Tempest FWI Predictor – A Machine Learning Model to Predict Fire Weather Index

Milestone 3 – Week 4 – Module 6 Learnings Documentation

Submitted by: Shaik Mohammad Irfan

Date: 14 December 2025

Week 4 Learnings Summary

In Week 4, I successfully deployed the trained Ridge Regression model as a fully functional web application using Flask. This module focused on transforming the machine learning model into a user-friendly, interactive tool that allows real-time prediction of the Fire Weather Index (FWI) based on weather and fuel conditions. The deployment ensures that the model can be used by non-technical users such as forest officials or fire management teams.

1. Flask Web Framework Overview

- Flask is a lightweight Python web framework ideal for deploying machine learning models.

- It allows creation of routes, handling of HTTP requests (GET/POST), and rendering of HTML templates.

- Key components used:

  - app.py → main application file

  - templates/ folder → HTML pages

  - Static assets (optional CSS for styling)

2. Application Structure Developed

- app.py

  - Loads the trained Ridge model (ridge.pkl) and StandardScaler (scaler.pkl) at startup

  - Defines two routes:

    - / → displays the input form

    - /predict (POST) → collects user input, preprocesses, predicts FWI, and shows result

  - Uses exact feature order as during training to ensure consistency

- templates/index.html
  - Clean, responsive input form with all 10 features:
    Region, Temperature, RH, Ws, Rain, FFMC, DMC, DC, ISI, BUI
  - Dropdown for Region (Bejaia = 0, Sidi-Bel Abbes = 1)
  - Default realistic values pre-filled for easy testing

- templates/home.html
  - Displays predicted FWI in large, bold format
  - Shows interpreted fire danger level:
    - < 1 → Very Low
    - < 5 → Low–Moderate
    - < 12 → High
    - < 25 → Very High
    - ≥ 25 → Extreme
  - Includes “New Prediction” button to return to form

3. Deployment Process in Google Colab

- Project folder created in Google Drive: FWI Predictor
- Subfolders: templates/ and static/ (for future CSS)
- Used %%writefile to create all files directly in Colab
- Integrated ngrok to expose local Flask server (port 5000) to the internet
- Generated secure public URL (https://xxxx.ngrok.io) for live demonstration
- No external server required – fully runnable within Colab environment

4. Key Technical Implementation Details

- Preprocessing Consistency
  - User inputs collected as float values
  - Converted to pandas DataFrame with correct column names
  - Applied the same saved StandardScaler before prediction
- Error Handling
  - Try-except block to catch invalid inputs

- User Experience
  - Professional styling with colors, shadows, and centered layout
  - Clear labels and intuitive flow

5. Testing & Validation

- Tested with multiple realistic inputs covering all fire danger categories
- Verified predictions match expected values from training data
- Confirmed model and scaler load correctly without errors

Key Takeaways Applied to Tempest FWI Predictor

- Model deployment bridges the gap between development and real-world use
- Saving both model and scaler separately is critical for correct inference
- Flask + HTML templates provide simple yet powerful web interface
- ngrok enables instant public sharing of Colab-hosted apps — perfect for project demos
- End-to-end pipeline now complete: Data → EDA → Model → Deployment

Module 6 Deliverables Completed:

- Fully functional Flask web application
- Input form with all required features
- Result page with predicted FWI and danger level interpretation
- Model and scaler properly loaded and used
- Live public URL generated via ngrok for demonstration
- Clean, professional UI ready for stakeholder use

Thank you!