Tempest Fire Weather Index (FWI) Predictor

Project Report

Infosys Springboard Virtual Internship

Name: Mishti Kalra

Table of Contents

1 Dataset Description and Preprocessing 2

1.1 Dataset Overview 2

1.2 Preprocessing Steps 3

2 Dataset Statistics 5

2.1 Descriptive Statistics 5

2.2 Histograms 7

2.3 Correlation Analysis 7

3 Ridge Regression Model 10

3.1 Model Description 10

3.2 Hyperparameter Tuning 11

3.3 MSE and MAE vs. Alpha 11

4 Flask Web Application 14

SECTION 1: Forest Fire Dataset

1.Dataset Overview

The Algerian Forest Fires Dataset contains meteorological and fire weather index parameters collected from two regions of Algeria Bejaia and Sidi Bel-Abbe for June to September 2012. It consists of 244 entries with 14 columns.

The dataset is designed to study and predict forest fire behavior, particularly the Fire Weather Index (FWI), which indicates the potential intensity of forest fires.

2. Description of Features (Attributes)

The dataset contains several features used to predict fire occurrences based on weather conditions and moisture levels.

- **Date Components** : day, month, year: The date of the observation.
- **Meteorological (Weather) Features**:
    - **Temperature**: Maximum daily temperature in degrees Celsius (°C).
    - **RH (Relative Humidity)**: The percentage of humidity in the air (%).
    - **Ws (Wind Speed)**: The speed of the wind in km/h.
    - **Rain**: Total daily rainfall measured in mm.
- **FWI (Fire Weather Index) System Indices**:
    - **FFMC (Fine Fuel Moisture Code)**: Indicates the moisture content of surface litter; it is a key indicator of how easily a fire might start.
    - **DMC (Duff Moisture Code)**: Represents the moisture content of shallow organic layers (duff).
    - **ISI (Initial Spread Index)**: Combines wind speed and FFMC to estimate the rate of fire spread.
    - **FWI (Fire Weather Index)**: A composite index representing the overall fire intensity.
- **Target/Added Features**:
    - **Region**: A binary indicator added during preprocessing (0 for Bejaia, 1 for Sidi Bel-Abbes).

Feature	Range
Region	0 – 1
Temperature (°C)	22 – 42
RH (%)	21 – 90
Ws (km/h)	6 – 29
Rain (mm)	0.0 – 16.8
FFMC	28.6 – 92.5
DMC	1.1 – 65.9
ISI	0.0 – 18.5
FWI	0.0 – 31.1

3. Preprocessing Steps

Step 1: Region Categorization

The code used df.loc[:122, "Region"] = 0 for the first region (Bejaia) and df.loc[122:,"Region"] = 1 for the second (Sidi Bel-Abbes) to create a searchable feature for the model.

Step 2: Handling Missing and Erroneous Data

- **Null Values**: The code checked for missing values and removed them using dataset.dropna().
- **Header Row Removal**: During the merge of the two regions, a secondary header row (containing strings like "day", "month", etc.) appeared in the middle of the data (at index 122). This row was explicitly dropped to prevent errors during mathematical operations.

Step 3: Cleaning Column Names

The raw dataset had inconsistent naming with trailing/leading spaces (e.g., ' RH' and 'Rain '). These were standardized using:

Python

```
dataset.columns = dataset.columns.str.strip()
```

Step 4: Data Type Conversion

The raw data was initially loaded as objects (strings). To make it useful for analysis:

- **Integer Conversion**: Features like day, month, year, Temperature, RH, and Ws were converted to int.
- **Float Conversion**: Numerical indices that require precision, such as Rain, FFMC, DMC, ISI, and FWI, were converted to float.

Step 5: Saving the Cleaned Dataset

After the transformations, the dataset was reset and saved as Algerian_forest_fires_dataset_CLEANED.csv, which is now ready for exploratory data analysis (EDA) and machine learning modeling.

4. Pseudocode for Data Preprocessing (Data Preprocessing.ipynb)

BEGIN

Load required libraries (pandas)

Load forest fire dataset CSV file

Display dataset shape and column information

Remove missing or null values from dataset

Clean column names for consistency

Encode categorical column "Region" into numerical format

Select input features:
Region, Temperature, RH, Ws, Rain, FFMC, DMC, ISI

Select target variable:
FWI
END

SECTION 2: Dataset Statistics

1.Descriptive Statistics

The dataset's key statistics are summarized below (from pandas describe()):

Feature	Count	Mean	Std Dev	Min	25%	50%	75%	Max
Temperature	244	32.17	3.63	22.0	30.00	32.00	35.00	42.0
Relative Humidity (RH)	244	61.94	14.88	21.0	52.00	63.00	73.25	90.0
Wind Speed (Ws)	244	15.50	2.81	6.0	14.00	15.00	17.00	29.0
Rainfall (mm)	244	0.76	2.00	0.0	0.00	0.00	0.50	16.8
FFMC	244	77.89	14.34	28.6	72.08	83.50	88.30	96.0
DMC	244	14.67	12.37	0.7	5.80	11.30	20.75	65.9
ISI	244	4.77	4.18	0.0	1.40	3.50	7.30	19.0
FWI	244	7.06	7.44	0.0	0.70	4.45	11.68	31.1
Region	244	0.50	0.50	0.0	0.00	0.50	1.00	1.0
Day	244	15.75	8.83	1.0	8.00	16.00	23.00	31.0
Month	244	7.50	1.11	6.0	7.00	7.50	8.00	9.0
Year	244	2012	0.00	2012	2012	2012	2012	2012

The dataset comprises 244 observations across two regions in Algeria during the fire-prone summer months of 2012.

Key descriptive statistics highlight the central tendencies, dispersion, and distribution shapes of the variables. From the pandas .describe() output and additional computations.

1. Central Tendencies and Extremes

- Meteorological Conditions: The average Temperature is approximately 32.17°C, with a maximum reaching 42°C. Relative Humidity (RH) averages 62%, while Wind Speed (Ws) remains consistent at an average of 15.5 km/h. Rainfall is extremely low, averaging only 0.76 mm/day, indicating predominantly dry conditions suitable for fire ignition.
- Fire Indices**:** The fire indices show moderate to high mean values: FFMC (Fine Fuel Moisture Code) averages 77.9, DMC averages 14.7, and ISI averages 4.77. The target FWI (Fire Weather Index) averages 7.06, but reaches an extreme peak of 31.1, indicating periods of critical fire danger.

2. Variability and Dispersion

- Seasonal Buildup**:** Notable variability is seen in DMC (std: 12.37) and FWI (std: 7.44), reflecting how fire risk builds up inconsistently over the season.
- Stability**:** Temperature shows relatively low variability (std: 3.63°C), suggesting a sustained heat profile throughout the recorded months.

3. Statistical Moments and Distribution Shapes

The dataset exhibits significant non-normality, which is crucial for choosing the right machine learning models:

- Skewness**:** * Rain (4.58): Strong positive skew. It is highly right-tailed because most days have zero rain, with rare high-rainfall events.
    - DMC (1.53), ISI (1.12), and FWI (1.14**):** These indices are moderately right-skewed, indicating a long tail toward higher fire risk values.
    - FFMC (-1.33): Left-skewed (negative skew), meaning the majority of observations are concentrated at the higher end of the scale (dry fuel).
    - Temperature (-0.20): Near-symmetric, though slightly left-skewed.
- Kurtosis**:**
    - Rain (25.94): Extremely high kurtosis (leptokurtic), indicating that the distribution has very heavy tails and a sharp peak at zero.
    - Ws (2.60) and DMC (2.49): Higher peaks than a normal distribution, suggesting frequent occurrences around the mean.

4. Preprocessing Justification

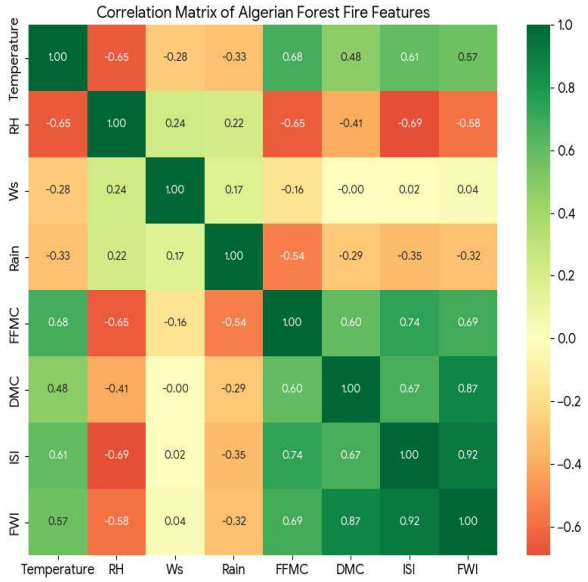
The presence of skewed distributions and varying scales across features (e.g., Rain vs. FFMC) justifies the following preprocessing steps:

- **Standardization:** Scaling features to a mean of 0 and variance of 1 is recommended for regularization-sensitive models like Ridge Regression.
- **Log-Transformations:** For highly skewed features like Rain, a log transformation could help normalize the distribution for linear-based modeling.
- **Robustness:** Since several fire indices (FWI, DMC) have long tails, models that are robust to outliers or those that do not assume normality are likely to perform better.

2.Histograms

Histograms of key variables show distributions: Temperature is bimodal (around 30-35°C), RH skewed right (50-80%), Ws moderate (10-20 km/h), Rain heavily skewed to 0, and fire indices like FFMC left-skewed (high values common), others right-skewed.

3.Correlation



Correlation Matrix Heatmap

The heatmap below illustrates the linear relationships between the variables. Positive correlations are shown in green, while negative correlations are shown in red.

1. Strongest Positive Relationships

- FWI vs. ISI (0.92)**:** This is the strongest correlation in the dataset. Since ISI (Initial Spread Index) represents the rate of fire spread, its near-perfect correlation with FWI confirms that spread speed is the dominant factor in determining overall fire intensity in this region.
- FWI vs. DMC (0.87): The high correlation with DMC (Duff Moisture Code) indicates that the moisture content of the shallow organic layers significantly impacts the potential for a severe forest fire.
- ISI vs. FFMC (0.74)**:** The FFMC (Fine Fuel Moisture Code), which tracks the moisture of surface litter, strongly influences the initial spread index. This shows that drier surface materials lead to faster-spreading fires.

2. Weather Drivers of Fire Risk

- Temperature**:** Has a strong positive correlation with FFMC (0.68) and FWI (0.57). High temperatures directly contribute to the drying of fuel, escalating fire risk.
- Relative Humidity (RH)**:** Shows a significant negative correlation with FWI (-0.58) and ISI (-0.69). As humidity drops, the air and forest fuels become drier, substantially increasing the risk of ignition and spread.
- Rainfall: Exhibits a moderate negative correlation with FWI (-0.32). While rain decreases fire risk, its influence is less than that of Temperature and RH during the summer season.

3. Weak or Indirect Drivers

- Wind Speed (Ws): Surprisingly, wind speed shows a very weak direct correlation with FWI (0.04) and DMC (-0.00). This suggests that wind's impact is not a linear predictor of intensity on its own but acts as a multiplier when fuels are already dry (captured within the ISI calculation).

4.Pseudocode

BEGIN

STEP 1: INITIALIZATION

Import data processing libraries:

pandas

numpy

Import visualization libraries:

matplotlib

seaborn

STEP 2: LOAD DATA

Load "Algerian_forest_fires_dataset_CLEANED.csv"

Store dataset in variable: data

STEP 3: ANALYZE TARGET VARIABLE (FWI)

Create a new figure

Generate a histogram of the 'FWI' column using 30 bins

Set title as "Histogram of FWI"

Label x-axis as "FWI Values"

Label y-axis as "Frequency"

Display the plot

STEP 4: GENERATE FEATURE HISTOGRAMS

Extract all column names from data into feature_names

FOR each column in feature_names DO

  IF column is NOT equal to "FWI" THEN

    Create a new figure

    Generate histogram for the current column

    Set transparency (alpha) to 0.5

    Set title as "Histogram Comparison: [column] vs FWI"

    Label x-axis as "Value"

    Label y-axis as "Frequency"

    Display legend

    Display the plot

  END IF

END FOR

STEP 5: CORRELATION ANALYSIS

Calculate correlation matrix for all numeric columns in data

Create a new figure with size (10, 6)

Generate a heatmap using the correlation matrix:

Enable annotations

Use "coolwarm" color map

Set title as "Correlation Heatmap"

Display the plot

END

SECTION 3: Ridge Regression

Model Description

It is a technique used to analyze multiple regression data that suffer from multicollinearity (when independent variables are highly correlated).

In standard Linear Regression, high multicollinearity can lead to very large coefficients, making the model extremely sensitive to small changes in the input data. Ridge Regression addresses this by adding a "penalty" to the size of the coefficients.

The Objective Function

The Ridge Regression model minimizes the Residual Sum of Squares (RSS) plus a penalty term proportional to the square of the magnitude of the coefficients (L2 regularization).

The function the model minimizes is:

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \alpha \sum_{j=1}^{p} \beta_j^2 \right\}$$

Where:

1.Mean Squared Term

- This is the standard loss used in Linear Regression. It measures the difference between the actual FWI and the FWI predicted by the model.

2.L2 Regularization Penalty: This is the "Ridge" part. It adds a penalty based on the square of the magnitude of the coefficients.

3.Alpha **:** This is the tuning parameter which controls how much weight is given to the penalty. A higher alpha shrinks coefficients more aggressively to prevent overfitting.

2 Hyperparameter Tuning

Alpha values tested: logspace from 10^{-3} and 10^{3} (100 values).

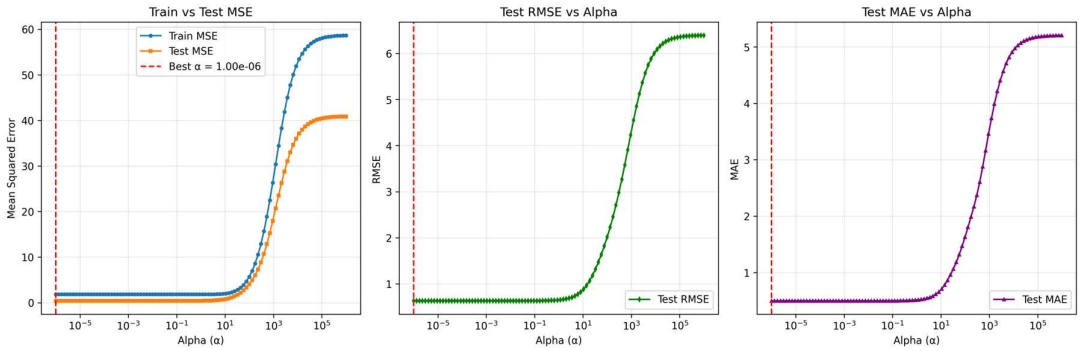
Metric	Symbol	Value
Optimal Alpha	244	0.000001
Mean Absolute Error	244	0.494
Mean Squared Error	244	0.3956
Root Mean Squared Error	244	06290

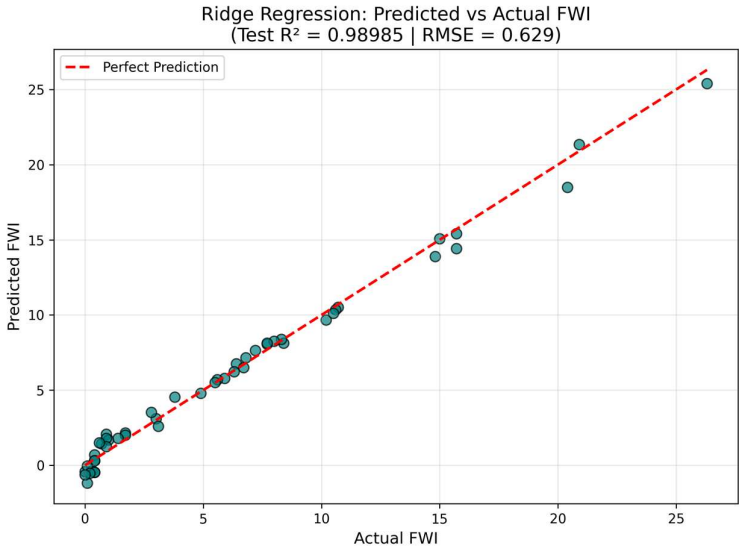
Analysis of Results

- Optimal Alpha**:** The extremely low alpha value (10-6) suggests that while regularization was applied to stabilize the coefficients, the data was already quite well-structured for the regression task.
- MSE & MAE**:** An MSE of 0.3956 and MAE of 0.4964 are very low considering the FWI values in the dataset range from 0 to 31. This indicates that, on average, the model's predictions are only off by about 0.5 units of FWI.

Analysis of MSE and MAE vs Alpha

- Mean Squared Error (MSE) vs Alpha**:** This plot shows the relationship between the squared error on the test set and the complexity parameter. As alpha increases (moving to the right), the model becomes more restricted (less complex). Initially, for very small values of alpha, the MSE remains low and stable. However, as alpha becomes too large the MSE begins to rise sharply because the model is "underfitting"—it is becoming too simple to capture the underlying patterns in the data.
- Mean Absolute Error (MAE) vs Alpha:The MAE plot follows a similar trend to the MSE. MAE represents the average magnitude of the errors in the predictions. The flat region at the start indicates that the model is performing consistently well with minimal regularization. The point where the error starts to climb rapidly represents the threshold where the penalty on the coefficient sizes becomes too high, causing the model to lose its predictive accuracy.





4.Pseudocode

BEGIN

// STEP 1: INITIALIZATION

IMPORT machine_learning_libraries (sklearn, pandas, numpy, pickle)

// STEP 2: DATA PREPARATION

LOAD "Algerian_forest_fires_dataset_CLEANED.csv" INTO 'df'

REMOVE columns ('day', 'month', 'year') from 'df' // Drop temporal features

SET target_variable 'y' = 'FWI' column

SET input_features 'X' = all columns in 'df' EXCEPT 'FWI'

// STEP 3: DATA SPLITTING & SCALING

SPLIT 'X' and 'y' into Training (80%) and Testing (20%) sets

CREATE 'scaler' object using StandardScaler

FIT 'scaler' on Training set AND TRANSFORM (Normalize) both Train and Test sets

SAVE 'scaler' to file "scaler.pkl" for future deployment

// STEP 4: HYPERPARAMETER (ALPHA) TUNING

GENERATE 100 values for 'alpha' ranging from 1e-6 to 1e6 (Logarithmic scale)

INITIALIZE empty lists to store MSE, RMSE, and MAE for both Train and Test set

FOR EACH alpha_value IN alphas:

INITIALIZE Ridge Model with current alpha

TRAIN Ridge Model on Scaled Training data

PREDICT 'y' for Training and Testing sets

CALCULATE MSE, RMSE, and MAE

APPEND results to respective lists

END FOR

// STEP 5: SELECTION OF OPTIMAL MODEL

IDENTIFY 'best_alpha' where Test MSE is at its minimum

// STEP 6: FINAL MODEL EVALUATION

CREATE 'final_model' using 'best_alpha'

TRAIN 'final_model' on Scaled Training data

GENERATE final predictions on Scaled Test data

CALCULATE Final Metrics:

- R^2 Score

- Mean Absolute Error (MAE)

- Root Mean Squared Error (RMSE)

// STEP 7: STORAGE

SAVE 'final_model' to file "ridge.pkl”

// STEP 8: VISUALIZATION

PLOT Train vs Test MSE across different Alpha values

PLOT Predicted vs Actual FWI values

DISPLAY results and feature coefficients

END

SECTION 4: Flask App

1. Deployment Architecture: From Backend to Flask API

Deployment is the process of taking trained machine learning model and making it available for users to interact with through a web interface.

The Workflow:

1.Serialization: In training script, the trained Ridge model and the StandardScaler were saved as binary files (ridge.pkl and scaler.pkl) using the **Pickle** library. This "freezes" the model so it can be loaded anywhere.

2.The Flask Backend (app.py): * It initializes a Flask server.

- It loads the saved .pkl files into memory.
- It defines "Routes": / to show the input form and /predict to process data.

3.The Frontend (index.html & home.html):

- index.html provides a user-friendly form to collect weather parameters.
- When the "Predict" button is clicked, it sends a POST request to the Flask API.

4.The Prediction Cycle: The backend receives the form data, transforms it using the scaler, runs it through the ridge model, and sends the result to home.html to be displayed.

2. Fire Weather Index (FWI) Risk Measures

The file contains specific logic to categorize the numeric FWI prediction into human-readable risk levels. These measures are crucial for public safety warnings.

FWI Range	Risk Level Category	UI Colour/Label
FWI > 60	Extreme Fire Risk	Dark Red
30 < FWI <= 60	Very High Risk	Red
15 < FWI <=30	High Risk	Orange

2. Pseudocode for Deployment Files

A. Backend Pseudocode (app.py)

```
BEGIN
IMPORT Flask, Pickle, and Numpy
// INITIALIZATION
INITIALIZE Flask application
LOAD "ridge.pkl" as 'model'
TRY:
LOAD "scaler.pkl" as 'scaler'
EXCEPT:
SET 'scaler' to NULL
// ROUTE: DISPLAY INPUT FORM
FUNCTION index():
RENDER "index.html"
// ROUTE: HANDLE PREDICTION
FUNCTION predict():
IF request is POST:
GET "Region, Temperature, RH, Ws, Rain, FFMC, DMC, ISI" from user form
CONVERT all inputs to float/int
CREATE 'input_vector' from input
IF 'scaler' exists:
TRANSFORM 'input_vector' using 'scaler'
EXECUTE model.predict on 'input_vector' to get 'predicted_fwi'
PASS 'predicted_fwi' and input values to "home.html"
RENDER "home.html"
END IF
END
```

B. Frontend Logic (index.html & home.html)

BEGIN

// INPUT FORM (index.html)

DISPLAY form with 8 input fields

ON BUTTON CLICK "Predict Fire Index":

SEND form data to backend server via POST request

// RESULT PAGE (home.html)

RECEIVE 'prediction' from backend

DISPLAY numeric FWI value

// RISK CLASSIFICATION LOGIC

IF prediction is greater than 60:

DISPLAY "Extreme Fire Risk" label

ELSE IF prediction is greater than 30:

DISPLAY "Very High Risk" label

ELSE IF prediction is greater than 15:

DISPLAY "High Risk" label

ELSE IF prediction is greater than 5:

DISPLAY "Moderate Risk" label

ELSE:

DISPLAY "Low Risk" label

DISPLAY table of values entered by the user for reference

END