PROJECT REPORT

TEAM ID	PNT2022TMID27576
PROJECT NAME	DEMANDEST - AI POWERED FOOD DEMAND FORECASTER

1. INTRODUCTION

1.1. PROJECT OVERVIEW:

Demand forecasting is one of the important inputs for a successful restaurant yield revenue management system. Sales forecasting is crucial for an independent restaurant and for restaurant chains as well. The sales transaction data collected by restaurant chains may be analyzed at both the store level and the corporate level. At the level of single store, exploring the large amounts of transaction data allows each restaurant to improve its operations management (e.g., labor scheduling) and product management (e.g., inventory replenishment, product preparation scheduling), and in consequence reducing restaurant operating costs and increasing quality of serving food.

1.2. PURPOSE:

The main aim of this project is to create an appropriate machine learning model to forecast the number of orders to gather raw materials for the next ten weeks. To achieve this, we should know the information about fulfillment centers like area, city etc., and meal information like category of food, sub category of food, price of the food or discount in a particular week. By using this data, we can use any classification algorithm to forecast the quantity for 10 weeks. A web application is built which is integrated with the model built.

2. LITERATURE SURVEY

2.1. EXISTING PROBLEM:

A food delivery service has to deal with a lot of perishable raw materials which makes it all, the most important factor for such a company is to accurately forecast daily and weekly demand. Too much inventory in the warehouse means more risk of wastage, and not enough could lead to out-of-stocks - and push customers to seek solutions from your competitors. The replenishment of the majority of raw materials is done on a weekly basis and since the raw material is perishable, the procurement planning is of utmost importance.

2.2. REFERENCES

1. **PAPER TITLE:** Food Quality Demand and Monitoring System.

AUTHOR: Atkare Prajwal, Patil Vaishali, zade payal, Dhapudkar Sumit

Food plays a very important role in our day to day life. With an increase in globalization, the quality of food decreases day by day. Most of the time various food processing is done to keep the food fresh. Various preservatives or the ingredients are added in the food so that it looks fresh or tempting. Now most of the food is preserved with the chemicals which causes the food contamination. This contamination leads to various diseases which results in the consumer wanting healthy food.

2. **PAPER TITLE:** Flexible Demand Forecasting in Intelligent Food Supply Chain Management.

AUTHOR: Srimathi Ravisankar,Kanimozhi Mahendran, Srilakshmi Arulmurugan M.R. Sumalatha

In the Food industry, Big data analytics concepts and techniques are being used in the food business for inventory optimization, which combines historical data with predictive techniques to improve supply chain management techniques. Demand forecasting, food tracing, and information exchange for suppliers, warehouses, and restaurants to connect with one another are the three modules covered in this paper that deal with managing the food supply chain.

3. **PAPER TITLE:** Daily Food Demand Forecast with Artificial Neural Networks: Kırıkkale University Case.

AUTHOR:

1. Zenep Centinkaya Department of Computer Engineering Kirıkkale University Kırıkkale, Turkey.
2. Erdal Department of Computer Engineering Kırıkkale University Kırıkkale, Turkey.

In food service organizations, demand estimation is very important in planning of production. When an accurate demand forecast is made, the resources are used more efficiently, and the production wants to be lost in some places. In the institutions where the number of people to make a request is not known clearly, the demand forecasts of the quantitative and qualitative targets are made.

4. **PAPER TITLE:** Demand forecasting in restaurants using machine learning and statistical analysis.

AUTHOR: Takashi Tanizaki,Tomohiro Hoshino ,Takeshi Shimmura,Takeshi Takenaka

In the paper, demand forecasting in restaurants using machine learning is proposed. Many researches have been proposed on demand forecasting innovation utilizing POS information. However,in order to make demand forecasts at a genuine store, it is important to lay out a store-explicit demand forecasting model in light of different factors, for example, the store area, the climate, occasions and so on .Thus, we developed an demand forecasting model that practically consolidates the previously mentioned information utilizing machine learning.

5. **PAPER TITLE:** Food Demand Prediction Using Machine Learning

AUTHOR:

1. K Aishwarya, Dept. of Computer Science & Engineering, NIE College, Mysore, Karnataka,
2. N Rao, Dept. of Computer Science & Engineering, NIE College, Mysore, Karnataka, India
3. Akshit Mishra, Dept. of Computer Science & Engineering, NIE College, Mysore, Karnataka.
4. Nikita Kumari, Dept. of Computer Science & Engineering, NIE College, Mysore, Karnataka, India

Demand forecasting is the process in which historical data is used to estimate the quantity of product a customer will purchase. This prediction activity is used in many fields like retailing, food industry etc. In Restaurants, prediction plays a vital role as most of the basic ingredients have short-shelf life. The demands depend upon many explicit and hidden contexts such as season, region etc. In this paper, the number of orders is used to forecast stock of items, using machine learning with internal and external data. In this we provide an appropriate algorithm for demand forecasting which is capable of overpowering the wastage of short life items. Proposed algorithms like Bayesian Linear Regression, LASSO, XGBoost algorithm are used that considerably improves the forecasting performance.

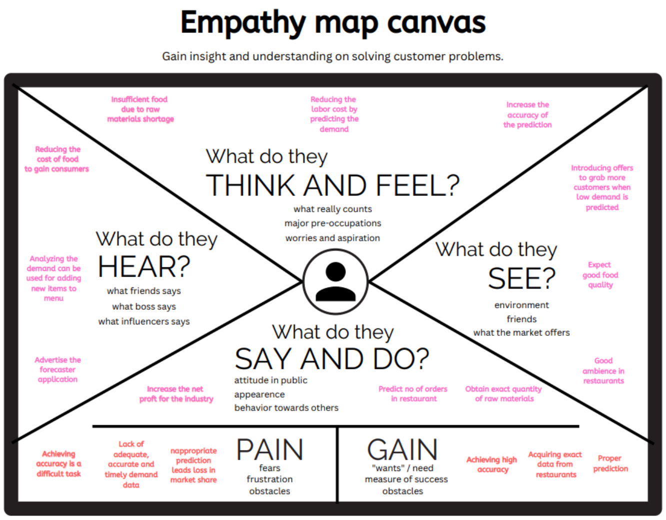
2.3. PROBLEM STATEMENT DEFINITION:

The success of a restaurant not only depends on taste, ambience but also on service. The most important part among the services is serving fresh food. In order to provide this, the restaurants need to prepare food daily, this requires buying some fresh self life food products every day. The major task that one would face in this will be predicting the quantity of products to be bought and prepared. It is very difficult to predict the number of orders in a given restaurant on a given day. A wrong prediction may end up purchasing and preparing less amount of food which will cause shortage or purchasing and preparing more which will lead to wastage of food. So, predicting the exact demand is a challenge because of uncertainty and fluctuations in consumer demand . These variations and fluctuations in demand may be because of price change, promotions, change in customer's preferences and weather changes. All these factors imply that some dishes are sold mostly during a limited period of time. Although we know that some regular seasonal pattern is expected, the features that predict these seasons are not directly observed. Thus, drops and rises in orders because of these seasonal changes are difficult to predict. In order to solve such problems, we are researching how to predict the demand . Here we are researching food demand forecasting methods using internal data such as number of orders.

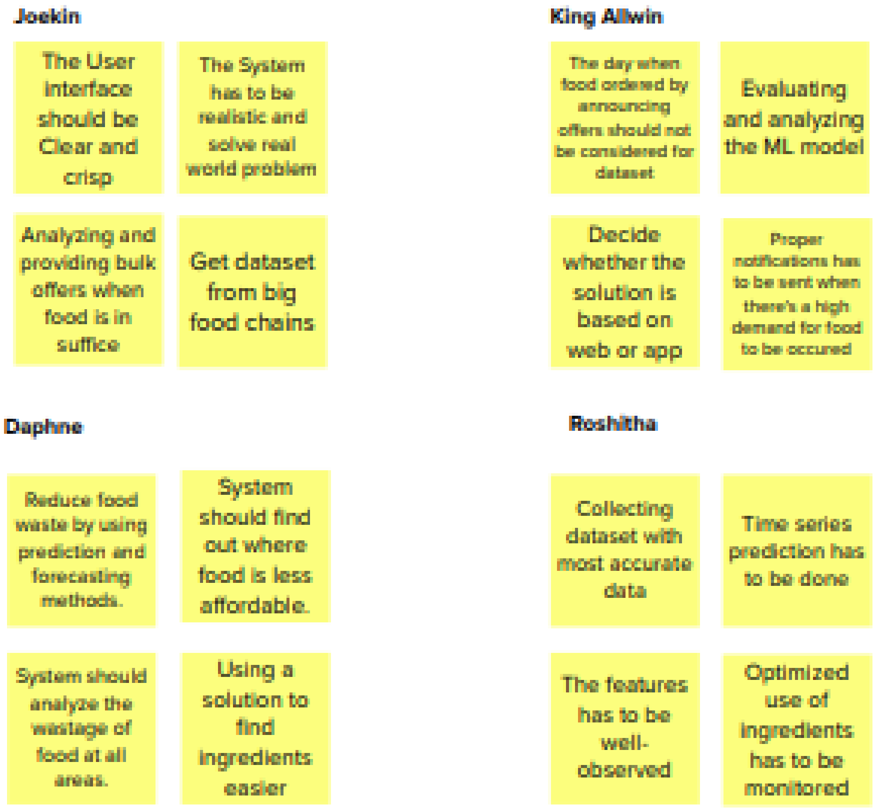
Problem Statement (PS)	**I am**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	A food vendor	Purchase raw ingredients	I purchased	I don't know how many food orders will be placed	Worried about wasting live ingredients
PS-2	A customer	Buy dish of my choice	The dish was sold out	The vendor provided less amount of food	Hungry and bad impression about the restaurant
PS-3	An inspector	Inspect food quality of the restaurant	I found that the food was not fresh	The ingredients used was not fresh	angry

3. IDEATHON AND PROPOSED SOLUTION

3.1. EMPATHY MAP CANVAS:


Empathy map canvas
Gain insight and understanding on solving customer problems.
Insufficient food due to raw materials shortage
Reducing the labor cost by predicting the demand
Increase the accuracy of the prediction
Reducing the cost of food to gain consumers
Introducing offers to grab more customers when low demand is predicted
What do they THINK AND FEEL?
what really counts
major pre-occupations
worries and aspiration
What do they HEAR?
what friends says
what boss says
what influencers says
Analyzing the demand can be used for adding new items to menu
What do they SEE?
environment
friends
what the market offers
Expect good food quality
What do they SAY AND DO?
attitude in public
appearence
behavior towards others
Advertise the forecaster application
Increase the net proft for the industry
Predict no of orders in restaurant
Obtain exact quantity of raw materials
Good ambience in restaurants
PAIN
fears
frustration
obstacles
Achieving accuracy is a difficult task
Lack of adequate, accurate and timely demand data
nappropriate prediction leads loss in market share
GAIN
"wants" / need
measure of success
obstacles
Achieving high accuracy
Acquiring exact data from restaurants
Proper prediction

3.2. IDEATHON AND BRAINSTORMING:



Joekin

The User interface should be Clear and crisp

The System has to be realistic and solve real world problem

Analyzing and providing bulk offers when food is in suffice

Get dataset from big food chains

King Allwin

The day when food ordered by announcing offers should not be considered for dataset

Evaluating and analyzing the ML model

Decide whether the solution is based on web or app

Proper notifications has to be sent when there's a high demand for food to be occured

Daphne

Reduce food waste by using prediction and forecasting methods.

System should find out where food is less affordable.

System should analyze the wastage of food at all areas.

Using a solution to find ingredients easier

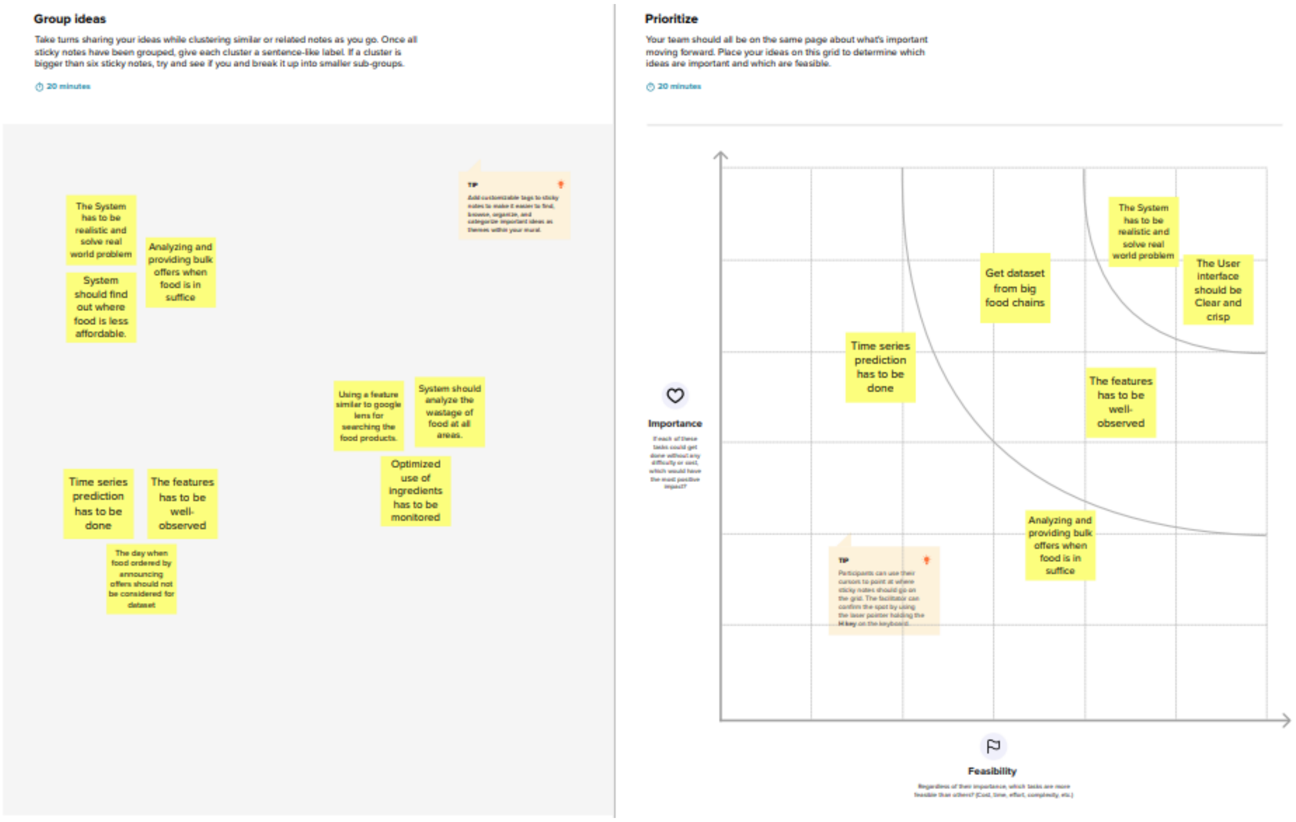
Roshitha

Collecting dataset with most accurate data

Time series prediction has to be done

The features has to be well-observed

Optimized use of ingredients has to be monitored



Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

The System has to be realistic and solve real world problem
Analyzing and providing bulk offers when food is in suffice
System should find out where food is less affordable.
Using a feature similar to google lens for searching the food products.
System should analyze the wastage of food at all areas.
Optimized use of ingredients has to be monitored
Time series prediction has to be done
The features has to be well-observed
The day when food ordered by announcing offers should not be considered for dataset

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

The System has to be realistic and solve real world problem
Get dataset from big food chains
The User interface should be Clear and crisp
Time series prediction has to be done
The features has to be well-observed
Analyzing and providing bulk offers when food is in suffice
Importance
Feasibility

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.

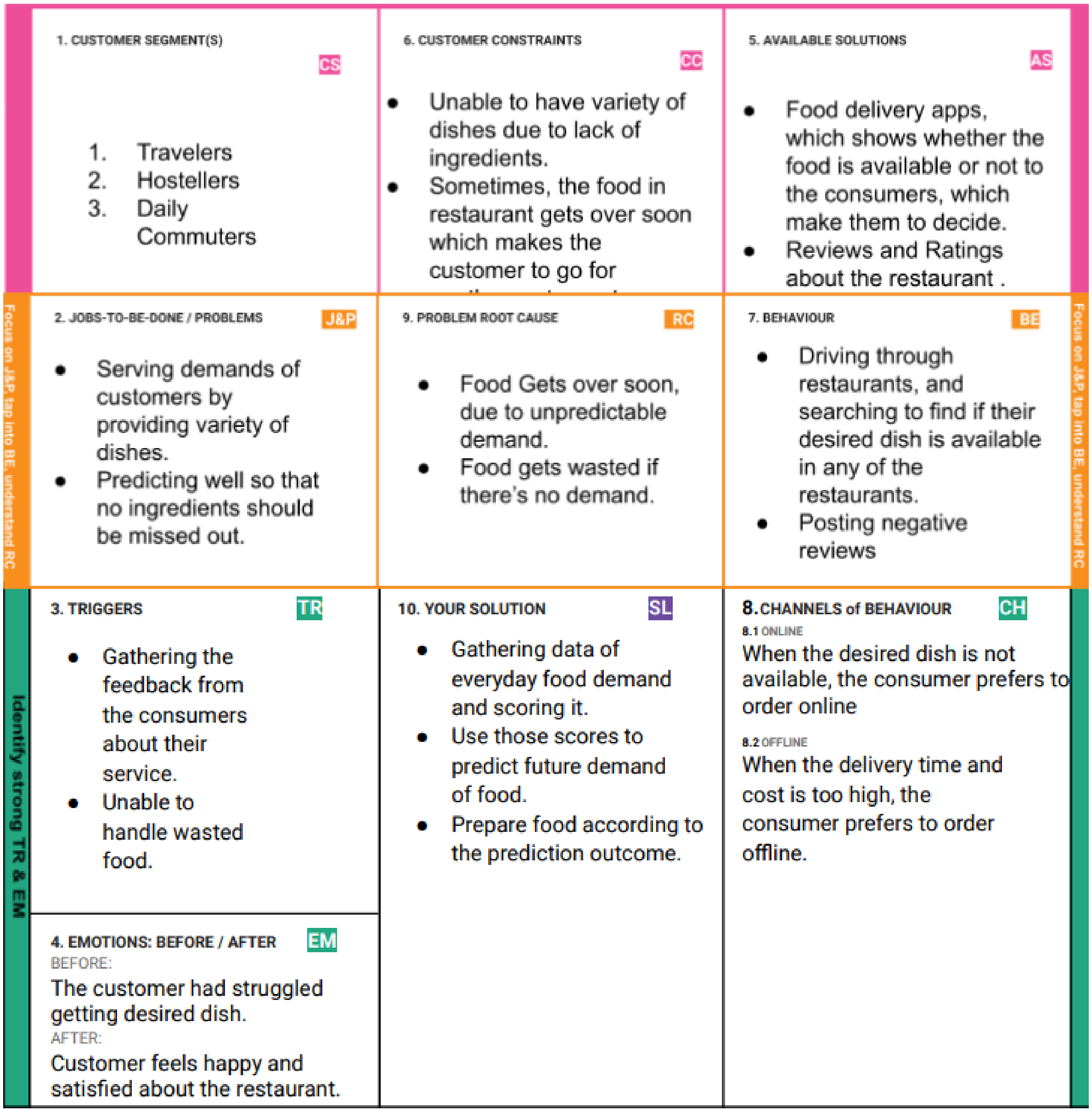
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3. PROPOSED SOLUTION:

S.NO	Parameter	Solution
1.	Problem Statement (Problem to be solved)	A food delivery service has to deal with a lot of perishable raw materials which makes it all, the most important factor for such a company is to accurately forecast daily and weekly demand. Too much inventory in the warehouse means more risk of wastage, and not enough could lead to out-of-stocks - and push customers to seek solutions from your competitors. The replenishment of the majority of raw materials is done on a weekly basis and since the raw material is perishable, the procurement planning is of utmost importance, the task is to predict the demand for the next 10 weeks.
2.	Idea / Solution description	The main aim of this project is to create an appropriate machine learning model to forecast the number of orders to gather raw materials for the next ten weeks. To achieve this, we should know the information about fulfillment centers like area, city etc., and meal information like category of food, sub category of food, price of the food or discount in a particular week. By using this data, we can use any classification algorithm to forecast the quantity for 10 weeks. A web application is built which is integrated with the model built.

3.	Novelty / Uniqueness	The system keeps updating users information based on their interest and helps the management to prepare the on demand food to reduce unnecessary preparation of other foods.
4.	Social Impact / Customer Satisfaction	Food wastage will be reduced Increase in profit for the management Decrease the use of raw materials
5.	Business Model (Revenue Model)	It is highly profitable. After examining the food-related data for each location, it will determine which location was most in demand.
6.	Scalability of the Solution	It gives day to day prediction of the foods being sold which enables scalability by reducing the wastage of food.

3.4. PROBLEM SOLUTION FIT:


Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Focus on J&P, tap into BE, understand RC

1. CUSTOMER SEGMENT(S) CS

1. Travelers
2. Hostellers
3. Daily Commuters

6. CUSTOMER CONSTRAINTS CC

- Unable to have variety of dishes due to lack of ingredients.
- Sometimes, the food in restaurant gets over soon which makes the customer to go for

5. AVAILABLE SOLUTIONS AS

- Food delivery apps, which shows whether the food is available or not to the consumers, which make them to decide.
- Reviews and Ratings about the restaurant .

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- Serving demands of customers by providing variety of dishes.
- Predicting well so that no ingredients should be missed out.

9. PROBLEM ROOT CAUSE RC

- Food Gets over soon, due to unpredictable demand.
- Food gets wasted if there's no demand.

7. BEHAVIOUR BE

- Driving through restaurants, and searching to find if their desired dish is available in any of the restaurants.
- Posting negative reviews

3. TRIGGERS TR

- Gathering the feedback from the consumers about their service.
- Unable to handle wasted food.

4. EMOTIONS: BEFORE / AFTER EM

BEFORE:
The customer had struggled getting desired dish.
AFTER:
Customer feels happy and satisfied about the restaurant.

10. YOUR SOLUTION SL

- Gathering data of everyday food demand and scoring it.
- Use those scores to predict future demand of food.
- Prepare food according to the prediction outcome.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
When the desired dish is not available, the consumer prefers to order online

8.2 OFFLINE
When the delivery time and cost is too high, the consumer prefers to order offline.

4. REQUIREMENTS ANALYSIS

4.1. FUNCTIONAL REQUIREMENTS:

The following is the functional requirements of the proposed solution.

FR NO.	Functional Requirement (Epic)	Sub Requirement (Story/Sub-Task)
FR-1	User Sign Up	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Dashboard	The time series analytics are shown in dashboard
FR-4	Predictions	The predictions graph are shown
FR-5	Feedback	The feedback is provided for the predictions
FR-6	Password Reset	The user should be able to reset the password / recover their account

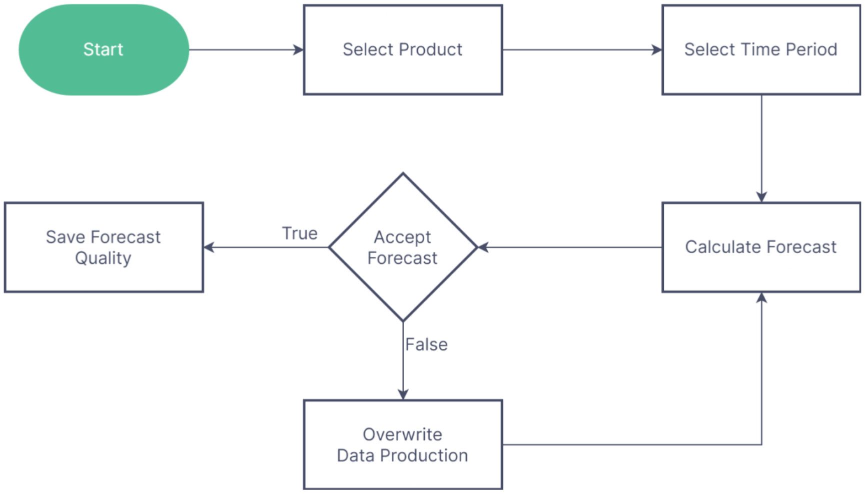
4.2. NON-FUNCTIONAL REQUIREMENTS:

The following are the non-functional requirements of the proposed solution.

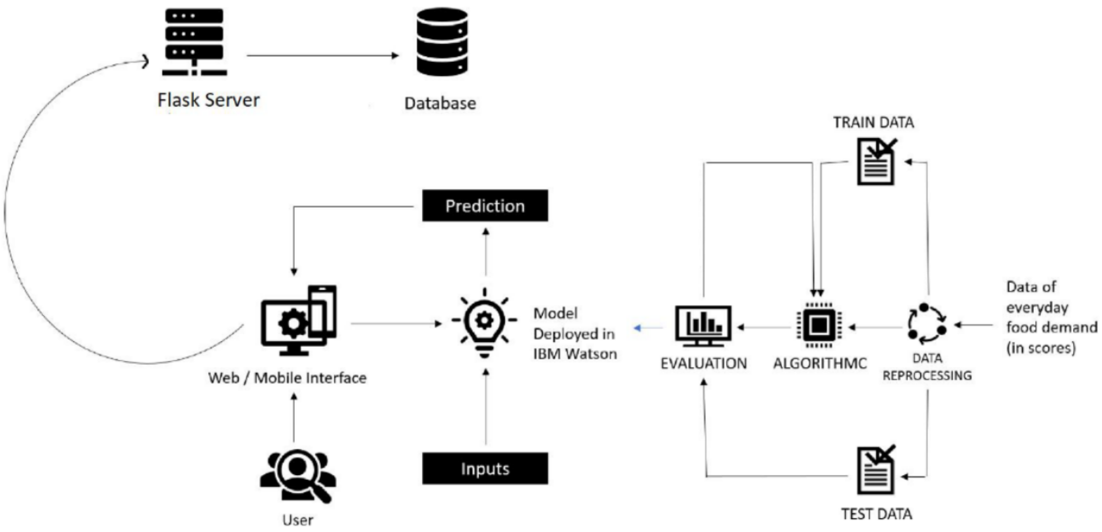
NFR NO.	Non-Functional Requirements	Description
NFR-1	Usability	The application should be usable in such a way that it provides accurate predictions.
NFR-2	Security	The application has to provide secured sessions for managing data.
NFR-3	Reliability	The predictions provided should be trustable by the users.
NFR-4	Performance	The application should exhibit good performance.
NFR-5	Availability	The application should have no downtime.
NFR-6	Scalability	The application should handle a huge amount of growing data.

5. PROJECT DESIGN

5.1. DATA FLOW DIAGRAMS:



5.2. SOLUTION AND TECHNICAL ARCHITECTURE:



5.3. USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User (Web User)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and	I can access my account / dashboard	High	Sprint-1

			confirming my password.			
		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
	Dashboard	USN-3	As a user, I can access all the services provided in the dashboard.	I can predict the orders for the upcoming day and I estimate the raw materials for the same.	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Administrator	Management	USN-4	As an admin I can maintain the application	I can perform maintenance of the app even after the release.	Medium	Sprint-1
		USN-5	As an admin I can update the new datasets to the model and train them.	I can periodically update the datasets	High	Sprint-1
		USN-6	As an admin I can update the features of the app and upgrade it to better versions	I can perform upgrades of features and versions.	Medium	Sprint-1
		USN-7	As an admin I can maintain all the user details stored and the user’s history	I can maintain the application user’s records.	High	Sprint-1

6. PROJECT PLANNING AND SCHEDULING

6.1. SPRINT PLANNING AND ESTIMATION:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.2. SPRINT DELIVERY SCHEDULE:

Sprint	Functional Requirements (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Pre – Requisites	USN-1	A prerequisite is a required prior condition. If something is required in advance of something else, like if you have to take a beginning Spanish class before signing up for Spanish, then it's a prerequisite.	10	Low	Joekin Roche Roshitha Daphne Sharon King Allwin
Sprint-1	Dataset collection	USN-2	A tool in Agile software development used to capture a description of a software feature from a user's perspective.	10	Medium	Joekin Roche King Allwin
Sprint-2	Data Pre-Processing. Importing the libraries	USN-3	In this post I am going to walk through the implementation of Data Pre-processing methods using Python.	5	High	Joekin Roche King Allwin
Sprint-2	Reading the dataset. Exploratory data analysis	USN-4	Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot	5	High	Roshitha Daphne Sharon

			anomalies, to test hypotheses and to check assumptions with the help of summary.			

Sprint-2	Checking for null values. Reading and merging.csv files.	USN-5	A null indicates that a variable doesn’t point to any object and holds no value.
			Step 1: Create & Export Multiple Data Frames. First, we’ll use the following code to create and export three data frames to CSV files:
			#create three data frames
			df1 <- data. …
			Step 2: Import & Merge Multiple CSV Files.
				2	Medium	Joekin Roche
						King Allwin

Sprint-2	Dropping columns. Label encoding	USN-6	First, you define the table name from which you wish to remove or delete the column. Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form.	6	Medium	Roshitha Daphne Sharon
Sprint-2	Dropping columns. Label encoding	USN-6	First, you define the table name from which you wish to remove or delete the column. Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form.	6	Medium	Joekin Roche King Allwin
Sprint-2	Splitting the dataset into dependent and independent variables. Split the dataset into train set and test set	USN-7	The simplest way to split the modeling dataset into training and testing sets is to assign 2/3 data points to the former and the remaining one-third to the latter.	2	Low	Roshitha Daphne Sharon
Sprint-3	Model Building	USN-8	What the person using the product wants to Be able to do. A traditional requirement focuses on functionality	10	High	Joekin Roche Roshitha Daphne Sharon King Allwin
Sprint-3	Train and test	USN-9	The train-test split	5	Low	Joekin Roche

	model algorithms Model evaluation		procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data			Roshitha
Sprint-3	Save the model. Predicting the output using the model.	USN-10	The predict passes the input vector through the model and returns the output tensor for each datapoint.	5	Medium	Daphne Sharon King Allwin
Sprint-4	Application building. Create an HTML file	USN-11	An app builder is an online software tool that allows everyone to create and publish apps for mobile devices without code development.	10	High	Joekin Roche Roshitha Daphne Sharon King Allwin
Sprint-4	Build python code. Run the app	USN-12	A tool provided by the Python Packaging Authority (PyPA) for building Python packages.	10	High	Joekin Roche Roshitha Daphne Sharon King Allwin

7. CODING AND SOLUTIONING

7.1. OBTAINING INPUT VALUES:

base.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"/>
    <meta name="description" content="" />
    <meta name="author" content="" />
    <title>DemandEst - Food Demand Forecaster</title>
    <link rel="icon" type="image/x-icon" href="assets/favicon.ico" />
    <!-- Font Awesome icons (free version)-->
    <script
      src="https://use.fontawesome.com/releases/v6.1.0/js/all.js"
      crossorigin="anonymous"
    ></script>
    <!-- Google fonts-->
    <link rel="preconnect" href="https://fonts.gstatic.com" />
    <link
    href="https://fonts.googleapis.com/css2?family=Tinos:ital,wght@0,400;0,700;1,400;1,700&amp;displa
    y=swap" rel="stylesheet"
    />
    <link
    href="https://fonts.googleapis.com/css2?family=DM+Sans:ital,wght@0,400;0,500;0,700;1,400;1,500;1,
    700&amp;display=swap" rel="stylesheet"
    />
    <!-- Core theme CSS (includes Bootstrap)-->
    <link
      href="{{ url_for('static', filename='css/styles.css') }}"
      rel="stylesheet"
    />
  </head>
  {% block body %} {% endblock body %}
</html>
```

upload.html

```
{% extends 'base.html' %} {% block body %}
<body>
  <div class="mission-statement">
    <video
      class="bg-video"
      playsinline="playsinline"
      autoplay="autoplay"
      muted="muted"
      loop="loop"
    >
      <source
        src="{{ url_for('static', filename='assets/video.mp4') }}"
        type="video/mp4"
      />
    </video>
  </div>
  <!-- Masthead-->
  <div class="d-flex justify-content-center card p-3 col-10 col-md-5 m-auto mt-5 shadow-lg" >
    <form class="card-body" action="/predict-value" method="post">
      <h2>Upload Required data</h2>
      <div class="radios py-3 dropdown">
        <p>Home Page Featured:</p>
          <input type="radio" id="yes" name="homepage_featured" value="1" />
        <label for="Yes">Yes</label><br />
          <input
          type="radio"
          id="no"
          name="homepage_featured"
          value="0"
          checked
        />
          <label for="No">No</label><br />
      </div>
      <div class="radios py-3 dropdown">
        <p>Emailer for production:</p>
        <input type="radio" id="yes" name="emailer_for_promotion" value="1" />
        <label for="Yes">Yes</label><br />
        <input
          type="radio"
          id="no"
          name="emailer_for_promotion"
          value="0"
          checked
        />
          <label for="No">No</label><br />
      </div>
      <label for="quantity" class="dropdown py-3"
        >OP_Area (between 2 and 7): <br />
        <input
          type="number"
          id="op_area"
          name="op_area"
          step="0.1"
          min="2"
          max="7"
          class="mt-2"
          required
        />
      </label>
      <!-- Cuisine -->
      <div class="dropdown">
        <p>Cuisine</p>
        <select
          name="cuisine"
          id="cuisine"
          class="btn btn-secondary dropdown-toggle"
        >
          <option value="0">Continental</option>
          <option value="1">Indian</option>
          <option value="2">Italian</option>
          <option value="3">Thai</option>
        </select>
      </div>
      <!-- End cuisine -->
      <!-- City Code -->
      <div class="dropdown py-3">
        <p>City Code</p>
        <select
          name="city_code"
          id="city_code"
          class="btn btn-secondary dropdown-toggle"
        >
          <option value="647">647</option>
          <option value="614">614</option>
          <option value="679">679</option>
          <option value="659">659</option>
        <option value="526">526</option>
        <option value="590">590</option>
        <option value="599">599</option>
        <option value="685">685</option>
        <option value="461">461</option>
        <option value="649">649</option>
        <option value="541">541</option>
        <option value="478">478</option>
        <option value="703">703</option>
        <option value="576">576</option>
        <option value="628">628</option>
        <option value="702">702</option>
        <option value="579">579</option>
        <option value="596">596</option>
        <option value="648">648</option>
        <option value="632">632</option>
        <option value="522">522</option>
        <option value="615">615</option>
        <option value="577">577</option>
        <option value="517">517</option>
        <option value="651">651</option>
        <option value="683">683</option>
        <option value="713">713</option>
        <option value="609">609</option>
        <option value="675">675</option>
        <option value="553">553</option>
        <option value="593">593</option>
        <option value="654">654</option>
        <option value="693">693</option>
        <option value="658">658</option>
        <option value="515">515</option>
        <option value="556">556</option>
        <option value="561">561</option>
        <option value="680">680</option>
        <option value="620">620</option>
        <option value="638">638</option>
        <option value="700">700</option>
        <option value="676">676</option>
        <option value="604">604</option>
        <option value="456">456</option>
        <option value="562">562</option>
        <option value="695">695</option>
        <option value="699">699</option>
        <option value="602">602</option>
        <option value="485">485</option>
        <option value="698">698</option>
        <option value="473">473</option>
       </select>
      </div>
      <!-- End of City Code -->
```

```html
<!-- Region Code -->
<div class="dropdown">
  <p>Region Code</p>
  <select
    name="region_code"
    id="region_code"
    class="btn btn-secondary dropdown-toggle"
  >
    <option value="56">56</option>
    <option value="85">85</option>
    <option value="77">77</option>
    <option value="34">34</option>
    <option value="35">35</option>
    <option value="71">71</option>
    <option value="93">93</option>
    <option value="23">23</option>
  </select>
</div>
<!-- Region Code -->
<!-- Categories -->
<div class="dropdown py-3">
  <p>Categories</p>
  <select
    name="category"
    id="category"
    class="btn btn-secondary dropdown-toggle"
  >
    <option value="0">Beverages</option>
    <option value="1">Biryani</option>
    <option value="2">Desert</option>
    <option value="4">Fish</option>
    <option value="5">Other Snacks</option>
    <option value="6">Pasta</option>
    <option value="7">Pizza</option>
    <option value="8">Rice Bowl</option>
    <option value="9">Salad</option>
    <option value="10">Sandwich</option>
    <option value="11">Seafood</option>
    <option value="12">Soup</option>
    <option value="13">Starters</option>
    <option value="3">Extras</option>
  </select>
</div>
<!-- Categories -->
<div class="d-flex justify-content-center my-4">
  <input
    type="submit"
    value="Predict Orders"
    class="btn btn-secondary"
    id="submit-btn"
```

```
        />
        <div class="spinner-border d-none" role="status" id="loader">
          <span class="sr-only">Loading...</span>
        </div>
      </div>
    </form>
  </div>
  <!-- Bootstrap core JS-->
  <script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
  <!-- Core theme JS-->
  <script src="{{ url_for('static', filename='js/scripts.js') }}"></script>
  <script src="https://cdn.startbootstrap.com/sb-forms-latest.js"></script>
</body>
{% endblock %}
```

7.2. PREDICTION

app.py

```
import pandas as pd
import numpy as np
import pickle
import os
from flask import Flask, request, render_template
import requests
from dotenv import load_dotenv
# Loading up the values
load_dotenv()
template_dir = os.path.abspath('./templates')
app = Flask(__name__, template_folder=template_dir)

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

@app.route('/predict')
def pred():
    return render_template("upload.html")

@app.route('/predict-value', methods=['POST'])
def predict():
    input_features = [float(x) for x in request.form.values()]

    features_name = ['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine', 'city_code',
    'region_code', 'category']

    predicted_value = predict_values(features_name, input_features)

    return render_template('result.html', no_of_orders = int(predicted_value))
```

```
def predict_values(feature_names, feature_values):
    # Access IBM Cloud
    # Get API KEY
    API_KEY = os.environ.get("API_KEY") or None

    # Get MLToken
    token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
    API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
    mltoken = token_response.json()["access_token"]

    # Prepare Payload
    payload = {"input_data": [{"fields": feature_names, "values": [feature_values]}]}

    response = requests.post(

    'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/7f16a51f-5465-40ad-a4d4-85b37976d269/predict
    ions?version=2022-11-18',
        json=payload,
    headers={'Authorization': 'Bearer ' + mltoken})

    return response.json()['predictions'][0]['values'][0][0]

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8000, debug=False)
```

8. TESTING

8.1. TEST CASES:

Test Case	Fail	Pass	Not Tested
Verify if the home page is loaded and responsive	-	✔	-
Verify the predict page is loaded	-	✔	-
Check if the form can handle values.	-	✔	-
Check if the application triggers the model deployed in IBM.	-	✔	-
Verify if the predicted values are accurate.	-	✔	-

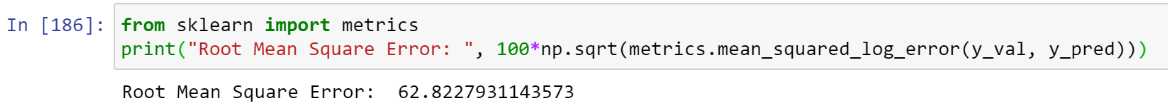
8.2. USER ACCEPTANCE TESTING:

The System is fed with different input values and the output is predicted and compared with the actual result.

S.No	Test Case	Predicted Result (No. of Orders)	Actual Result
1.	**Homepage featured:** No **Emailer For Promotion:** No **OP Area:** 2.0 **Cuisine:** Thai **City code:** 647 **Region code:** 56 **Category:** Beverages	192	181
2.	**Homepage featured:** Yes **Emailer For Promotion:** No **OP Area:** 4.5 **Cuisine:** Continental **City code:** 473 **Region code:** 77 **Category:** Fish	186	169
3.	**Homepage featured:** No **Emailer For Promotion:** No **OP Area:** 2.0 **Cuisine:** Indian **City code:** 647 **Region code:** 56 **Category:** Beverages	38	41

8.3. PERFORMANCE TESTING:

S.No.	Parameter	Values
1.	Metrics	**Classification Model:** Accuray Score
Screenshot		

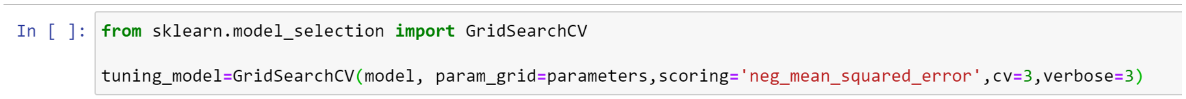


```
In [186]: from sklearn import metrics
          print("Root Mean Square Error: ", 100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

          Root Mean Square Error:  62.8227931143573
```

2.	Tune the Model	Hyperparameter Tuning - Validation Method -

Screenshot



```
In [ ]: from sklearn.model_selection import GridSearchCV

        tuning_model=GridSearchCV(model, param_grid=parameters,scoring='neg_mean_squared_error',cv=3,verbose=3)
```

9. RESULTS

9.1. PERFORMANCE METRICS:

By Implementing the DemandEst - AI powered food demand forecaster, we have observed that the Restaurants can predict the order volume efficiently. The Revenue and profit of the restaurants has been increased gradually.

Scalability: The Application can be scaled so that it can handle huge traffic.

Availability: The System is well tested and it is highly available.

10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- The Application is Cost Free
- Prediction can be done seamlessly
- Very few inputs are required.

DISADVANTAGES:

- It is more Coupled with the deployed model. Hence it relies on the availability of the deployed model.
- The user needs to know the required information for predicting the demand.

11. CONCLUSION

Food demand prediction is an important and challenging problem. To solve this problem, we have used a Decision Tree Regressor to predict the food demand. As we go through different algorithms for prediction the accuracy rate keeps on improving. There was not a big difference other than the precision rate of forecasting.

12. FUTURE SCOPE

This evaluation is used practically for restaurants. Furthermore, in future more refined predictions can be made based on many other factors like cultural habits, religious holidays, consumer preferences etc. In future, this method can be used for predicting work force requirements, automated food ordering based on forecasting results.

13. APPENDIX

GITHUB AND PROJECT DEMO LINK:

Github Link: https://github.com/IBM-EPBL/IBM-Project-38233-1660375482

Project Demo Link:

https://drive.google.com/file/d/1crSR3Ot5wAp-U0CjBZuihppclToxzBKw/view?usp=sharing