**Step 1: Set Up Your Development Environment**

**✅ 1.1 Install Python (3.10+ recommended)**

* 📥 Download from: <https://www.python.org/downloads/>
* During installation, **check the box** that says **"Add Python to PATH"**.

**✅ 1.2 Install Visual Studio Code (VS Code)**

* 📥 Download from: https://code.visualstudio.com/

**✅ 1.3 Install Git (for version control)**

* 📥 Download from: <https://git-scm.com/downloads>

**🗂️ Step 2: Create a Project Folder**

Create a folder anywhere on your computer:

CopyEdit

post\_harvest\_predictor/

**📦 Step 3: Create a Virtual Environment**

Open a terminal or command prompt:

bash

CopyEdit

cd path/to/post\_harvest\_predictor

python -m venv venv

Activate the environment:

* On **Windows**:

CopyEdit

venv\Scripts\activate

* On **Mac/Linux**:

bash

CopyEdit

source venv/bin/activate

**🧩 Step 4: Install Required Python Libraries**

bash

CopyEdit

pip install pandas numpy scikit-learn matplotlib seaborn plotly streamlit joblib requests

(You can also save this in a requirements.txt file.)

**📁 Step 5: Prepare Your Dataset**

You can either:

* **Create a simulated dataset** using Python, or
* Use open datasets such as:
  + FAO: <https://www.fao.org/statistics/en/>
  + Kaggle: https://www.kaggle.com/ (search for “fruit spoilage” or “supply chain”)

Create a sample CSV like:

csv

CopyEdit

temperature,humidity,transport\_time,distance,shelf\_life\_remaining

25,80,5,120,7

30,90,6,100,4

**🧠 Step 6: Build the Machine Learning Model (Random Forest)**

Create a file model\_train.py:

python

CopyEdit

import pandas as pd

from sklearn.ensemble import RandomForestRegressor

from sklearn.model\_selection import train\_test\_split

from joblib import dump

# Load your dataset

data = pd.read\_csv('data.csv') # change to your actual file

X = data[['temperature', 'humidity', 'transport\_time', 'distance']]

y = data['shelf\_life\_remaining']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

model = RandomForestRegressor()

model.fit(X\_train, y\_train)

# Save the trained model

dump(model, 'shelf\_life\_model.joblib')

Then run:

bash

CopyEdit

python model\_train.py

**🌐 Step 7: Create the Streamlit Dashboard**

Create a new file called app.py:

python

CopyEdit

import streamlit as st

import numpy as np

import pandas as pd

from joblib import load

# Load the model

model = load('shelf\_life\_model.joblib')

st.title("Post-Harvest Shelf Life Predictor")

st.write("Enter current environmental and transport conditions:")

temperature = st.slider("Temperature (°C)", 10, 40, 25)

humidity = st.slider("Humidity (%)", 30, 100, 70)

transport\_time = st.number\_input("Transport Time (hours)", min\_value=0.0, step=0.5)

distance = st.number\_input("Distance (km)", min\_value=0.0, step=1.0)

if st.button("Predict Shelf Life"):

input\_data = np.array([[temperature, humidity, transport\_time, distance]])

prediction = model.predict(input\_data)[0]

st.success(f"Estimated Shelf Life Remaining: {prediction:.2f} days")

Run the dashboard:

bash

CopyEdit

streamlit run app.py

**🔗 Step 8: Optional - Integrate APIs**

**OpenWeatherMap (to fetch real-time weather)**

* Sign up: https://openweathermap.org/api
* Use requests to call:

python

CopyEdit

import requests

API\_KEY = 'your\_api\_key'

city = 'Colombo'

url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={API\_KEY}&units=metric"

res = requests.get(url).json()

temp = res['main']['temp']

humidity = res['main']['humidity']

**📤 Step 9: Export Prediction Results**

You can allow exporting predictions as CSV:

Add this to app.py:

python

CopyEdit

import pandas as pd

if st.button("Download Report"):

report\_df = pd.DataFrame([{

"Temperature": temperature,

"Humidity": humidity,

"Transport Time": transport\_time,

"Distance": distance,

"Predicted Shelf Life": prediction

}])

csv = report\_df.to\_csv(index=False).encode('utf-8')

st.download\_button("Download as CSV", csv, "shelf\_life\_report.csv", "text/csv")

**📄 Step 10: Final Documentation and Deliverables**

Include:

1. **Project Report** – Background, problem, method, results.
2. **Presentation Slides** – Overview, screenshots, conclusions.
3. **README.md** – How to run your system.
4. **GitHub Repository** (Optional but recommended):
   * Create a repo at <https://github.com/>
   * Push your code there using Git.

**✅ Checklist of Final Deliverables**

| **Item** | **Status** |
| --- | --- |
| Predictive ML Model | ✅ |
| Streamlit Dashboard | ✅ |
| Data Integration (optional) | 🔄 |
| Exportable Prediction Results | ✅ |
| Report & Presentation | 🔄 |
| Technical Documentation | 🔄 |

Would you like me to generate sample code files or a GitHub repository structure? I can also help simulate a dataset or set up weather API integration for you.

Ask ChatGPT