Internship Report

On

FLAVOR FORECAST

- Personalized Recipe Suggestion Based on Mood and Weather

Submitted by

Chatla Chetan Kumar
21K91A6622
TKR College of Engineering and Technology

Submitted to

Mallika Srivastava
Head, Training Delivery
Eisystems Services

&
Mayur Dev Sewak
Head, Internships & Trainings
Eisystems Services

Student's Declaration

I, CHATLA CHETAN KUMAR, a student of B. Tech program, Roll No. 21K91A6622 of the Department of CSE (AI & ML) College do hereby declare that I have completed the mandatory internship in Eisystems Technologies under the faculty guideship of Dr. D JAGADISWARY, Department of CSE (AI & ML), TKR College of Engineering and Technology.

(Signature and Date)

Endorsements

Dr. D JAGADISWARY
Department of CSE (AI & ML)
TKR College of Engineering and Technology

Dr. B Sunil Srinivas
Department of CSE (AI & ML)
TKR College of Engineering and Technology

Table of Content

List of Figures

List of Tables

Executive Summary

This internship spanned a period of eight weeks, during which I was part of both the training and project phases at EISystems, focusing on course Python with Machine Learning. The primary objective of this internship was to gain practical exposure and applying theoretical knowledge to real-world scenarios, and enhancing my professional skills.

Learning Objectives:

The main Objective to take this internship in field of Python with Machine Learning is to be able to formulate machine learning problems corresponding to different applications and range of machine learning algorithms along with their strengths and weaknesses.

Learning Outcomes:

Upon completion of the internship, I achieved the following outcomes:

- **Technical Proficiency:** Enhanced my skills in field of Machine Learning and gained hands-on experience by doing Project.
- **Project Management:** Learned to manage time and resources effectively, contributing to the successful completion of my Flavor Forecast Project.
- **Professional Development:** Improved problem-solving skills and logical thinking ability.

Summary of Activities:

- During the training phase,
  - I have studied in detail about the Python Programming which includes foundations of python, variables, data types, data structures etc.
  - After learning about the Python Programming, I have studied the various libraries which are used for machine learning and learned about the various machine learning model such as linear regression, classifications etc.
- During the project phase,
  - We have team up and chosen the project named Flavor Forecast which is a Personalized Recipe Suggestion Based on Mood and Weather where I undertook the following key activities:
    - Created the required dataset manually by considering the key moods and the weather conditions.
    - Developed a machine learning model to train and test the dataset which was created manually.
    - Collaborated with team members to troubleshoot and resolve issues, ensuring the project's successful completion.

Overview of Organization

EISystems

India's leader in workshops & trainings at IITs, NITs & top engineering colleges

EISystems Services is a leading Indian technology identity with operations across India. EISystems (We call it EISys) offers trainings in Cybersecurity, Machine Learning, Automobiles, Internet of Things, Robotics and social media for enterprises and student community. Till date we have trained approximately 50000 students and impacted around 2 lakhs students through our various outreach initiatives since our founding.

Project Summary

Objective:

People often seek different types of food based on how they feel. For example, someone feeling happy might crave something light and refreshing, while someone feeling sad might prefer comfort food. By incorporating mood into the recipe recommendation process, the application aims to offer more relevant and satisfying suggestions.

Weather conditions also play a crucial role in our food choices. On a cold, rainy day, a person might prefer a warm, hearty meal, whereas on a hot, sunny day, they might lean towards something cool and hydrating. Integrating weather data ensures that the recipe suggestions are contextually appropriate, enhancing the user's overall experience.

The inspiration for the project Flavor Forcast comes from the understanding that our emotions and environmental factors significantly influence our food preferences and cravings.

Abstract:

The project Flavor Forecast aims to develop a machine learning-powered web application that suggests recipe based on the user's mood and current weather conditions. By leveraging data preprocessing techniques and a Random Forest Classifier, the system predicts the most suitable recipe for the user. The application is built using Python's Flask web framework, making it accessible through a user-friendly web interface.

Software Requirements:

1. Programming Language

- Python: The language used for data processing, machine learning, and backend development.

2. Libraries and Frameworks

- Pandas: For data manipulation and analysis.
- Scikit-Learn: For machine learning model building, including preprocessing, training, and evaluation.
- Flask: A lightweight web framework for building the web application.
- Numpy: For numerical operations and array handling.

3. Machine Learning Libraries

- RandomForestClassifier: Part of Scikit-Learn, used for building the classification model.
- LabelEncoder: From Scikit-Learn, used to encode target labels with value between 0 and n_classes-1.
- OneHotEncoder: From Scikit-Learn, used to convert categorical data to a format that can be provided to ML algorithms.

- StandardScaler: From Scikit-Learn, used to standardize features by removing the mean and scaling to unit variance.

4. Development Environment

- Jupyter Notebook: For developing and experimenting with the data preprocessing, model training, and evaluation.
- IDE/Text Editor: Visual Studio Code

7. HTML/CSS/JavaScript

- HTML: For creating the web form and user interface.
- CSS: For styling the web pages.
- JavaScript: For adding interactivity to the web pages, including AJAX for asynchronous form submission.

Hardware Requirements:

Devices : Computer or Laptop

RAM: as per requirement

Hard Disk: as per requirement

Working of this Project:

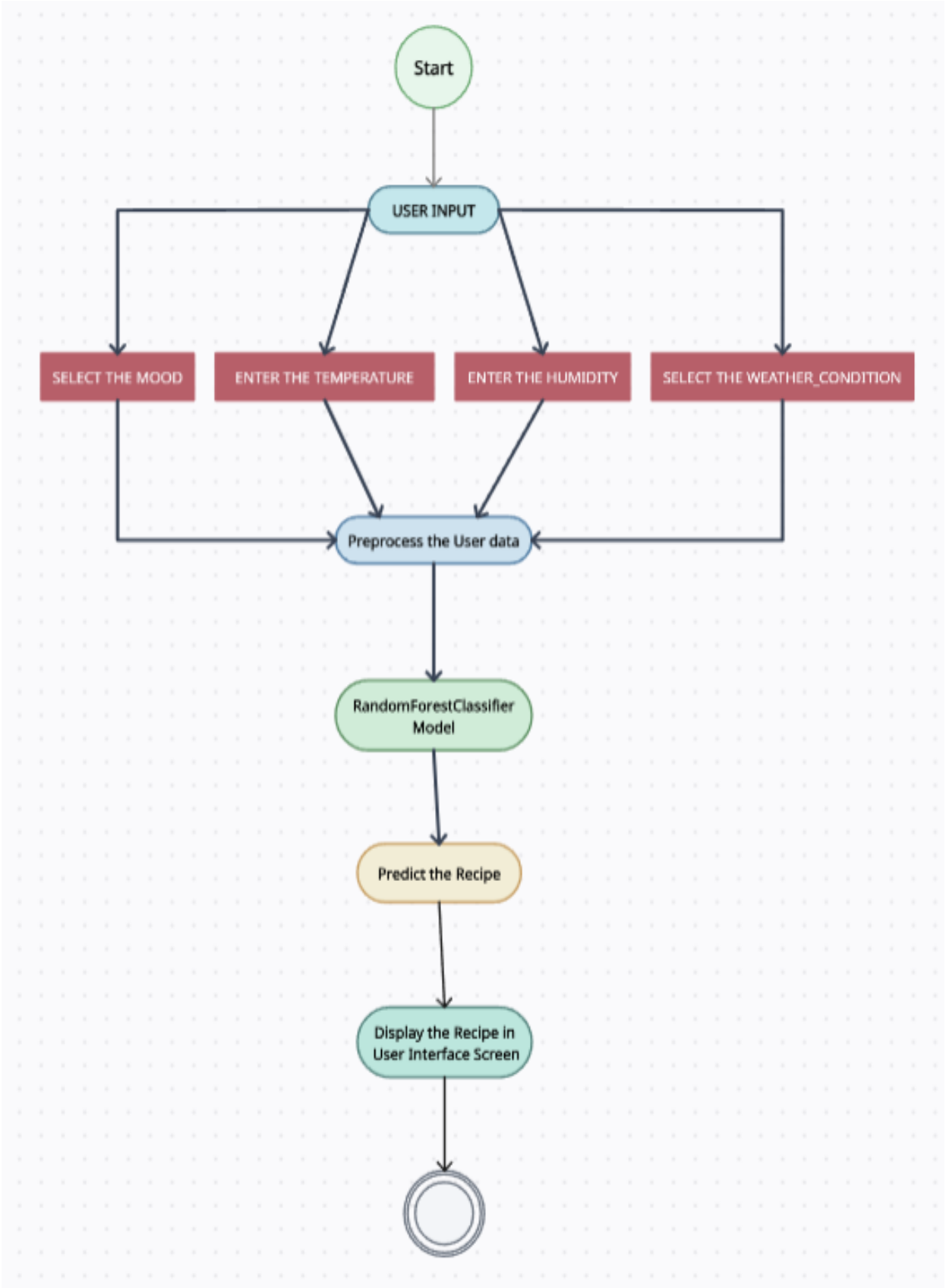
The core functionality involves preprocessing the input data, which includes categorical features (mood, weather condition) and numerical features (temperature, humidity). These inputs are transformed using One-Hot Encoding for categorical features and Standard Scaling for numerical features to ensure they are suitable for the model.

The machine learning model is trained on a labeled dataset containing various recipe categories and corresponding mood and weather conditions. The Random Forest Classifier is employed due to its robustness and effectiveness in handling both numerical and categorical data. Once trained, the model can predict the recipe category based on new inputs provided by users.

The web interface allows users to input their mood, temperature, humidity, and current weather condition. Upon submission, the application processes the input through the pre-trained model to generate a recipe category suggestion. The use of AJAX in the frontend ensures that the prediction result is displayed seamlessly on the same page without requiring a page reload.

This project demonstrates the integration of machine learning with web development to create an interactive and intelligent application. It provides a practical example of how data science techniques can be applied to enhance user experience in everyday activities such as cooking, making it more personalized and enjoyable.

Data Flow Diagram / Process Flow



Code:

Frontend Code (HTML)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Recipe Suggestion</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 50px;
            background-color: #b2dfdb;
        }
        form {
            max-width: 500px;
            margin: auto;
            padding: 10px;
            border: 1px solid #ccc;
            border-radius: 10px;
            background-color: #00796b;
        }
        div {
            margin-bottom: 15px;
            background-color: #afb42b;
        }
        header {
            background-color: #00796b;
            text-align: center;
            padding: 10px;
        }
        h1 {
            color: #e6ee9c;
        }
        p {
            color: #f0f4c3;
        }
        label {
            display: block;
            margin-bottom: 5px;
        }
        input, select {
            width: 100%;
```

```
            padding: 10px;
            box-sizing: border-box;
        }
        button {
            padding: 10px 15px;
            background-color: #004d40;
            border: none;
            color: #e0f7fa;
            cursor: pointer;
            border-radius: 5px;
        }
        button:hover {
            background-color: #827717;
        }
        #result {
            background-color: #00796b;
            color: #e6ee9c;
            text-align: center;
            padding: 10px;
            font-size: 30px;
        }
    </style>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script>
        $(document).ready(function() {
            $('form').on('submit', function(event) {
                event.preventDefault();

                $.ajax({
                    url: '/predict',
                    method: 'POST',
                    data: $('form').serialize(),
                    success: function(response) {
                        $('#result').text('Suggested Recipe is: ' +
response.recipe_category);
                    }
                });
            });
        });
    </script>
</head>
<body>
    <header>
        <h1>Flavour Forecast</h1>
        <p>This is a machine learning model to suggest recipes based
on user-provided mood and weather conditions.</p>
    </header>
    <br><br>
```

```
    <form action="/predict" method="post">
        <div>
            <label for="mood">Mood:</label>
            <select name="mood" id="mood">
                <option disabled="" selected="">Select Your Current
Mood</option>
                <option value="happy">Happy</option>
                <option value="sad">Sad</option>
                <option value="neutral">Neutral</option>
            </select>
        </div>
        <div>
            <label for="temperature">Temperature (°C):</label>
            <input type="text" name="temperature" id="temperature">
        </div>
        <div>
            <label for="humidity">Humidity (g/m3):</label>
            <input type="text" name="humidity" id="humidity">
        </div>
        <div>
            <label for="weather_condition">Weather
Condition:</label>
            <select name="weather_condition" id="weather_condition">
                <option disabled="" selected="">Select The Current
Weather Condition</option>
                <option value="sunny">Sunny</option>
                <option value="rainy">Rainy</option>
                <option value="cloudy">Cloudy</option>
            </select>
        </div>
        <button type="submit">Get Recipe</button>
    </form>
    <br><br>
    <div id="result"></div>

</body>
</html>
```

Backend Code (Python)

In [1]:
```
!pip install flask
```

Requirement already satisfied: flask in c:\users\chetan\anaconda3\lib\site-packages (2.2.5)
Requirement already satisfied: Werkzeug>=2.2.2 in c:\users\chetan\anacond3\lib\
site-packages (from flask) (2.2.3)
Requirement already satisfied: Jinja2>=3.0 in c:\users\chetan\anaconda3\lib\site-packages (from flask) (3.1.3)
Requirement already satisfied: its dangerous>=2.0 in c:\users\chetan\anaconda3\lib\ site-packages (from flask) (2.0.1)
Requirement already satisfied: click >=8.0 in c:\users\chetan\anaconda3\lib\ site-packages (from flask) (8.1.7)
Requirement already satisfied: colorama in c:\users\chetan\anaconda3\lib\ site-packages (from click >=8.0->flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\chetan\anaconda3\lib\ site-packages (from Jinja2>=3.0->flask) (2.1.3)

In [2]:
```
# Reading the recipe data from CSV file
import pandas as pd
df = pd.read_csv('recipe.csv')
```

In [3]:
```
#displaying the data
df
```

Out [3]:

	mood	temperature	humidity	weather_condition	recipe_category
0	happy	22	50	sunny	Biryani
1	sad	15	70	rainy	Soup
2	neutral	30	30	sunny	Icecream
3	happy	25	45	cloudy	Grill
4	sad	10	80	rainy	Tea
5	neutral	35	25	sunny	Smoothie
6	happy	20	55	sunny	Biryani
7	sad	5	85	rainy	Soup
8	neutral	40	20	sunny	Icecream
9	happy	36	30	cloudy	Maggie
10	sad	17	60	cloudy	Tea
11	sad	23	68	sunny	Smoothie
12	happy	24	58	rainy	Maggie
13	happy	40	50	rainy	Grill
14	neutral	30	40	cloudy	Tea
15	neutral	40	69	sunny	Biryani
16	sad	49	60	sunny	Icecream
17	sad	17	22	rainy	Grill
18	sad	23	34	cloudy	Smoothie
19	happy	40	67	sunny	Smoothie
20	happy	29	10	cloudy	Soup
21	sad	40	30	sunny	Biryani
22	neutral	20	40	rainy	Soup
23	neutral	50	40	sunny	Icecream
24	happy	10	10	cloudy	Tea
25	sad	40	20	cloudy	Soup
26	happy	30	40	sunny	Biryani
27	neutral	20	10	rainy	Tea
28	sad	40	89	sunny	Smoothie
29	happy	50	10	cloudy	Grill
30	neutral	20	10	rainy	Grill
31	happy	30	50	cloudy	Biryani
32	sad	20	50	cloudy	Maggie

33	sad	50	90	sunny	Icecream
34	happy	60	90	sunny	Biryani
35	neutral	80	70	sunny	Smoothie
36	sad	30	30	cloudy	Soup
37	sad	90	80	sunny	Icecream
38	happy	35	52	cloudy	Biryani
39	happy	20	59	cloudy	Grill

In [4]:

```
# Separate dependent and independent attributes
x = df.drop("recipe_category", axis=1)
x
```

Out [4]:

	mood	temperature	humidity	weather_condition
0	happy	22	50	sunny
1	sad	15	70	rainy
2	neutral	30	30	sunny
3	happy	25	45	cloudy
4	sad	10	80	rainy
5	neutral	35	25	sunny
6	happy	20	55	sunny
7	sad	5	85	rainy
8	neutral	40	20	sunny
9	happy	36	30	cloudy
10	sad	17	60	cloudy
11	sad	23	68	sunny
12	happy	24	58	rainy
13	happy	40	50	rainy
14	neutral	30	40	cloudy
15	neutral	40	69	sunny
16	sad	49	60	sunny
17	sad	17	22	rainy
18	sad	23	34	cloudy
19	happy	40	67	sunny
20	happy	29	10	cloudy
21	sad	40	30	sunny
22	neutral	20	40	rainy
23	neutral	50	40	sunny
24	happy	10	10	cloudy
25	sad	40	20	cloudy
26	happy	30	40	sunny
27	neutral	20	10	rainy
28	sad	40	89	sunny
29	happy	50	10	cloudy
30	neutral	20	10	rainy
31	happy	30	50	cloudy
32	sad	20	50	cloudy
33	sad	50	90	sunny
34	happy	60	90	sunny
35	neutral	80	70	sunny
36	sad	30	30	cloudy
37	sad	90	80	sunny
38	happy	35	52	cloudy
39	happy	20	59	cloudy

In [5]:
```
y = df["recipe_category"]
y
```

Out [5]:
```
1    Biryani
2    Soup
3    Icecream
4    Grill
5    Tea
6    Smoothie
7    Biryani
8    Soup
9    Icecream
10   Maggie
11   Tea
12   Smoothie
13   Maggie
14   Grill
15   Tea
16   Biryani
17   Icecream
18   Grill
19   Smoothie
20   Smoothie
21   Soup
22   Biryani
23   Soup
24   Icecream
25   Tea
26   Soup
27   Biryani
28   Tea
29   Smoothie
30   Grill
31   Grill
32   Biryani
33   Maggie
34   Icecream
35   Biryani
36   Smoothie
37   Soup
38   Icecream
39   Biryani
40   Grill
Name: recipe_category, dtype: object
```

In [6]:
```
# Preprocessing the data since we have target attribute as string type
 from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)
y
```

Out [6]:
```
array ([0, 5, 2, 1, 6, 4, 0, 5, 2, 3, 6, 4, 3, 1, 6, 0, 2, 1, 4, 4, 5, 0,
        5, 2, 6, 5, 0, 6, 4, 1, 1, 0, 3, 2, 0, 4, 5, 2, 0, 1])
```

In [7]:
```
#Differentiating the categorical and numerical data
categorical_features = ['mood', 'weather_condition']
numerical_features = ['temperature', 'humidity']
```

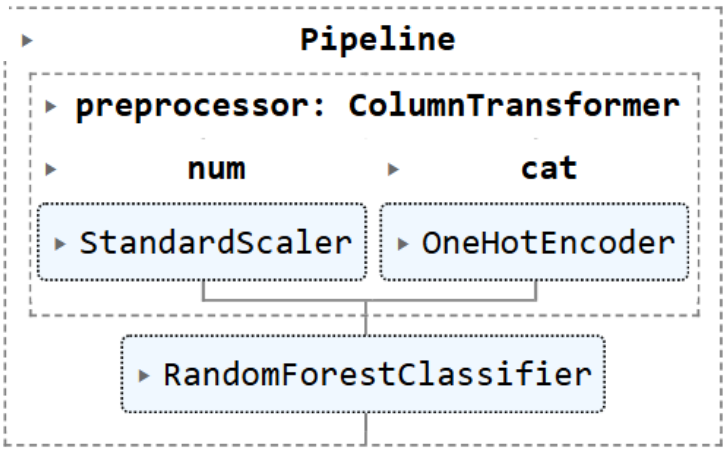
In [8]:
```
# Define column transformer with one-hot encoding for categorical features
#StandardScaler is for  transforming numerical_features
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_features),
        ('cat', OneHotEncoder(), categorical_features)
    ])
```

In [9]:
```
# Create and train the model pipeline
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
model = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', RandomForestClassifier(n_estimators=100, random_state=42))
])
```

In [10]:
```
# Split data into training and test sets
from sklearn.model_selection import train_test_split
train_X, test_X, train_y, test_y = train_test_split(x, y, test_size= 0.3, random_state= 42)
```

In [11]:
```
# Train the model
model.fit(train_X, train_y)
```

Out [11]:



'

*

In [11]:
```
# Prediction function
def suggest_recipe(mood, temperature, humidity, weather_condition):
    user_data = pd.DataFrame({
        'mood': [mood],
        'temperature': [temperature],
        'humidity': [humidity],
        'weather_condition': [weather_condition]
    })
    predicted_category = model.predict(user_data)[ 0]
    recipe_category = label_encoder.inverse_transform([predicted_category])[ 0]
    return recipe_category
```

In [12]:
```
#removing all the warnings
import warnings
warnings.filterwarnings("ignore")
```

In [13]:
```
from flask import Flask, request, jsonify, render_template
from threading import  Thread
```

In [14]:
```
app = Flask(__name__)

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

@app.route('/predict', methods=['POST'])
def predict():
    mood = request.form.get('mood')
    temperature = float(request.form.get('temperature'))
    humidity = float(request.form.get('humidity'))
    weather_condition = request.form.get('weather_condition')

    # Predict the recipe category
    recipe_category = suggest_recipe(mood, temperature, humidity, weather_condition)
    return jsonify({'recipe_category': recipe_category})

def run_app():
    app.run()

# Run Flask app in a separate thread to prevent blocking
Thread(target=run_app).start()
```

*Serving Flask app '__main__'
*Debug mode: off

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

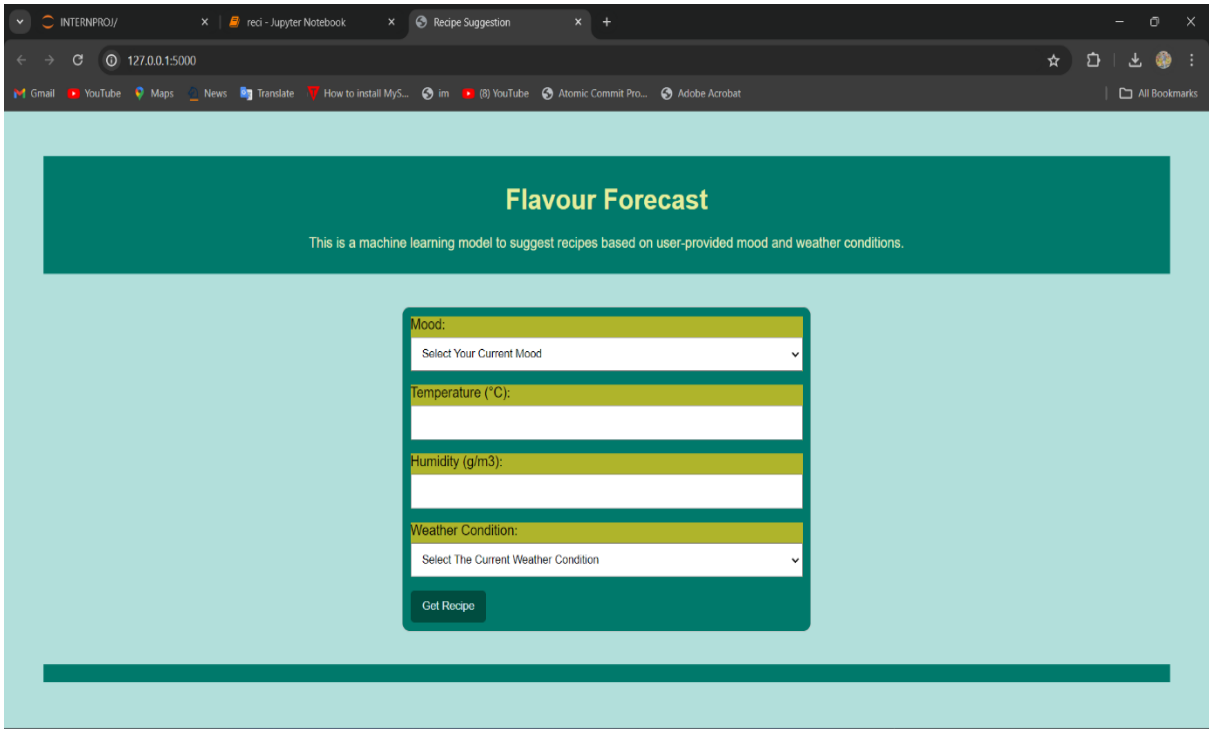
Input / Output with Datasets & Supported Screenshots

Input Dataset:

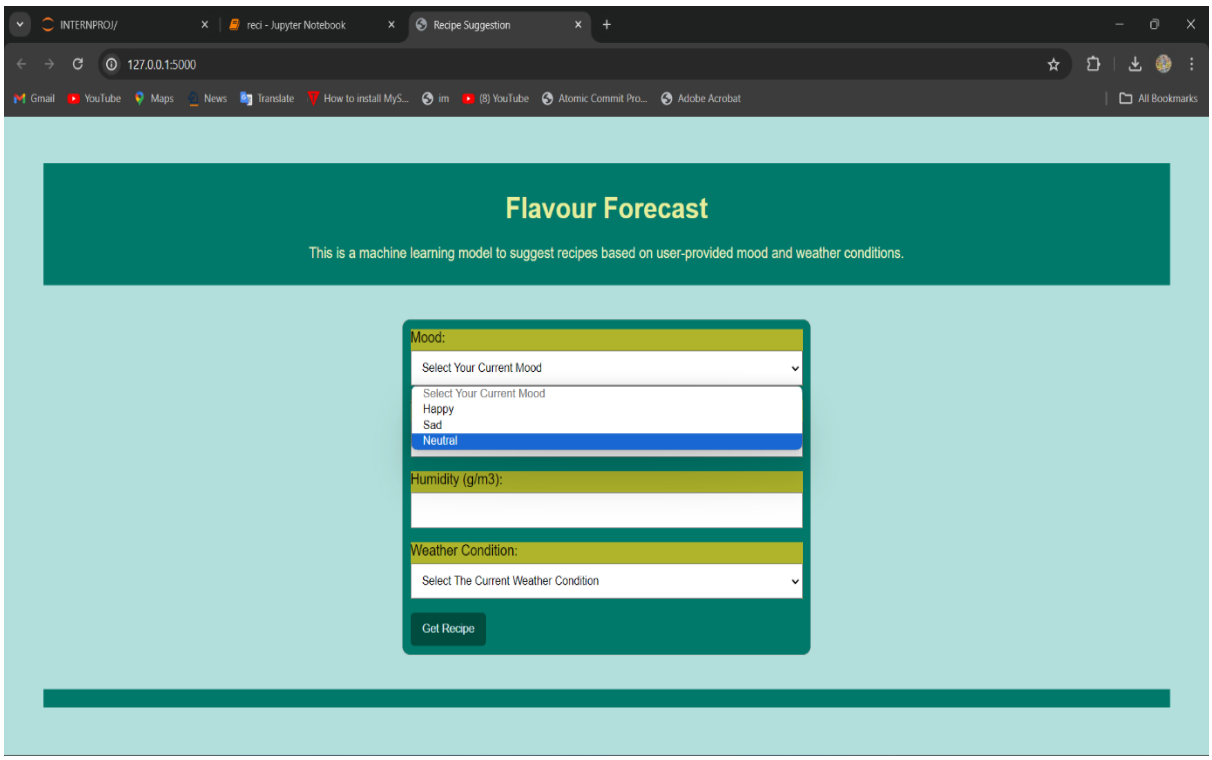
mood	temperature	humidity	weather_condition	recipe_category
happy	22	50	sunny	Biryani
sad	15	70	rainy	Soup
neutral	30	30	sunny	Icecream
happy	25	45	cloudy	Grill
sad	10	80	rainy	Tea
neutral	35	25	sunny	Smoothie
happy	20	55	sunny	Biryani
sad	5	85	rainy	Soup
neutral	40	20	sunny	Icecream
happy	36	30	cloudy	Maggie
sad	17	60	cloudy	Tea
sad	23	68	sunny	Smoothie
happy	24	58	rainy	Maggie
happy	40	50	rainy	Grill
neutral	30	40	cloudy	Tea
neutral	40	69	sunny	Biryani
sad	49	60	sunny	Icecream
sad	17	22	rainy	Grill
sad	23	34	cloudy	Smoothie
happy	40	67	sunny	Smoothie
happy	29	10	cloudy	Soup
sad	40	30	sunny	Biryani
neutral	20	40	rainy	Soup
neutral	50	40	sunny	Icecream
happy	10	10	cloudy	Tea
sad	40	20	cloudy	Soup
happy	30	40	sunny	Biryani
neutral	20	10	rainy	Tea
sad	40	89	sunny	Smoothie
happy	50	10	cloudy	Grill
neutral	20	10	rainy	Grill
happy	30	50	cloudy	Biryani
sad	20	50	cloudy	Maggie
sad	50	90	sunny	Icecream
happy	60	90	sunny	Biryani
neutral	80	70	sunny	Smoothie
sad	30	30	cloudy	Soup
sad	90	80	sunny	Icecream
happy	35	52	cloudy	Biryani
happy	20	59	cloudy	Grill

Output Screenshots:

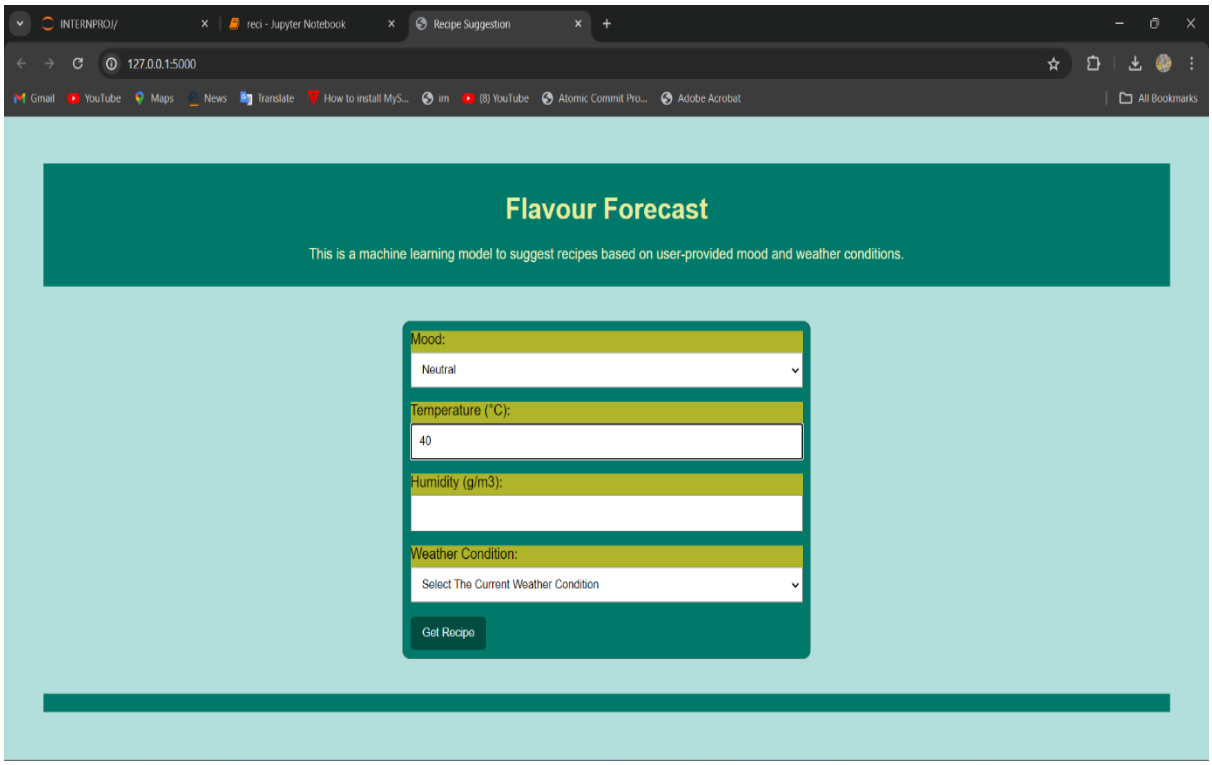
1. Normal index.html page:



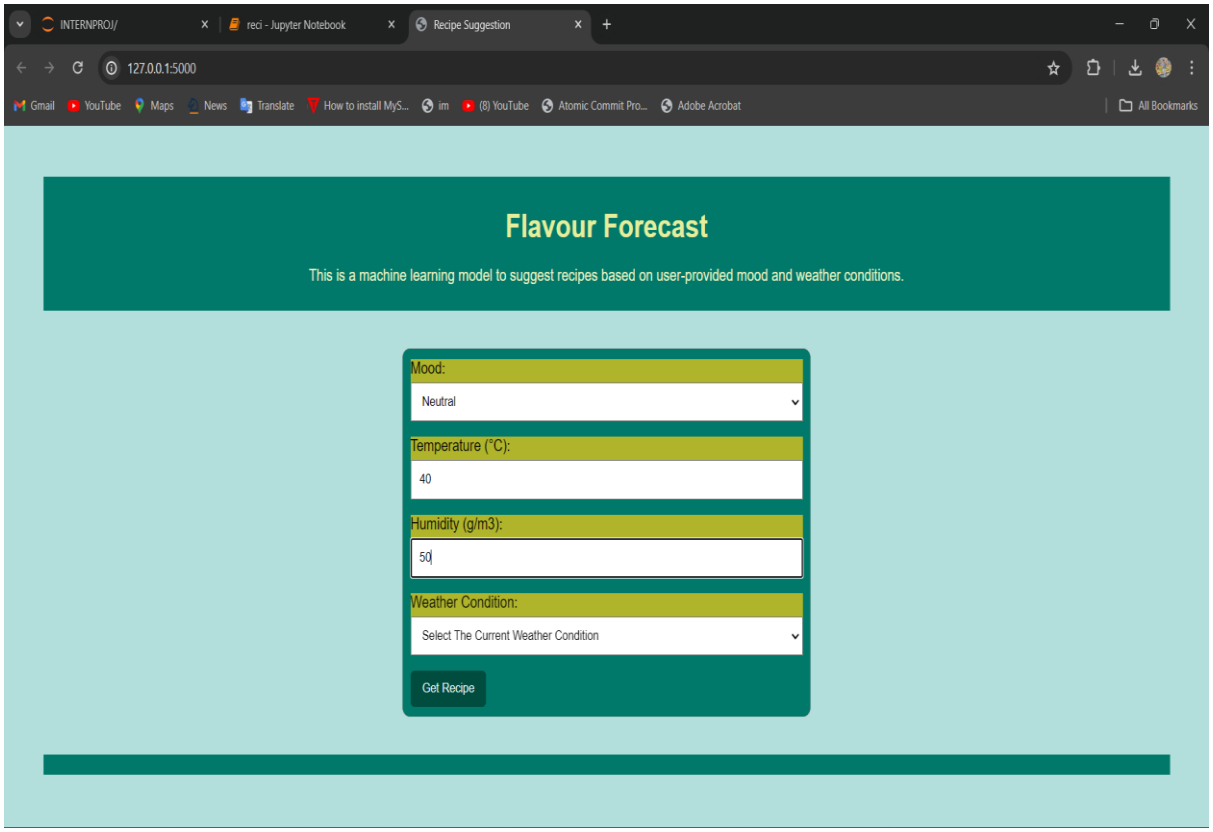
2. Selecting the current mood (happy/neutral/sad):



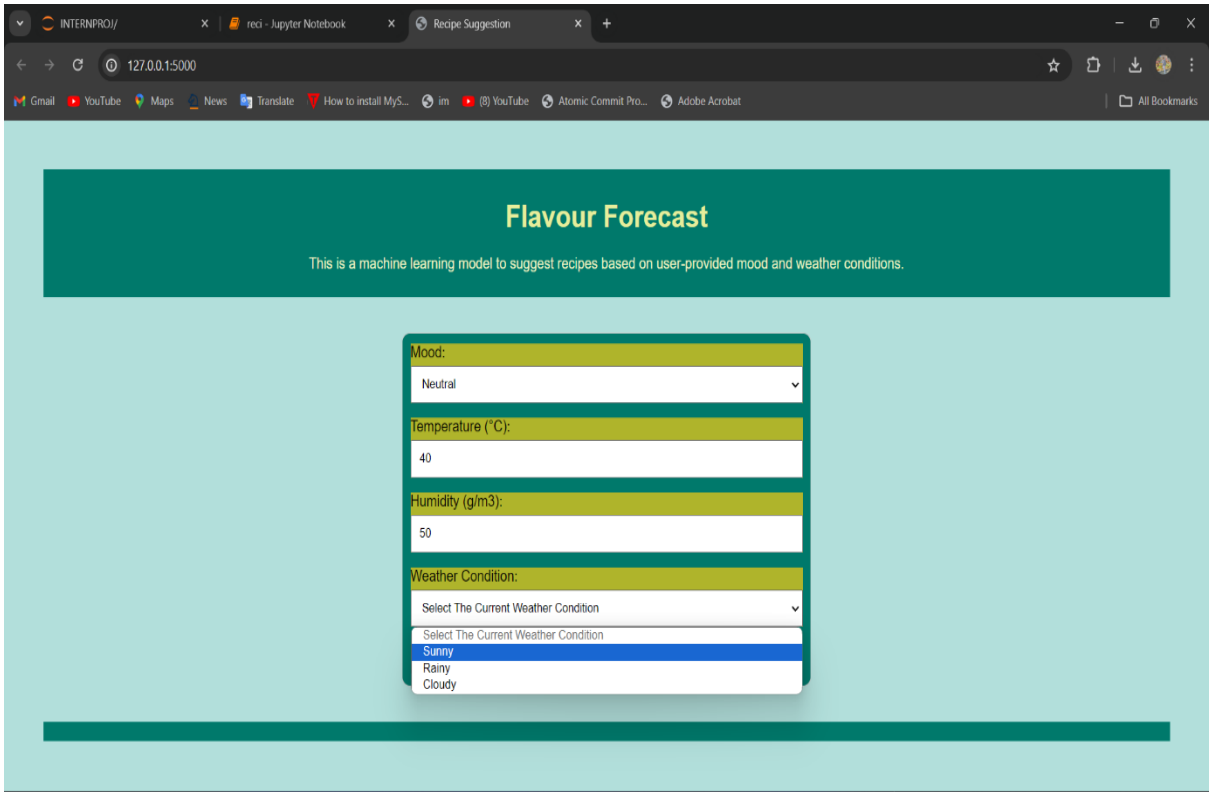
3. Entering the Temperature in °C:



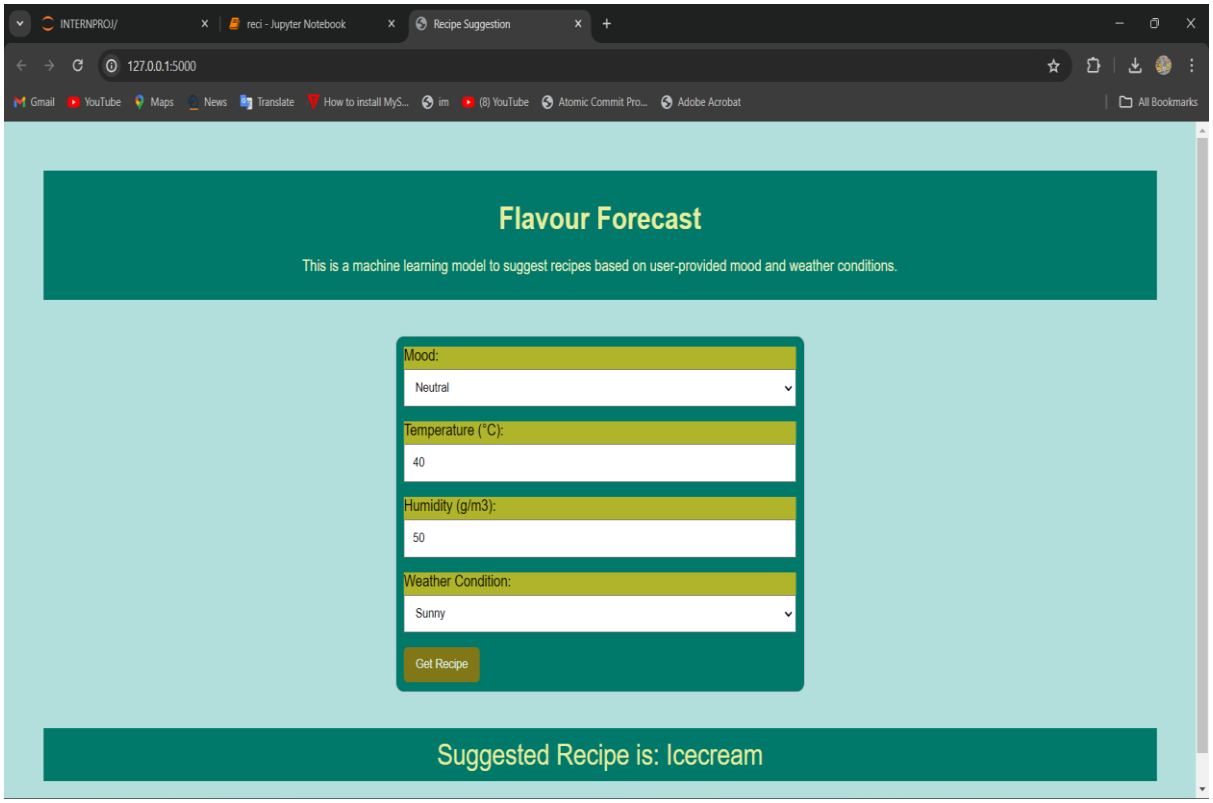
4. Entering the Humidity in g/m^3:



5. Selecting the weather condition (sunny/cloudy/rainy):



6. Clicking on the Get Recipe and displaying the Suggested Recipe:



References

GitHub link: https://github.com/priyanshrastogi/mood-based-food-recommender

GitHub link: https://github.com/diggyg97/Mood-Based-Recommendation-System

Student *Self Evaluation of the Short-Term Internship*

Please rate your performance in the following areas:

1)	Oral communication	1	2	3	4	5
2)	Written communication	1	2	3	4	5
3)	Initiative	1	2	3	4	5
4)	Interaction with staff	1	2	3	4	5
5)	Attitude	1	2	3	4	5
6)	Dependability	1	2	3	4	5
7)	Ability to learn	1	2	3	4	5
8)	Planning and organization	1	2	3	4	5
ss9)	Professionalism	1	2	3	4	5
10)	Creativity	1	2	3	4	5
11)	Quality of work	1	2	3	4	5
12)	Productivity	1	2	3	4	5
13)	Progress of learning	1	2	3	4	5
14)	Adaptability to organization's culture/policies	1	2	3	4	5
15)	OVERALL PERFORMANCE	1	2	3	4	5

Rating Scale: 5 will be Best while 1 will be Worst

CHATLA CHETAN KUMAR
Signature of the Student

Annexure 1
Daily Activity Report

Week No: 08

Day & Date	Brief Description of Daily Activity	Learning Outcome	Person In-Charge
Day 1	Foundations of Python	Learned the Introduction of Python Programming	Ms. Mallika Srivastava
Day 2	Installation of Python	Learned How to install Python	Ms. Mallika Srivastava
Day 3	Variables and keywords in python	Learned the basics of Python Programming	Ms. Mallika Srivastava
Day 4	First Python Program (print function & comments)	Printed the hello world! program	Ms. Mallika Srivastava
Day 5	Data Types (number Data Types)	Learned about the numeric Data Types	Ms. Mallika Srivastava
Day 6	Data Types (string Data Types)	Learned about the String Data Types	Ms. Mallika Srivastava
Day 7	Introduction to the Data Structures	Introduction to the various Data Structures in Python Programming	Ms. Mallika Srivastava
Day 8	List Data Structure	Learned working with Lists	Ms. Mallika Srivastava
Day 9	List Methods	Understood the various functions used in Lists	Ms. Mallika Srivastava
Day 10	Tuple Data Structure	Learned working with Tuples	Ms. Mallika Srivastava
Day 11	Dictionary Data Structure	Learned working with Dictionaries	Ms. Mallika Srivastava
Day 12	Set Data Structure	Learned working with Sets	Ms. Mallika Srivastava
Day 13	Boolean Data Structure	Learned working with Boolean data	Ms. Mallika

			Srivastava
Day 14	User Input & Type Casting	Understood how to take input from user and concept of type casting	Ms. Mallika Srivastava
Day 15	Control Statements	Understood the syntax for the control statements in Python Programming	Ms. Mallika Srivastava
Day 16	Quiz game and Introduction to loops	Developed a Quiz game and introduced to the Looping	Ms. Mallika Srivastava
Day 17	Loop in Python	Understood the for and while loop	Ms. Mallika Srivastava
Day 18	File Handling in Python	Learned about File handlings in Python	Ms. Mallika Srivastava
Day 19	Functions in Python	Learned about user-defined and pre-defined functions	Ms. Mallika Srivastava
Day 20	Packages and Modules	Introduced to the various packages and modules in Python	Ms. Mallika Srivastava
Day 21	Exception Handling	Learned about Exception Handling Techniques	Ms. Mallika Srivastava
Day 22	OOPS Concepts	Understood the OOPS in Python	Ms. Mallika Srivastava
Day 23	Installation of Jupyter Notebook	Learned to handle the Jupyter Notebook	Ms. Mallika Srivastava
Day 24	NumPy	Learned how to create NumPy arrays	Ms. Mallika Srivastava
Day 25	Pandas	Learned working with data frames using pandas.	Ms. Mallika Srivastava
Day 26	Matplotlib	Learned data visualization using Matplotlib	Ms. Mallika Srivastava
Day 27	Introduction to Machine Learning	Understood the correct meaning of Machine Learning	Ms. Mallika Srivastava
Day 28	Linear Regression Model	Learned about Linear Regression Model	Ms. Mallika Srivastava

Day 29	Types of Machine Learning Model	Understood about the Supervised, Unsupervised and Reinforcement models	Ms. Mallika Srivastava
Day 30	Create a Linear Regression Model	Developed Area Price Prediction Project	Ms. Mallika Srivastava
Day 31	Create Multi Regression Model	Developed a project using Iris dataset	Ms. Mallika Srivastava
Day 32	Create a Binary Classification Model	Developed a project using Titanic Survival dataset	Ms. Mallika Srivastava
Day 33	Support Vector Machine	Learned about SVM	Ms. Mallika Srivastava
Day 34	Create Image Classification	Developed a project of image classification	Ms. Mallika Srivastava
Day 35	Model Deployment	Created the user interfaces for the above-mentioned projects	Ms. Mallika Srivastava
Day 36	Choosing the Project for Internship	Selected a project named Flavor Forecast	Ms. Mallika Srivastava
Day 37	Dataset Creation and Model Demonstration	Created dataset manually and Developed the RandomForestClassifier model to predict the recipe	Ms. Mallika Srivastava
Day 38	Model Deployment	Created the user interface using Flask library and HTML	Ms. Mallika Srivastava
Day 39	Project Report	Created the Project Report	Ms. Mallika Srivastava
Day 40	Project report Submission	Submission of the project report	Ms. Mallika Srivastava

Annexure 2
Weekly Progress Report

Week No: 08

Week(s)	Summary of Weekly Activity
Week 1	Foundations of Python, Installation of Python, Variables and keywords in python, First Python Program (print function & comments), Data Types (number Data Types).
Week 2	Data Types (string Data Types), Introduction to the Data Structures, List Data Structure, List Methods, Tuple Data Structure.
Week 3	Dictionary Data Structure, Set Data Structure, Boolean Data Structure, User Input & Type Casting, Control Statements.
Week 4	Quiz game and Introduction to loops, Loop in Python, File Handling in Python, Functions in Python, Packages and Modules.
Week 5	Exception Handling, OOPS Concepts, Installation of Jupyter Notebook, NumPy, Pandas.
Week 6	Matplotlib, Introduction to Machine Learning, Linear Regression Model, Types of Machine Learning Model, Create a Linear Regression Model.
Week 7	Create Multi Regression Model, create a Binary Classification Model, Support Vector Machine, Create Image Classification, Model Deployment.
Week 8	Choosing the Project for Internship, Dataset Creation and Model Demonstration, Model Deployment, Project Report, Project Report Submission.