – 2023

CERTIFICATE OF COMPLETION

UG PROJECT PHASE-1

This is to certify that the UG Project Phase-1 entitled **"RECIPE PREDICTION USING IBM WATSON"** is being submitted by ***MD.TAJUDDIN(*H.NO:19UK1A0588), *K.SUBASH VARDHAN* (H.NO:19UK1A05C8),*Y.VAMSHI*(H.NO:19UK1A05C2),*B.SHIVAKUMAR*(H.NO:19UK1A0592)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad** during the academic year **2022-2023**, is a record of work carried out by them under the guidance and supervision.

Project Guide
Mr. P. Ilanna
(Assistant Professor)

Head of the Department
Dr. R. Naveen Kumar
(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr.R.NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide, **Mr. P.ILANNA** , Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

MD. TAJUDDIN	**(19UK1A0588)**
K.SUBASH VARDHAN	**(19UK1A05C8)**
Y.VAMSHI	**(19UK1A05C2)**
B.SHIVA KUMAR	**(19UK1A0592)**

ABSTRACT

Food is an essential component of our individual and social life. Eating habits have a direct impact on our health and wellbeing, while ingredients, flavour and cooking recipes shape specific cuisines that are part of our personal and collective cultural identities. But there are also interesting applications of automatic food recognition to self-service restaurants and dining halls. For instance, accurate detection and segmentation of the different food items in a food tray can be used for monitoring food intake and nutritional information, and automatic billing to avoid the cashier bottleneck in self-service restaurants. This work deals with the problem of automated recognition of a photographed cooking dish and the subsequent output of the appropriate recipe. In this project, we focus on applications of automatic food recognition and identify the recipe in food by using convolutional neural networks. And this model will classify images into food categories and to output a matching recipe.

Keywords –recipe prediction,pizza,samosa,French fries,IBM watson

TABLE OF CONTENTS:-

PAGE NO

LIST OF FIGURES PAGE NO

8. APPENDIX

1. INTRODUCTION

1.1.MOTIVATION:

Food preferences around the World are changing specially in large cities due to factors like increasing income, migration and cultural mixing, appearance of new food fads and life-styles. These dietary changes often imply changes in the quantity and/or types of specific ingredients demanded.The demand for ingredients, their processing and supplying to consumers often happens in specific locations.Understanding their relevance in recipes used in specific locations can potentially inform design of interventions to improve, among others, nutrition and health, environmental and urban planning. Inorder to characterize ingredients used in specific locations, innovative tools are being used around theWorld to generate location-specific datasets.

1.2.PROBLEM DEFINITION:

Recipe prediction is mainly used in the restaurents big hotels to know the customer choice of the recipe and comfort to deliver the exact their choice of recipe and deliver the food item with exact recipe contain

PROJECT OBJECTIVE:

The main objective of this project is to analyze the food item like Pizza,Samosa and Frenchfries and to know their exact recipe by recipe prediction and gain broad understanding about image data and user can predict the food recipe by choosing the image in web application.

1.1.LIMITATIONS OF PROJECT:

Mixed dishes or multi-ingredient foods represents the majority of items in diets worldwide.These include not only foods prepared in the home but also foods prepared in the home but also foods prepared in restaurants,by food vendors,in institutions such as hospitals,schools,and the military,and by the food industry.To enable dietitians,nutritionists,and epidemiologists to evalute the role of these foods in the health of individuals,there is a need for composition data on these foods.Obtaining and using data on the content of multi-ingredient foods present a number of inherent difficulties,primarily because of the abundance and diversity of these kinds of foods.

1.1. ORGANIZATION OF DOCUMENTATION:

Actually there has been many theoretical projects and several experimental projects individually done based on machine learning and many different algorithms have been developed for forecasting the recipe prediction. But Machine Learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Our aim from the project

is to make use of NumPy or pandas libraries in Machine Learning and predict the food recipe choosen in the web application. It has own to visualize the data by using graphs. In the end, we are predicting the food recipe like samosa,pizza and French fries. The prediction is to be done using Machine Learning algorithms and withdrawing the conclusions.

2. PROBLEM STATEMENT

The food is more important to live man. And the food contains more recipes that shows the food is more interest to man how much he love the food when he eat.but there is some problem when taking the food it contains more demand in market business. It contains more multi-ingrient and it presents a number of inherent difficulties,primarily because of the abundance and diversity of these kinds foods.

There are some statements about recipe predictions:

Samosa: a samosa or singara is a fried or baked pastry with a savory filling,including ingredients suchbas spiced potatoes,onions,and peas. It may take different forms,including triangular,cone,or half-moon shapes,depending on the region.

Health issues:No,this recipe is not healthy.The samosa are deep fried which does not work for healthy living and it has a stuffing of potatoes.your fat levels increase as deep frying increase oil absorption.

Pizza: Is a dish of Italian origin consisting of a usually round,flat base of leavened wheat-based dough topped with tomatoes,cheese,and often various other ingredients, which is then baked at a high temperature , traditionally in a wood-fired oven.A small is some times called a pizzete.

Health issue: And it isn't health.Depending on the type of crust,the amount of cheese and the toppings used,pizza can rank any where from nutrionally decent to a diet disaster. Even healthy pizzas deliver a good amount of sodium from tomato sause and cheese,so if you are watching your salt intake ,you should eat with caution.

French fries: French fries are served hot, either soft or crispy, and are generally eaten as part of lunch or dinner or by themselves as a snack, and they commonly appear on the menus of diners, fast food restaurants, pubs, and bars. They are often salted and may be served with ketchup, vineger, mayonnaise, tomato sauce, or other local specialities. Fries can be topped more heavily, as in the dishes of poutine or chillicheese fries. French fries can be made from sweet potatoes instead of potatoes. A baked variant, oven fries, uses less or no oil.

Health issues: Given the fact that they're deep-fried in hydrogenated oils, fries come packed with a high amount of trans fat, which raises your bad cholesterol and lowers your good cholesterol. The profound effect of this is that you'll boost your heart disease risk.

3. LITERATURE SURVEY

3.1. INTRODUCTION:

Nowadays, with easy access to internet, food is delivered at our doorsteps just on the click of a button due to which people have started to consume higher amount of fast food. This has accelerated the chances of suffering from a chronic disease known as obesity. Since obesity has become such a widespread disease, various mobile e-health applications have been developed for assistive calorie measurement to help people fight against health-related problems.

- Stomach pain
- Food poison
- Over fat
- Obesity
- Diabeties
- Heart stroke
- Face problem
- Loss of nutrition
- Hair issues

3.2. EXISTING SYSTEM:

The purpose of the present model is to predict the whether a the food contain coorect recipe offer by the customer.The model examines data from food recipe concentrating on relationships between a key list of food items taste,recipe and variety using them to try and predict the likeliness of recipe. Here we are building a model by applying various machine learning algorithms find the best accurate model. And integrate to flask based web application. User can predict the disease by entering parameters in the web application.

3.3. DISADVANTAGES OF EXISTING SYSTEM:

- Collecting large amount of data set.
- Storing the large amount of image data sets.

3.4. PROPOSED SYSTEM:

Here we are building a model by applying various machine learning algorithms find the best accurate model. Then it will predict the exact food recipe to remove confusion in the data set. Some of the machines learning algorithms are:

1.Linear Regression:

Linear Regression is **a supervised machine learning algorithm where the predicted output is continuous and has a constant slope**. It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog).

Steps to implement Linear regression model:

1. Initialize the parameters.
2. Predict the value of a dependent variable by given an independent variable.
3. Calculate the error in prediction for all data points.
4. Calculate partial derivative w.r.t a0 and a1.
5. Calculate the cost for each number and add them.

2. Multiple Linear Regression:

Multiple Linear Regression is one of the important regression algorithms which models the linear relationship between a single dependent continuous variable and more than one independent variable. Multiple regression is a broader class of regressions that encompasses linear and nonlinear regressions with multiple explanatory variables.

3.Random Forest:

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

Working of Random Forest Algorithm

Step 1 − First, start with the selection of random samples from a given dataset.

Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result...

Step 3 − In this step, voting will be performed for every predicted result.

Step 4 − At last, select the most voted prediction result as the final prediction result.

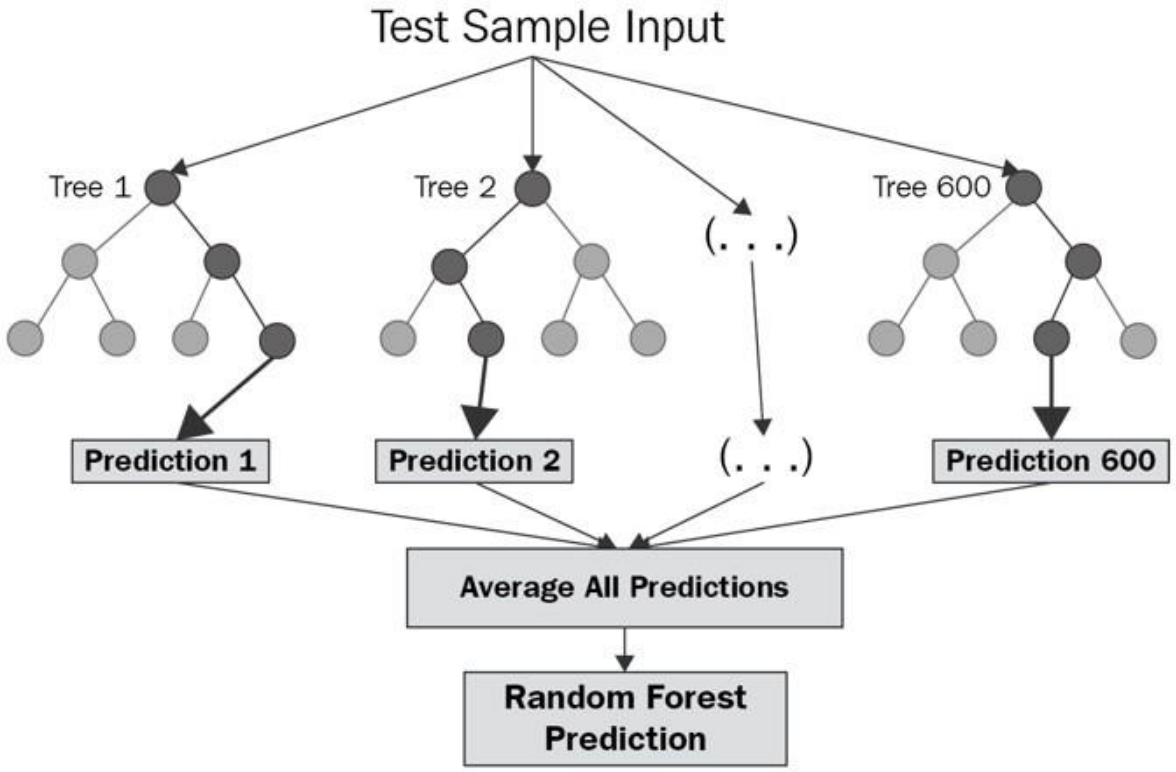

Figure 1 : Random Forest

4.<u>Logistic regression:</u>

- Logistic regression is **a supervised learning classification algorithm used to predict the probability of a target variable**. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.... Mathematically, a logistic regression model predicts P(Y=1) as a function of X.
- Logistic Regression is used when the dependent variable (target) is categorical. For example,
- To predict whether an email is a spam (1) or (0)
- Whether the tumor is malignant (1) or not (0)

You will need to train the datasets to run smoothly and see an incremental improvement in the prediction rate.

5. <u>k-nearest neighbor algorithm:</u>

- It is a supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problem statements. The number of nearest neighbor's to a new unknown variable that has to be predicted or classified is denoted by the symbol 'K'.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- K-Nearest Neighbors (KNN) is one of the simplest algorithms used in **Machine Learning for regression and classification problem**. KNN algorithms use data and classify new data points based on similarity measures (e.g. distance function). The data is assigned to the class which has the nearest neighbors.
- It's also worth noting that the KNN algorithm is also part of a family of ―lazy learning‖ models, meaning that it only stores a training dataset versus undergoing a training stage. This also means that all the computation occurs when a classification or prediction is being made. Since it heavily relies on memory to store all its training data, it is also referred to as an instance-based or memory-based learning method.

- The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of **K number of neighbors**

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

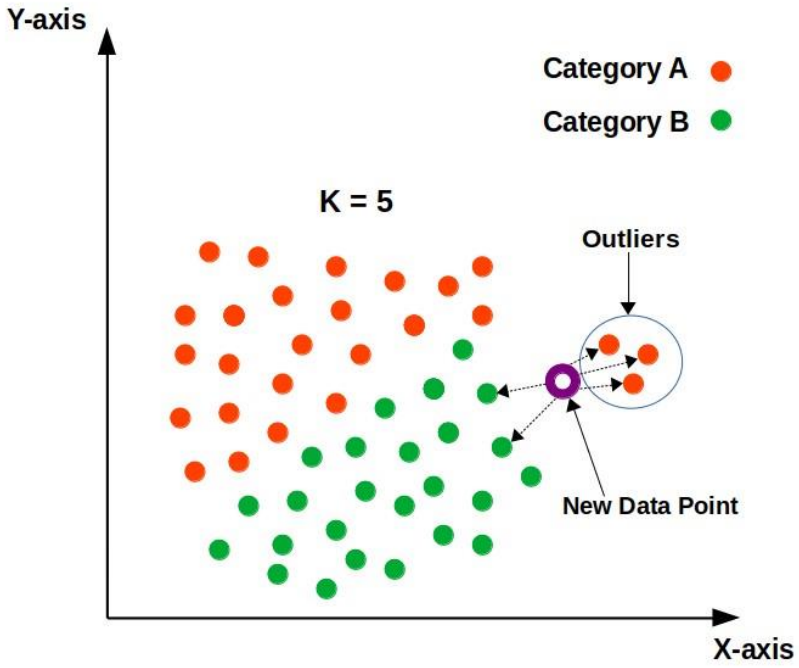

Figure 2 : k-nearest neighbor

4. EXPERIMENTAL ANALYSIS

Milestone 1: Data Collection

ML depends heavily on data, without data, a machine can't learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions.

You can collect datasets from different open sources like kaggle.com, data.gov; UCI machine learning repository etc. The dataset used for this project was obtained from Kaggle.

Milestone 2: Data Pre-processing

Data Pre-processing includes the following main tasks

Importing the libraries.

- Importing the dataset.
- Analyse the data.
- Taking care of Missing Data.
- Data Visualisation.
- Splitting Data into Train and Test.

Milestone 3: Model Building

The model building process involves setting up ways of collecting data, understanding and paying attention to what is important in the data to answer the questions you are asking, finding a statistical, mathematical or a simulation model to gain understanding and make predictions. Model Building Includes:

- Import the model building libraries.
- Initialising the model.
- Training the model.
- Model Evaluation.
- Save the Model.

Milestone 4: Application Building

- Create an HTML File.
- Build python code.
- Run the app in local browser.
- Show casting the prediction on UI.

4.1. PROJECT ARCHITECTURE:

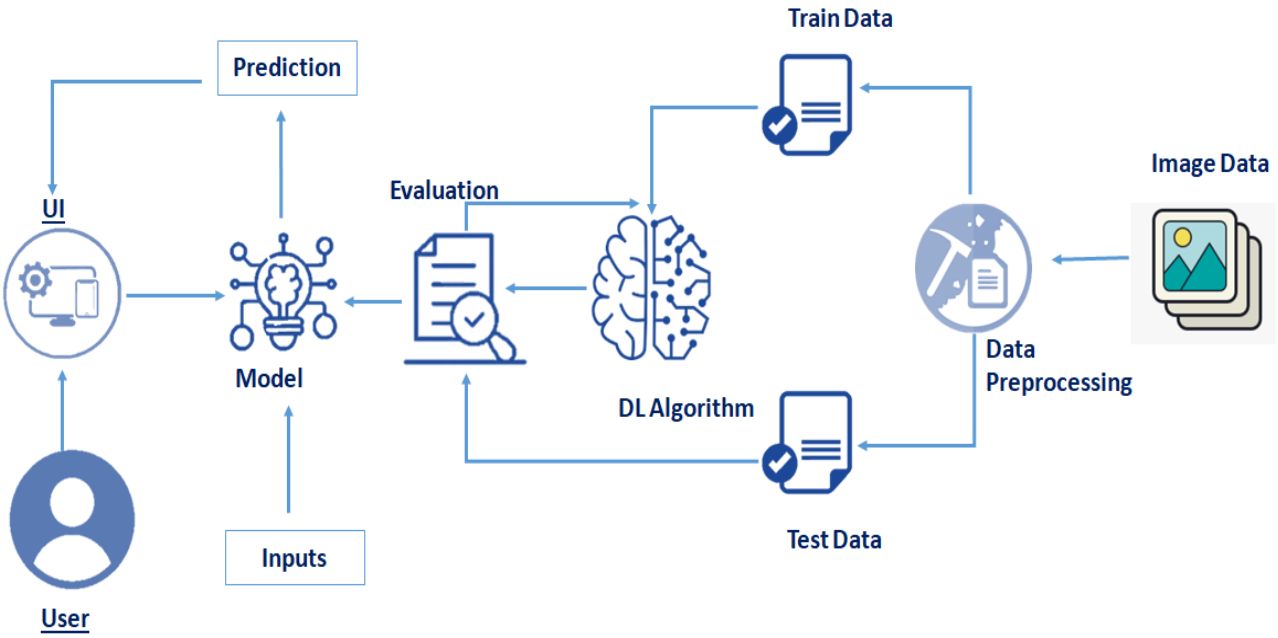

Figure 3 : Project Architecture

4.2. SOFTWARE AND HARDWARE REQUIREMENTS:

Software Requirements:

- Anaconda Environment
- Flask
- Python 3.9
- And other python libraries like NumPy, pandas.

Hardware Requirements:

- Operating system
- Processing
- RAM
- Operating system specifications

- Disk space

4.3. BLOCK DIAGRAM:

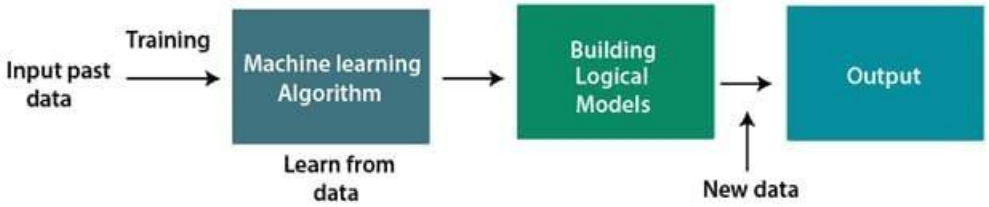

Figure 4 : Block Diagram

4.4. PROJECTFLOW:

- User interacts with the UI (User Interface) to upload the input features.
- Uploaded features/input is analysed by the model which is integrated.
- Once a model analyses the uploaded inputs, the prediction is showcased on the UI.

5. DESIGN

5.1. USE CASEDIAGRAM:

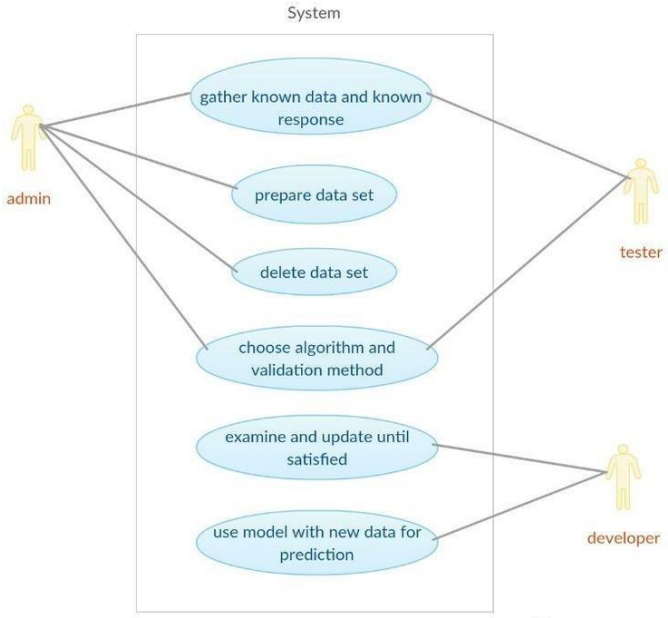

Figure 5: Use case Diagram

5.2. FLOWCHART:

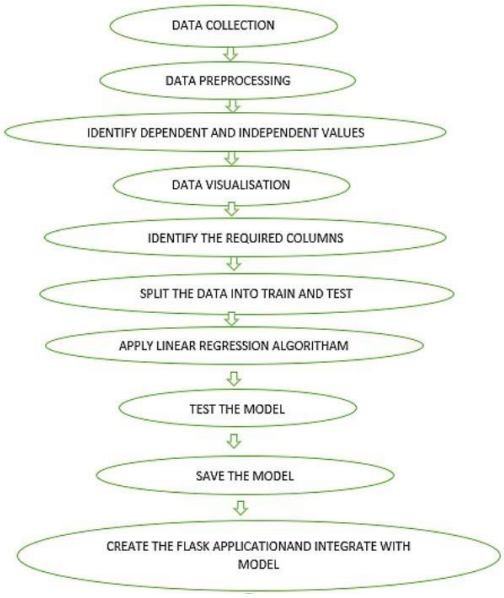

Figure 6: Flowchart

5.3. DECISION TREE:

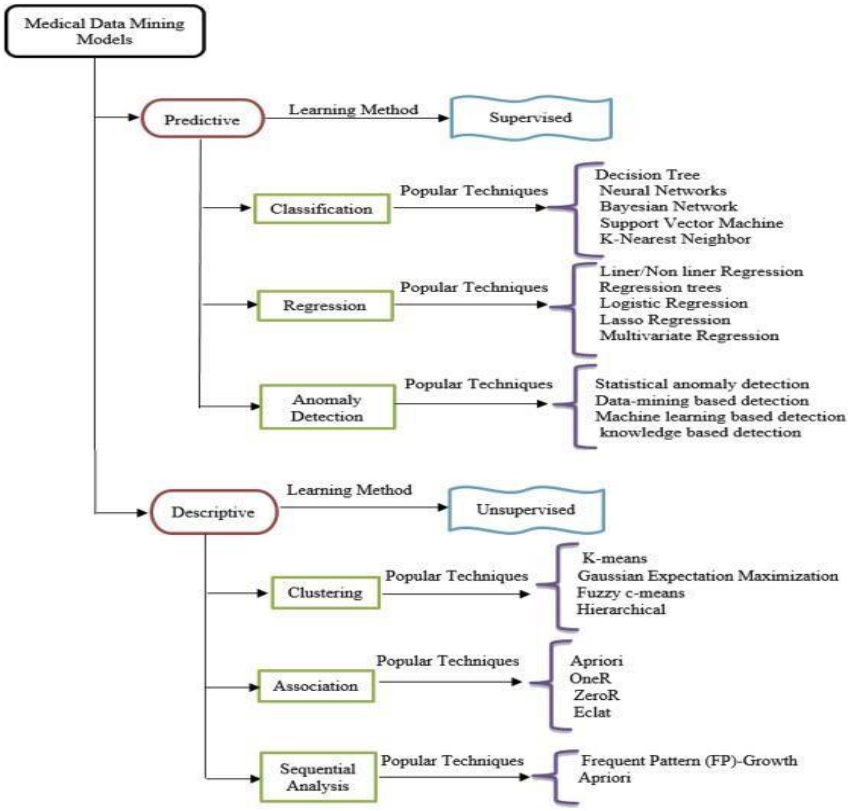

Figure 7: Decision Tree

5.3. SEQUENCE DIAGRAM:

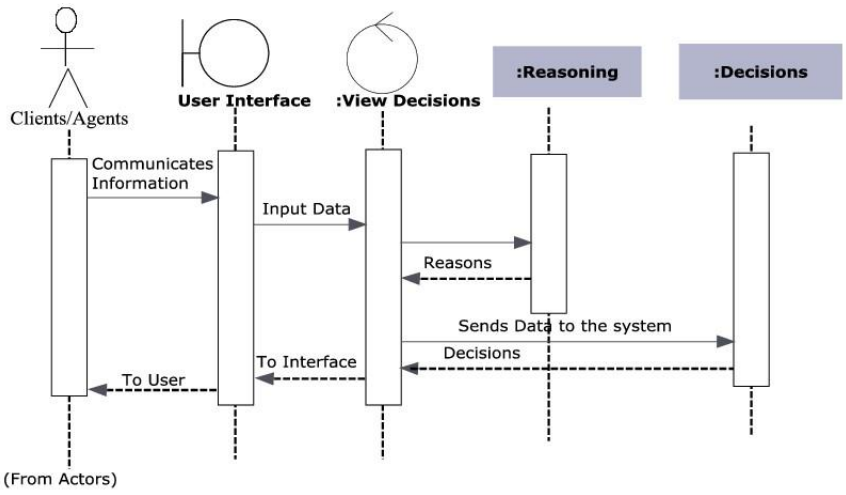

Figure 8: Sequence Diagram

5.4. COMMUNICATIO DIAGRAM:

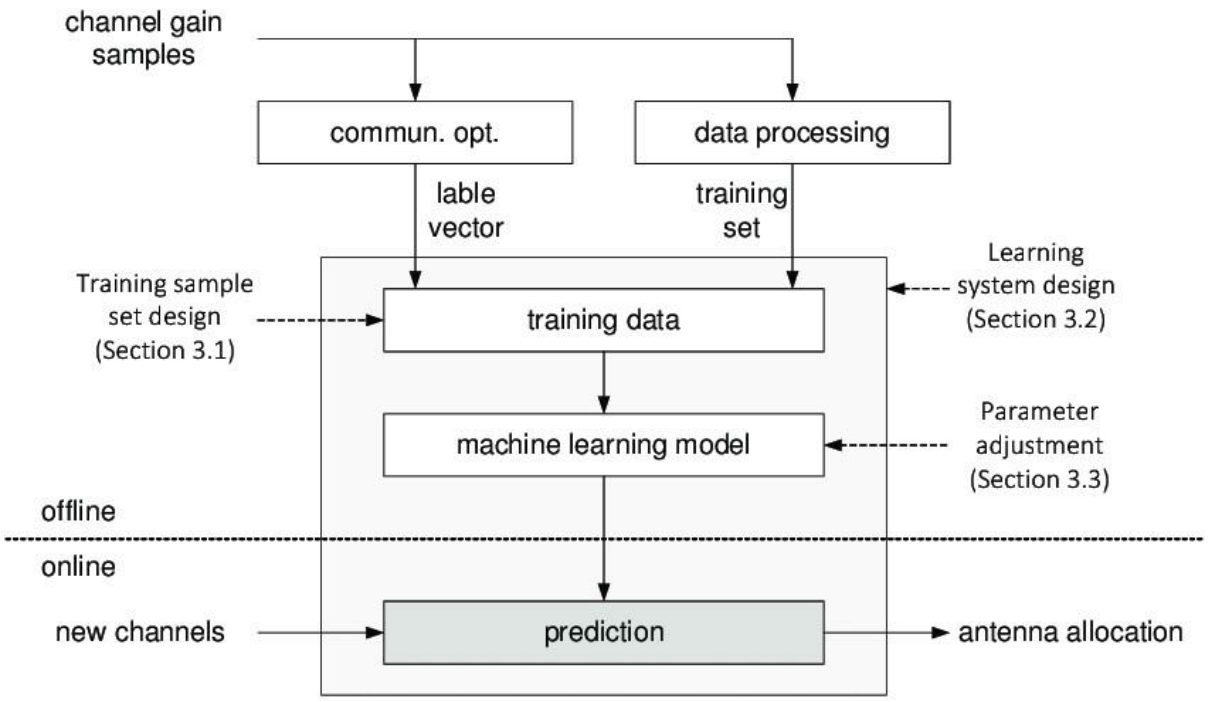

Figure 9: Communication Diagram

6. CONCLUSION

In UG Project Phase-1, we have worked on problem statement, literature survey and also done the experimental analyses which are required for the project to move forward. In experimental analysis we have discussed about the machine learning concepts and models and explained the algorithms to be used in the project. We also discussed about the flowcharts, use case diagrams, decision tree and sequence diagrams which are used in the project. Based on the experimental analysis we have designed the model for the project.

Entire designing part is involved in UG Project Phase-1.

7.FUTURE SCOPE

UG Project Phase-1 about recipe recognisation and All the implementation is done and conclusions will be retrieved in this phase. We will work on the applications, advantages, and disadvantages of the project in this phase.

8. APPENDIX

A.SOURCE CODE:

1.Image data augmentation

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset.

The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class.
Let us import the ImageDataGenerator class from Keras.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np#used for numerical analysis
```

```
#setting parameter for Image Data agumentation to the traing data
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
```

Arguments:

- directory: Directory where the data is located. If labels are "inferred", it should contain subdirectories, each containing images for a class. Otherwise, the directory structure is ignored.
- batch_size: Size of the batches of data. Default: 32.
- target_size: Size to resize images to after they are read from disk.
- class_mode:

- ‘int': means that the labels are encoded as integers (e.g. for sparse_categorical_crossentropy loss).
- 'categorical' means that the labels are encoded as a categorical vector (e.g. for categorical_crossentropy loss).
- 'binary' means that the labels (there can be only 2) are encoded as float32 scalars with values 0 or 1 (e.g. for binary_crossentropy).
- None (no labels).

```
#performing data agumentation to train data
x_train=train_datagen.flow_from_directory(directory=r'E:\FOOD Classification\Food-Classification-from-I
                                          ,target_size=(64,64),batch_size=32,class_mode='categorical')
```

```
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
#performing data agumentation to test data
x_test=test_datagen.flow_from_directory(directory=r'E:\FOOD Classification\Food-Classification-from-Ima
                                        ,target_size=(64,64),batch_size=32,class_mode='categorical')
```

2.Model Building:

The neural network model is to be built by adding different network layers like convolution, pooling, flattening, dropout, and neural layers.

In this milestone, we start building our model by:

1.Initializing the mode

2.Adding Convolution layers

3. Adding Pooling layers

4.Flatten layer

5.Full connection layers which include hidden layers

At last, we compile the model with layers we added to complete the neural network structure

```
import numpy as np#used for numerical analysis
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense,Flatten
#Faltten-used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D,MaxPooling2D #Convolutional layer
#MaxPooling2D-for downsampling the image
```

```
# adding model layer
model.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))#convolutional layer
model.add(MaxPooling2D(pool_size=(2,2))) #MaxPooling2D-for downsampling the input

model.add(Conv2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))#droping input randomly for preventing from overfiting


model.add(Flatten())#flatten the dimension of the image
model.add(Dense(32))#deeply connected neural network layers.
```

```
model.summary()#summary of our model
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 29, 29, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 6272)              0
_________________________________________________________________
dense (Dense)                (None, 32)                200736
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 99
=================================================================
Total params: 210,979
Trainable params: 210,979
Non-trainable params: 0
_________________________________________________________________
```

```
# Compile model
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

```
# Save the model
model.save('food.h5')
```

```
from tensorflow.keras.models import load_model
from keras.preprocessing import image
model = load_model("food.h5") #loading the model for testing
```

```
img = image.load_img(r"E:\FOOD Classification\Food-Classification-from-Images-Using-Convolutional-Neural-
x = image.img_to_array(img)#image to array
x = np.expand_dims(x,axis = 0)#changing the shape
pred = model.predict_classes(x)#predicting the classes
pred
```

```
index=['french_fries', 'pizza', 'samosa']
result=str(index[pred[0]])
result
```

```
'samosa'
```

3.FLASK:

After the model is built, we will be integrating it into a web application so that users can interact with the model.

```
import os
import numpy as np #used for numerical analysis
from flask import Flask,request,render_template#Flask-It is our framework which
#we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
#render_template- used for rendering the html pages
from tensorflow.keras.models import load_model#to load our trained model
from tensorflow.keras.preprocessing import image
```

```
app=Flask(__name__)#our flask app
model=load_model('food.h5')#loading the model

@app.route("/") #default route
def upload_file():
    return render_template("RR.html")#rendering html page

@app.route("/about") #route about page
def upoad_file1():
    return render_template("RR.html")#rendering html page


@app.route("/upload") # route for info page
def upload_file2():
    return render_template("RRP.html")#rendering html page
```

```python
@app.route("/predict",methods=["GET","POST"]) #route for our prediction
def upload():
    if request.method=='POST':
        f=request.files['file'] #requesting the file
        basepath=os.path.dirname('__file__')#storing the file directory
        filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in uploads folder
        f.save(filepath)#saving the file

        img=image.load_img(filepath,target_size=(64,64)) #load and reshaping the image
        x=image.img_to_array(img)#converting image to array
        x=np.expand_dims(x,axis=0)#changing the dimensions of the image

        pred = model.predict_classes(x)  # predicting classes
        print(pred)  # printing the prediction
        index = ['French Fries', 'Pizza', 'Samosa']
        result = str(index[pred[0]])
        if (result=="French Fries"):
            return render_template("0.html",showcase =  str(result))
        elif (result=="Pizza"):
            return render_template("1.html",showcase =  str(result))
        else:
            return render_template("2.html",showcase =  str(result))
        #return result#resturing the result
    else:
        return None

#port = int(os.getenv("PORT"))
if __name__=="__main__":
    app.run(debug=False)#running our app
    #app.run(host='0.0.0.0', port=8000,debug=False)
```

```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```