Project Report Format

TEAM ID : 8683-1658927046

INTRODUCTION

1.1 Project Overview

The concept of a balance point between supply and demand is used to explain various situations in our daily lives, from bread in the neighborhood bakery, which can be sold at the equilibrium price, which equals the quantities desired by buyers and sellers, to the negotiation of securities of companies in the stock market. On the supply side, a definition of the correct price to be practiced and mainly the quantity are common issues in the planning and execution of the strategy of several companies. The projection of demand is often built through historical sales data, growth prospects for the sector or even targets set to engage sales of a certain product. The expansion of a sector does not result in a growth of the same magnitude for the entire product mix.

1.2 Purpose

During the S&OP process, demand forecasting supported by AI facilitates the work of the marketing and sales areas, as well as reducing uncertainty and increasing predictability for the supply chain areas. In the S&OE process, AI can be used to identify new opportunities and to correct deviations from what was planned.

2. LITERATURE SURVEY

2.1 Existing problem

For companies, mainly large ones with economies of scale and geographic capillarity, an error in the forecast of demand can cause several consequences, such as:

- Stock break;
- Perishable waste (What is FIFO?);
- Drop in production;
- Idle stock (slow moving)
- Pricing errors

2.2 Reference

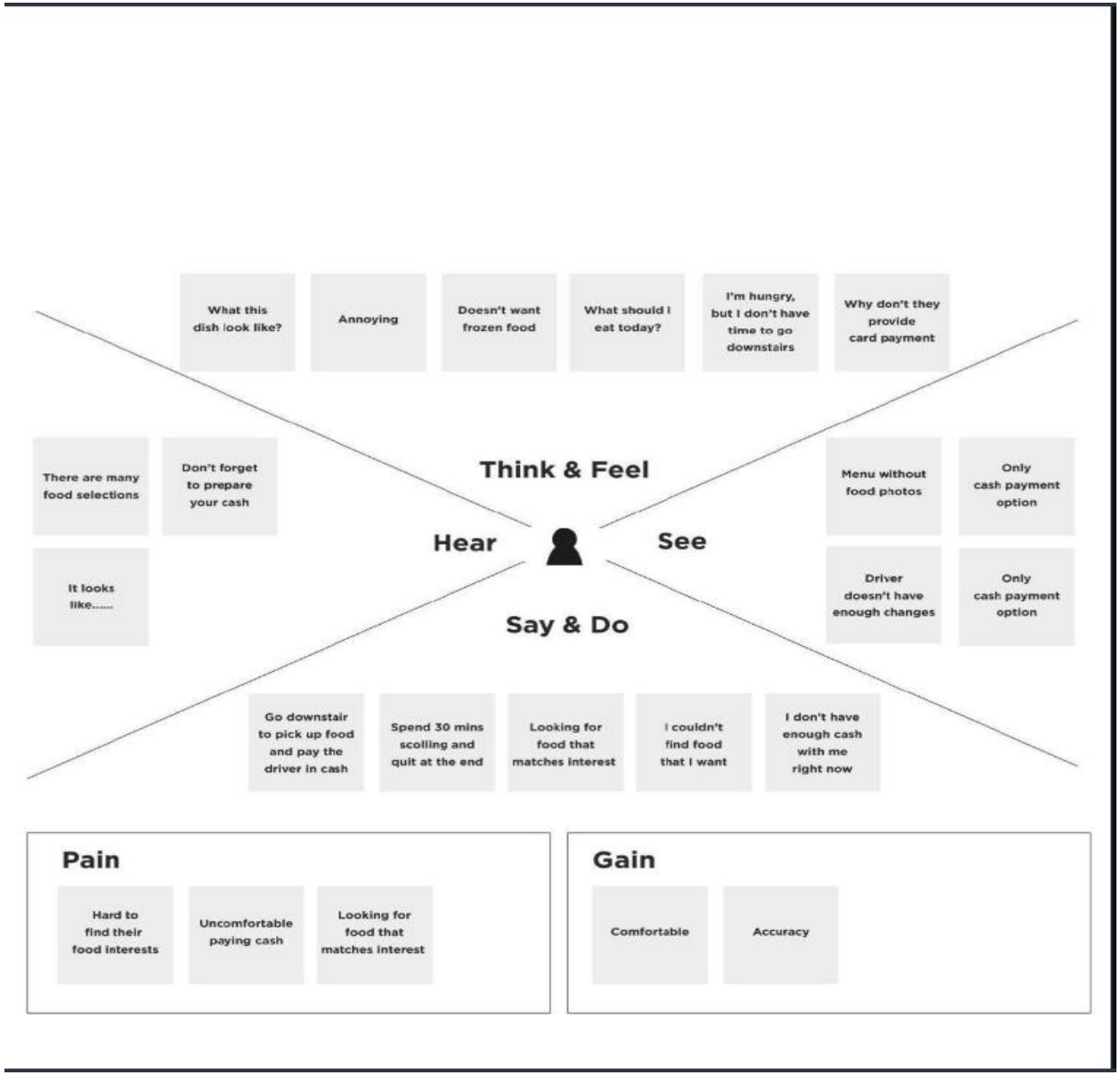
- AQUAREL
- 09Solution
- Kaggle

2.3 Problem Statement Definition

Demand forecasting is a key component to every growing online business. Without proper demand forecasting processes in place, it can be nearly impossible to have the right amount of stock on hand at any given time. A food delivery service has to deal with a lot of perishable raw materials which makes it all the more important for such a company to accurately forecast daily and weekly demand.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy map canvas


What this dish look like?
Annoying
Doesn't want frozen food
What should I eat today?
I'm hungry, but I don't have time to go downstairs
Why don't they provide card payment
There are many food selections
Don't forget to prepare your cash
Think & Feel
Hear
See
Menu without food photos
Only cash payment option
It looks like.....
Driver doesn't have enough changes
Only cash payment option
Say & Do
Go downstair to pick up food and pay the driver in cash
Spend 30 mins scolling and quit at the end
Looking for food that matches interest
I couldn't find food that I want
I don't have enough cash with me right now
Pain
Hard to find their food interests
Uncomfortable paying cash
Looking for food that matches interest
Gain
Comfortable
Accuracy

3.2 proposed solution

Project Design Phase-I
Proposed Solution Template

Date	9 October 2022
Team ID	PNT2022TMID00524
Project Name	DemandEst - AI powered Food Demand Forecaster
Maximum Marks	2 Marks

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description

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 majority of raw materials is done on 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. 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 next ten weeks. To achieve this, we should know the information about of fulfilment Center like area, city etc., and meal information like category of food sub category of food price of the food or discount in 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.	Idea / Solution description	This section describes the model developed to solve the stated aggregate production planning problem. Prior to this project, the company has already reviewed its demand forecasting process that now provides reliable aggregate monthly demand forecasts for production planning. Demand forecasts and the other input data as costs and production rates should be carefully estimated; otherwise, the results achieved by the aggregate production planning model will be useless

3.	Novelty / Uniqueness	▪ Artificial intelligence ▪ Machine learning
4.	Social Impact / Customer Satisfaction	Customer (help to predict food requirement ,and satisfy the customer)
5.	Business Model (Revenue Model)	Predictive Model.
6.	Scalability of the Solution	It performs the detection in an accurate manner. It works efficiently.

3.3 Problem Solution fit

4. PROJECT DESIGN

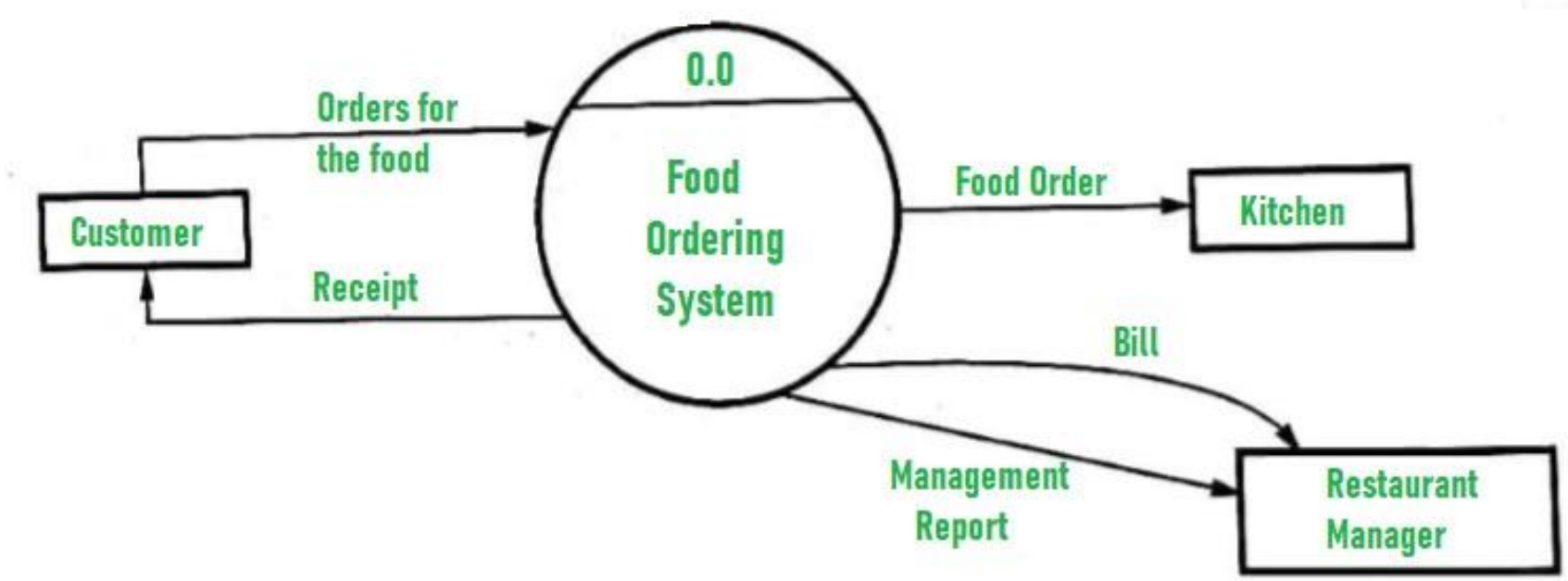
4.1 Data flow diagram

Project Design Phase-II
Data Flow Diagram & User Stories

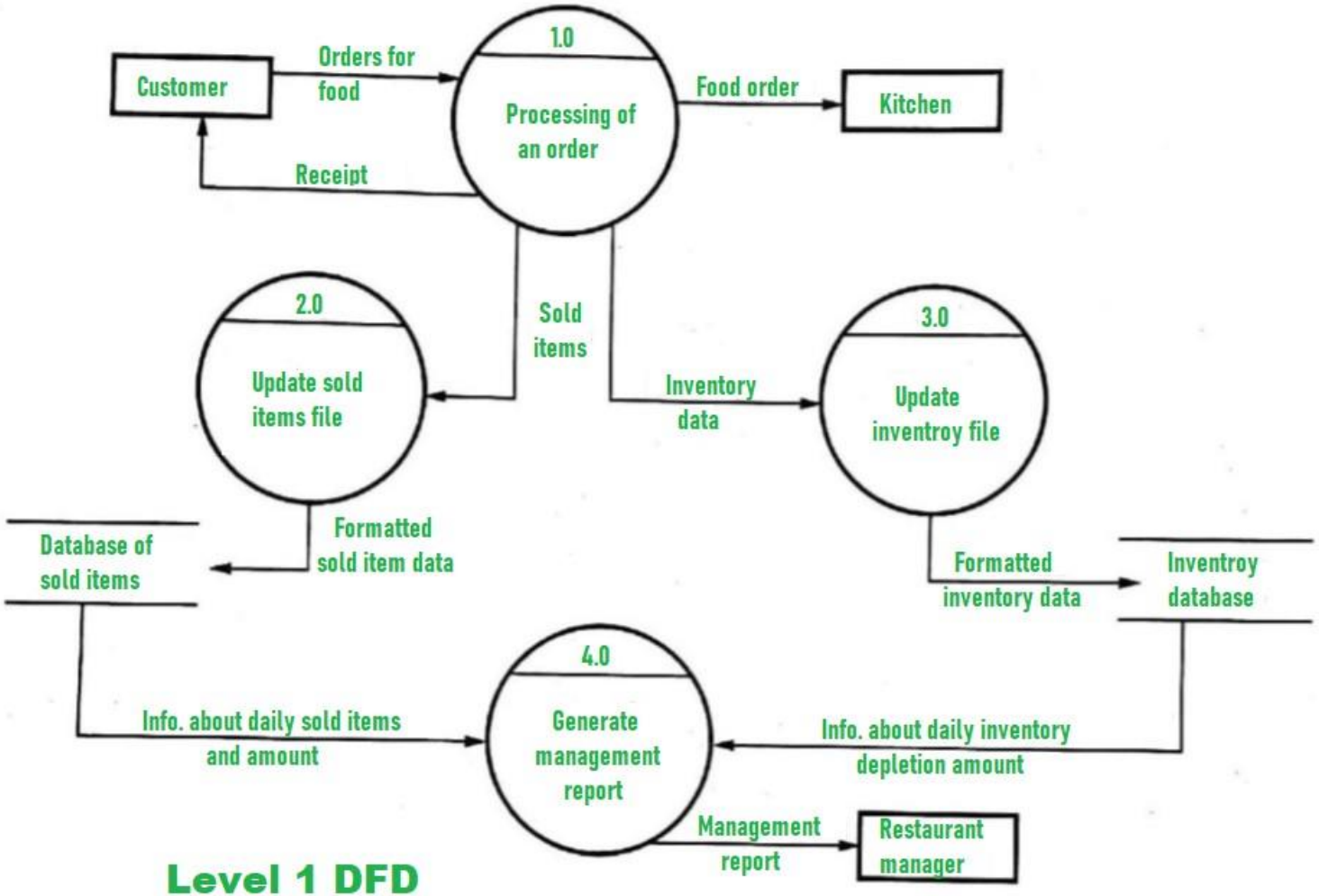
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		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
		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
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard through Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can login to the application by entering respective email & password.	High	Sprint-1
	Dashboard	USN-6	As a user, I can access all the services provided in the dashboard.	I can predict the orders for next 10 weeks and I estimate of raw materials for the same.	High	Sprint-1
Customer (Web user)	Login & Dashboard	USN-8	As a user, I can login through web application and access the resources in the dashboard.	I can login with the credentials required and I can access the services	High	Sprint-1
User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
				provided through web application.		
Customer Care Executive	Support	USN-9	As a user I can get support from the help desk and can get my queries cleared.	I can get guidance and any support to use the application.	High	Sprint-2
Administrator	Management	USN-10	As an admin I can maintain the application.	I can perform maintenance of the app even after the release.	Medium	Sprint-1

		USN-11	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-12	As an admin I can update the features of the app and upgrade it to better versions .	I can perform upgrading of features and versions.	Medium	Sprint-1
		USN-13	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

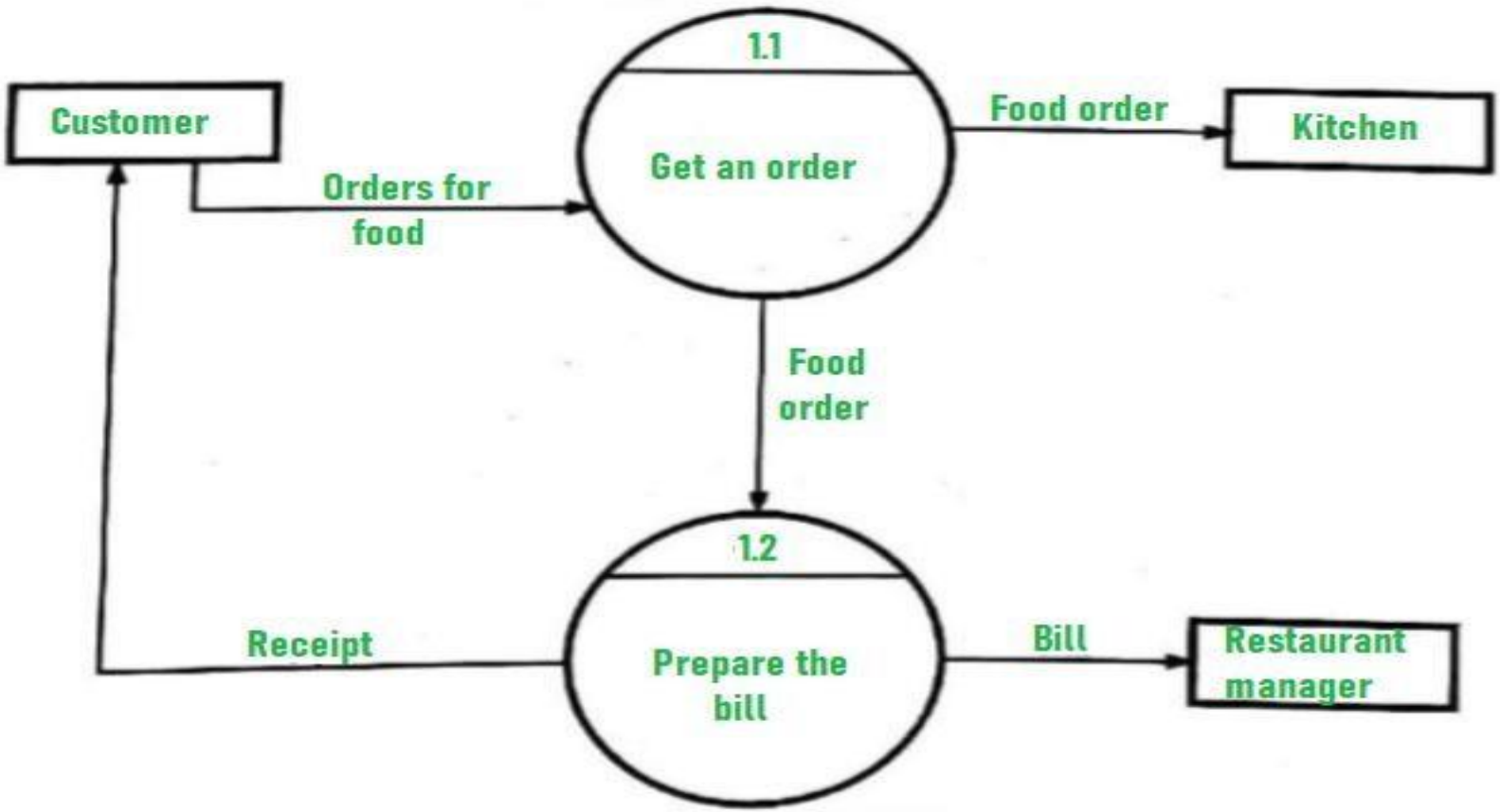
Date	11 November 2022
Team ID	PNT2022TMID00524
Project Name	DemandEst – AI Powered Food Demand Forecaster.
Maximum Marks	4 Marks



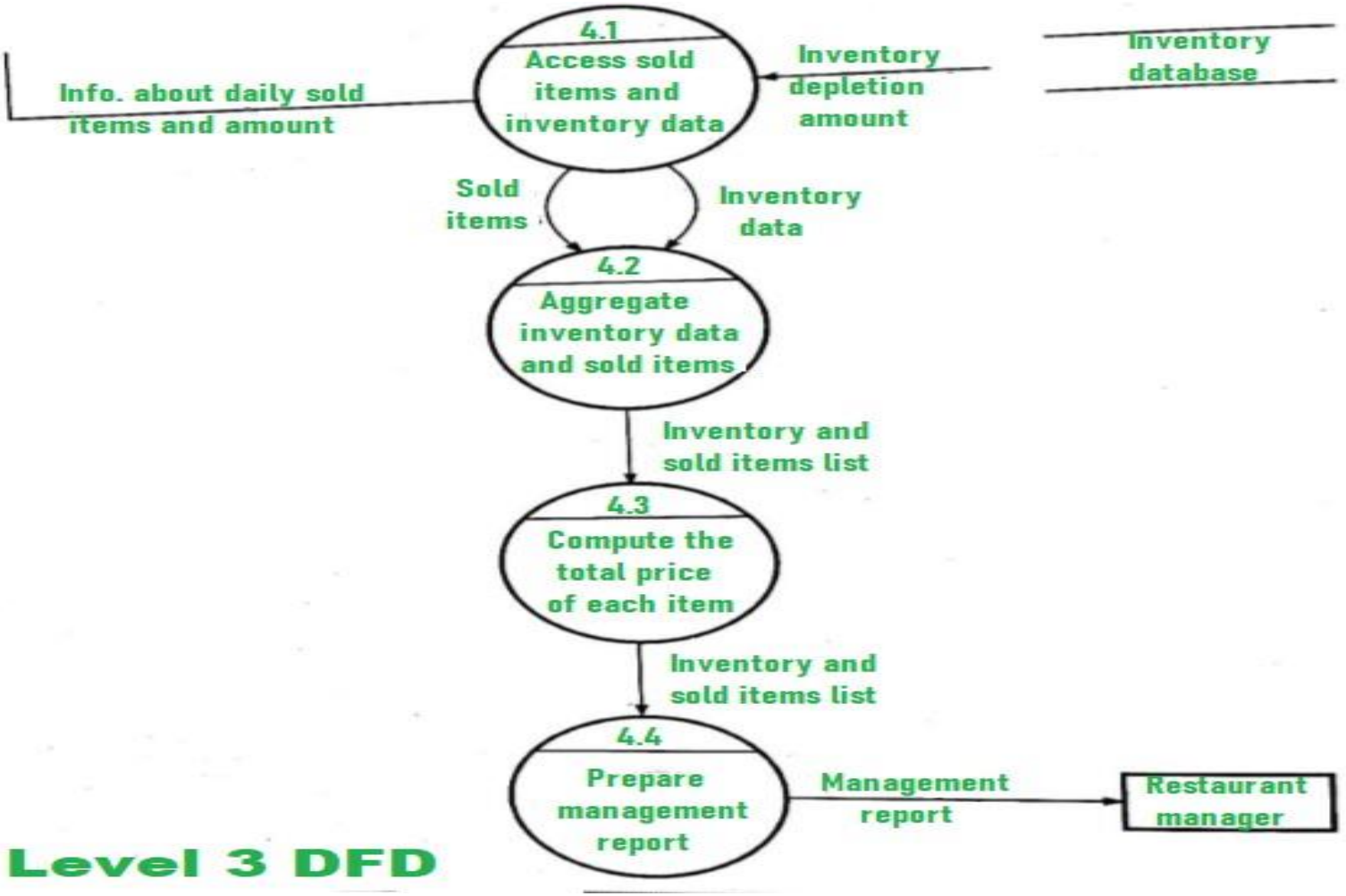
Level 0 DFD (Context Level


1.0
Processing of an order
Customer
Orders for food
Food order
Kitchen
Receipt
2.0
Update sold items file
Sold items
3.0
Update inventroy file
Inventory data
Database of sold items
Formatted sold item data
Formatted inventory data
Inventroy database
4.0
Generate management report
Info. about daily sold items and amount
Info. about daily inventory depletion amount
Management report
Restaurant manager

Level 1 DFD


1.1
Customer
Get an order
Food order
Kitchen
Orders for food
Food order
1.2
Receipt
Prepare the bill
Bill
Restaurant manager

Level 2 DFD


4.1
Access sold items and inventory data
Info. about daily sold items and amount
Inventory depletion amount
Inventory database
Sold items
Inventory data
4.2
Aggregate inventory data and sold items
Inventory and sold items list
4.3
Compute the total price of each item
Inventory and sold items list
4.4
Prepare management report
Management report
Restaurant manager

Level 3 DFD

5. PROJECT PLANNING & SCHEDULING

5.1 Sprint planning & Est

Sprint 1:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link type="text/css" rel="stylesheet" href="/Flask/static/style.css">
   <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;600;800&display=swap" rel="stylesheet">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/css/all.min.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/css/v4-shims.min.css">
<style>
body, html {
  height: 100%;
```

```
  margin: 0;
  font-family: 'Poppins', sans-serif;
}
* {
  box-sizing: border-box;
}

.bg-image {

  background-image: url("https://d27k8xmh3cuzik.cloudfront.net/wp-content/uploads/2018/06/belgian-waffles-og-image.jpg");
  filter: blur(8px);
  -webkit-filter: blur(8px);
  height: 100%;
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}
```

```
.bg-text {
  background-color: rgb(0,0,0);
  background-color: rgba(0,0,0, 0.4);
  color: white;
  font-weight: bold;
  border: 3px solid #f1f1f1;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  z-index: 2;
  width: 80%;
  padding: 20px;
  text-align: center;
}
ul {
  list-style-type: none;
  margin: 0;
```

```
  padding: 0;
  overflow: hidden;
  background-color:gray;
}

li {
  float: right;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

li a:hover {
```

```
  background-color:gray;
}
</style>
</head>
<body>


    <ul>
        <li style="font-size:20px;"><a href="./home.html">Home</a></li>
        <li style="font-size:20px;"><a href="./upload.html">Predict</a></li>
    </ul>
<div class="bg-image"></div>
<div class="bg-text">
  <h2>About Us</h2>
  <h1 style="font-size:50px">Food Demand Forecasting</h1>
  <p>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 majority of raw materials is done on 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.</p>
```

```
</div>

</body>
</html>
```

ii)

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body, html {
  height: 100%;
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
}

* {
```

```
  box-sizing: border-box;
}



.bg-image {
  background-image: url("https://who-euro.shorthandstories.com/food-and-nutrition-tips-during-self-quarantine/assets/WoizMxIOjR/cover-photo-2560x1922.jpeg");



  filter: blur(8px);
  -webkit-filter: blur(8px);



  height: 100%;

  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}
```

```
.bg-text {
  background-color: rgb(0,0,0);
  background-color: rgba(0,0,0, 0.4);
  color: white;
  font-weight: bold;
  border: 3px solid #f1f1f1;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  z-index: 2;
  width: 80%;
  padding: 20px;
  text-align: center;
}
```

```
</style>
</head>
<body>

<div class="bg-image"></div>

<div class="bg-text">
  <h2> Home </h2>
  <h1 style="font-size:50px">Food Demand Forecasting</h1>
<p> Demand forecasting is the process in which
historical data is used to estimate the quantity of product
customer will purchase. This prediction activity is used in
many fields like retailing, food industry etc. In Restaurants,
prediction play a vital role as most of the basic ingredients
have short-shelf life. The demands depend upon many explicit
and hidden context such as season, region etc. We will be considering
number of order is used to forecast stock of items, using
machine learning with internal and external data. We will be predicting
```

in such a way that it is capable of overpowering the wastage of short life

items.

</p>

</body>

</html>

iii)

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>predict</title>
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;600;800&display=swap"
rel="stylesheet">
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/css/all.min.css">
<style>
body, html {
  height: 100%;
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;

}

* {
  box-sizing: border-box;

}

.bg-image {

  background-image: url("https://img.freepik.com/free-photo/chicken-wings-barbecue-sweetly-sour-sauce-picnic-summer-menu-tasty-food-top-view-flat-lay_2829-6471.jpg?w=2000");
```

```
  filter: blur(8px);
  -webkit-filter: blur(8px);



  height: 100%;



  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}



.bg-text {
  background-color: rgb(0,0,0);
  background-color: rgba(0,0,0, 0.4);
  color: white;
  font-weight: bold;
```

```
  border: 3px solid #f1f1f1;
  border-radius: 25px;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  z-index: 2;
  width: 80%;
  padding: 20px;
  text-align: center;
}
</style>
</head>
<body>

<div class="bg-image"></div>

<div class="bg-text">
```

```
<div class="container">
        <center> <div id="content" style="margin-top:2em">
        <h1><center>Food Demand Forecasting</center></h1>
            <form action="{{ url_for('predict') }}" method="POST">

  <select id="homepage_featured" name="homepage_featured">
   <option value="">homepage_featured</option>
     <option value="0">No</option>
     <option value="1">Yes</option>

    </select><br><br>
  <select id="emailer_for_promotion" name="emailer_for_promotion">
   <option value="">emailer_for_promotion</option>
     <option value="0">No</option>
     <option value="1">Yes</option>

    </select><br><br>
```

```
<input class="form-input" type="text" name="op_area" placeholder="Enter the op_area(2-7)"><br><br>
    <select id="cuisine" name="cuisine">
    <option value="">Cuisine</option>
       <option value="0">Continental</option>
       <option value="1">Indian</option>
       <option value="2">Italian</option>
       <option value="3">Thai</option>

     </select><br><br>
     <input class="form-input" type="text" name="city_code" placeholder="Enter city_code"><br><br>
    <input class="form-input" type="text" name="region_code" placeholder="Enter region_code"><br><br>
    <select id="category" name="category">
    <option value="">Category</option>
       <option value="0">Beverages</option>
       <option value="1">Biryani</option>
       <option value="2">Desert</option>
       <option value="3">Extras</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>
</select><br><br>

            <input type="submit" class="my-cta-button" value="Predict">
          </form>

  <br>
   <h1 class="predict">Demand is: {{ prediction_text }}</h1>
     </div></center>
  </div>
```

```
</div>
</div>

</body>
</html>
```

5.2 Sprint delivery schedule

Sprint 2:

Ibmapp:

```
import pandas as pd
import numpy as np
import pickle
import os
from flask import Flask,request, render_template
app=Flask(__name__,template_folder="templates")
@app.route('/', methods=['GET'])
def index():
```

```
    return render_template('home.html')
@app.route('/home', methods=['GET'])
def about():
    return render_template('home.html')
@app.route('/predict',methods=['GET'])
def page():
    return render_template('upload.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    print("[INFO] loading model...")
    model = pickle.load(open('fdemand.pkl', 'rb'))
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]
    print(features_value)

    features_name = ['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine',
      'city_code', 'region_code', 'category']
    prediction = model.predict(features_value)
```

```
    output=prediction[0]
    print(output)
    return render_template('upload.html', prediction_text=output)


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

iv) main.py

```
import pandas as pd
import numpy as np
import requests
import os


from flask import Flask,request, render_template
app=Flask(__name__,template_folder='templates')
```

```
@app.route('/',methods=['GET'])
def index():
    return render_template('index.html')
@app.route('/home',methods=['GET'])
def about():
    return render_template('intro.html')
@app.route('/pred',methods=['GET'])
def page():
    return render_template('upload.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    print("[INFO] loading model...")
    input_features = [float(x) for x in request.form.values()]
    features_value = [input_features]
    print(features_value)

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

```
    # NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
    API_KEY = "V0FedIvcsn9vpDN7cIG2cmB8T8zpenX6vPs8tufhqE6b"
    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"]
    header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
    # NOTE: manually define and pass the array(s) of values to be scored in the next line
    payload_scoring = {"input_data": [{"values": features_value}]}
    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/cfbed64a-29cb-44e2-
bc53-e0a418c3077e/predictions?version=2022-11-14', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring Endpoint")
    print(response_scoring.json())
    pred = response_scoring.json()

    output=pred['predictions'][0]['values'][0][0]
    print(output)
    return render_template('upload.html', prediction_text=output)
```

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	RAJESH KANNA J S MANOJ S KISHORE S S MANOJ KUMAR V A R
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	RAJESH KANNA J S MANOJ S KISHORE S S MANOJ KUMAR V A R
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	RAJESH KANNA J S MANOJ S KISHORE S S MANOJ KUMAR V A R
Sprint-2		USN-4	As a user, I can register for the application through Gmail	2	Medium	RAJESH KANNA J S MANOJ S KISHORE S S MANOJ KUMAR V A R
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	RAJESH KANNA J S MANOJ S KISHORE S S MANOJ KUMAR V A R
Sprint-1	Dashboard	USN-6	As a user, I can access the services and information provided in the dashboard	2	High	RAJESH KANNA J S MANOJ S KISHORE S S MANOJ KUMAR V A R
Sprint-1	Login	USN-7	As a user, I can log into the web application and access the dashboard	1	High	RAJESH KANNA J S MANOJ S KISHORE S S MANOJ KUMAR V A R

Sprint-4	Helpdesk	USN-8	As a user, I can get the guidance from the customer care	1	High	RAJESH KANNA J S MANOJ S KISHORE S S MANOJ KUMAR V A R
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-3	Management	USN-9	As an administrator, I can collect new datasets and keep the model trained	2	High	RAJESH KANNA J S MANOJ S KISHORE S S MANOJ KUMAR V A R
Sprint-3		USN-10	As an administrator, I can update other features of the application	2	Medium	RAJESH KANNA J S MANOJ S KISHORE S S MANOJ KUMAR V A R
Sprint-3		USN-11	As an administrator, I can maintain the information about the user	2	Medium	RAJESH KANNA J S MANOJ S KISHORE S S MANOJ KUMAR V A R
Sprint-4		USN-12	As an administrator, I can maintain third-party services	1	Low	RAJESH KANNA J S MANOJ S KISHORE S S MANOJ KUMAR V A R

```
if __name__ == '__main__':
    app.run()
```

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	11 November 2022
Team id	**PNT2022TMID00524**
Project Name	Project – DemandEst - AI Powered Food Demand Forecaster
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)	
Sprint-1	7	6 Days	24 Oct 2022	29 Oct 2022	7	29 Oct 2022	
Sprint-2	4	6 Days	31 Oct 2022	05 Nov 2022	4	05 Nov 2022	
Sprint-3	6	6 Days	07 Nov 2022	12 Nov 2022	6	12 Nov 2022	
Sprint-4	2	6 Days	14 Nov 2022	19 Nov 2022	2	19 Nov 2022	

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

AV (Sprint 1) = 7/6 = 1

AV (Sprint 2) = 4/6 = 1

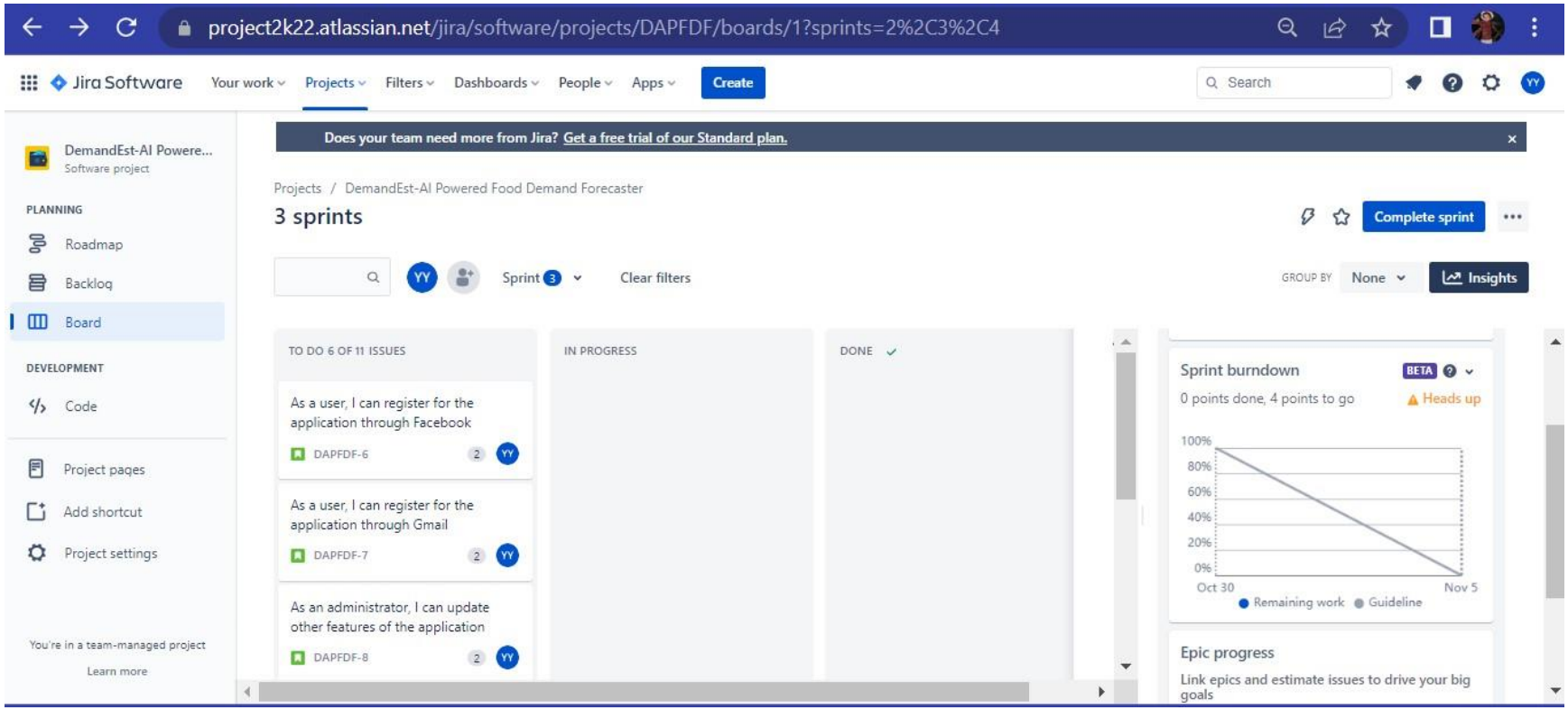
AV (Sprint 3) = 6/6 = 1

AV (Sprint 4) = 2/6 = 1

AV (Total) = 21/24 = 1

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



7. CODING & SCHEDULING:(Explain the feature added in the project along with code)

a. Feature 1:

Home.html:

<!DOCTYPE html>

```
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link type="text/css" rel="stylesheet" href="/Flask/static/style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;600;800&display=swap" rel="stylesheet">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/css/all.min.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/css/v4-shims.min.css">
<style>
body, html {
  height: 100%;
  margin: 0;
  font-family: 'Poppins', sans-serif;
}
* {
  box-sizing: border-box;
```

```
}

.bg-image {

  background-image: url("https://d27k8xmh3cuzik.cloudfront.net/wp-content/uploads/2018/06/belgian-waffles-og-image.jpg");
  filter: blur(8px);
  -webkit-filter: blur(8px);
  height: 100%;
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}

.bg-text {
  background-color: rgb(0,0,0);
  background-color: rgba(0,0,0, 0.4);
  color: white;
  font-weight: bold;
```

```
  border: 3px solid #f1f1f1;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  z-index: 2;
  width: 80%;
  padding: 20px;
  text-align: center;
}
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color:gray;
}
```

```
li {
  float: right;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

li a:hover {
  background-color:gray;
}
</style>
</head>
<body>
```

```
    <ul>
        <li style="font-size:20px;"><a href="./home.html">Home</a></li>
        <li style="font-size:20px;"><a href="./upload.html">Predict</a></li>
    </ul>
<div class="bg-image"></div>
<div class="bg-text">
  <h2>About Us</h2>
  <h1 style="font-size:50px">Food Demand Forecasting</h1>
  <p>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 majority of raw materials is done on 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.</p>
</div>


</body>
</html>
```

Upload.html:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body, html {
  height: 100%;
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
}

* {
  box-sizing: border-box;
}


.bg-image {
  background-image: url("https://who-euro.shorthandstories.com/food-and-nutrition-tips-during-self-quarantine/assets/WoizMxIOjR/cover-photo-2560x1922.jpeg");
```

```
  filter: blur(8px);
  -webkit-filter: blur(8px);


  height: 100%;

  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}


.bg-text {
  background-color: rgb(0,0,0);
  background-color: rgba(0,0,0, 0.4);
  color: white;
```

```
  font-weight: bold;
  border: 3px solid #f1f1f1;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  z-index: 2;
  width: 80%;
  padding: 20px;
  text-align: center;
}

</style>
</head>
<body>

<div class="bg-image"></div>
```

```
<div class="bg-text">
  <h2> Home </h2>
  <h1 style="font-size:50px">Food Demand Forecasting</h1>
<p> Demand forecasting is the process in which
historical data is used to estimate the quantity of product
customer will purchase. This prediction activity is used in
many fields like retailing, food industry etc. In Restaurants,
prediction play a vital role as most of the basic ingredients
have short-shelf life. The demands depend upon many explicit
and hidden context such as season, region etc. We will be considering
number of order is used to forecast stock of items, using
machine learning with internal and external data. We will be predicting
in such a way that it is capable of overpowering the wastage of short life
items.
</p>

</div>
```

```
</body>
</html>
```

App.py:

```
import pandas as pd
import numpy as np
import pickle
import os
from flask import Flask,request, render_template
app=Flask(__name__,template_folder="templates")
@app.route('/', methods=['GET'])
def index():
    return render_template('home.html')
@app.route('/home', methods=['GET'])
def about():
    return render_template('home.html')
@app.route('/predict',methods=['GET'])
def page():
    return render_template('upload.html')
```

```
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    print("[INFO] loading model...")
    model = pickle.load(open('fdemand.pkl', 'rb'))
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]
    print(features_value)

    features_name = ['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine',
      'city_code', 'region_code', 'category']
    prediction = model.predict(features_value)
    output=prediction[0]
    print(output)
    return render_template('upload.html', prediction_text=output)


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

Ibmapp.py:

```
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
import sklearn as sk
train = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")
train.head()
test.head()
train.info()
train['num_orders'].describe()
train.isnull().sum()
meal_info = pd.read_csv("meal_info.csv")
center_info = pd.read_csv("fullfilment_center_info.csv")
trainfinal = pd.merge(train, meal_info, on="meal_id", how="outer")
trainfinal = pd.merge(trainfinal, center_info, on="center_id", how="outer")
```

In [ ]:

```
trainfinal.head()
trainfinal=trainfinal.drop(['center_id','meal_id'],axis=1)
trainfinal.head()
cols=trainfinal.columns.tolist()
print(cols)
cols=cols[:2]+cols[9:]+cols[7:9]+cols[2:7]
print(cols)
trainfinal=trainfinal[cols]
trainfinal.dtypes
```

```
from sklearn.preprocessing import LabelEncoder
```

In [ ]:

```
lb1=LabelEncoder()
trainfinal['center_type']=lb1.fit_transform(trainfinal['center_type'])
lb2=LabelEncoder()
trainfinal['category']=lb1.fit_transform(trainfinal['category'])
lb1=LabelEncoder()
trainfinal['cuisine']=lb1.fit_transform(trainfinal['cuisine'])
```

In [ ]:

```
trainfinal.head()
```

```
trainfinal.shape
plt.style.use('fivethirtyeight')
plt.figure(figsize=(12,7))
sb.distplot(trainfinal.num_orders,bins=25)
plt.xlabel("num_orders")
plt.ylabel("Number of Buyers")
plt.xlabel("num_orders Distribution")
trainfinal2=trainfinal.drop(['id'],axis=1)
correlation=trainfinal2.corr(method='pearson')
columns=correlation.nlargest(8,'num_orders').index
columns
```

```
Index(['num_orders', 'homepage_featured', 'emailer_for_promotion', 'op_area',
       'cuisine', 'city_code', 'region_code', 'category'],
      dtype='object')
```

```
correlation_map=np.corrcoef(trainfinal[columns].values.T)
sb.set(font_scale=1.0)
```

```
heatmap=sb.heatmap(correlation_map, cbar=True, annot=True, square=True, fmt='.2f',
yticklabels=columns.values, xticklabels=columns.values)

features=columns.drop(['num_orders'])
trainfinal3=trainfinal[features]
x=trainfinal3.values
y=trainfinal['num_orders'].values


trainfinal3.head()

from sklearn.model_selection import train_test_split
x_train, x_val, y_train, y_val=train_test_split(x,y,test_size=0.25)
```

main.py:

```
import pandas as pd

import numpy as np

import requests

import os



from flask import Flask,request, render_template

app=Flask(__name__,template_folder='templates')
```

```
@app.route('/',methods=['GET'])
def index():
    return render_template('index.html')
@app.route('/home',methods=['GET'])
def about():
    return render_template('intro.html')
@app.route('/pred',methods=['GET'])
def page():
    return render_template('upload.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    print("[INFO] loading model...")
    input_features = [float(x) for x in request.form.values()]
    features_value = [input_features]
    print(features_value)

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

```
    # NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
    API_KEY = "V0FedIvcsn9vpDN7cIG2cmB8T8zpenX6vPs8tufhqE6b"
    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"]
    header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
    # NOTE: manually define and pass the array(s) of values to be scored in the next line
    payload_scoring = {"input_data": [{"values": features_value}]}
    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/cfbed64a-29cb-44e2-
bc53-e0a418c3077e/predictions?version=2022-11-14', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring Endpoint")
    print(response_scoring.json())
    pred = response_scoring.json()

    output=pred['predictions'][0]['values'][0][0]
    print(output)
    return render_template('upload.html', prediction_text=output)
   import pandas as pd
```

```
import numpy as np
import requests
import os



from flask import Flask,request, render_template
app=Flask(__name__,template_folder='templates')

@app.route('/',methods=['GET'])
def index():
    return render_template('index.html')
@app.route('/home',methods=['GET'])
def about():
    return render_template('intro.html')
@app.route('/pred',methods=['GET'])
def page():
    return render_template('upload.html')
@app.route('/predict', methods=['GET', 'POST'])
```

```
def predict():
    print("[INFO] loading model...")
    input_features = [float(x) for x in request.form.values()]
    features_value = [input_features]
    print(features_value)


    features_name = ['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine',
      'city_code', 'region_code', 'category']
    # NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
    API_KEY = "V0FedIvcsn9vpDN7cIG2cmB8T8zpenX6vPs8tufhqE6b"
    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"]
    header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
    # NOTE: manually define and pass the array(s) of values to be scored in the next line
    payload_scoring = {"input_data": [{"values": features_value}]}
    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/cfbed64a-29cb-44e2-
bc53-e0a418c3077e/predictions?version=2022-11-14', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
```

```
print("Scoring Endpoint")
print(response_scoring.json())
pred = response_scoring.json()

output=pred['predictions'][0]['values'][0][0]
print(output)
return render_template('upload.html', prediction_text=output)
```

Result:

c. Performance Metrics – he evaluation metric for this competition is 100*RMSLE where RMSLE is Root of Mean Squared Logarithmic Error across all entries in the test set where our accuracy 92% , rsme – 0.8934

8. ADVANTAGES & DISADVANTAGES

ADVANTAGE:

- In supply chain networks, demand forecasting with the aid of AI-based techniques can cut errors by 30 to 50 percent. By implementing these approaches, organisations may be able to forecast accurately at all levels.

DIS-ADVANTAGE:

- Not every situation can be predicted

9. CONCLUSION

Therefore, this complete representation shows the progress on the topic in an systematically view .This implementation along with several code has separate topics to evolve around for the best outome as a report.

10. FUTURE SCOPE

Predictions , availability, Scalability , Demand , everything will be followed on a correct procedure .

11. APPENDIX :

https://github.com/IBM-EPBL/IBM-Project-8683-1658927046