Fire Weather Index Predictor

(A Machine Learning model to predict Fire Weather)

Infosys Springboard Virtual Internship Program

Submitted By

Potukuchi Sneha

Under the guidance of mentor Praveen

Project Statement

This project focuses on building a Fire Weather Index (FWI) Predictor that estimates wildfire risk using essential environmental and fire-danger features such as temperature, relative humidity, wind speed, rainfall, FFMC, DMC, DC, ISI, and BUI. The dataset is thoroughly cleaned, pre-processed, and analysed to ensure reliable inputs, followed by visual exploration to understand feature distributions and relationships. Machine learning techniques are then applied to learn how weather conditions influence fire danger levels, enabling accurate prediction of the FWI value. The system also incorporates regional information to study area-wise fire behaviour and improve prediction relevance. Overall, the model aims to support early detection, risk assessment, and informed decision-making for effective wildfire management.

Expected Outcome

• A predictive ML model trained using Ridge Regression to forecast FWI.

• A pre-processing pipeline using StandardScaler for normalisation.

• A Flask-based web app where users can input environmental values and get FWI predictions.

• A system that can help forest departments, emergency planners, and climate researchers make data-driven decisions.

Modules to be Implemented

• Data Collection

• Data Exploration (EDA) and Data Preprocessing

• Feature Engineering and Scaling

• Model Training using Ridge Regression

• Evaluation and Optimisation

• Deployment via Flask App

• Presentation and Documentation

System Requirements

Software:

- Python
- Python libraries (pandas, numpy, matplotlib, seaborn etc)
- Flask

Milestone 1

1. Module 1 (Data Collection)

The dataset was collected by exploring multiple online sources and selecting one that contained the essential environmental features required for FWI prediction, including Temperature, Relative Humidity, Wind Speed, Rain, FFMC, DMC, ISI, and Region. After loading the chosen dataset into a Pandas DataFrame, an initial inspection was carried out to understand its structure and quality. This included checking datatypes, identifying null values, reviewing memory usage, and generating statistical summaries to examine the distribution and range of numerical features. The dataset's shape and duplicate entries were also analysed to ensure completeness and reliability, providing a solid foundation for further preprocessing and modelling.

Conducted initial inspection to understand feature distributions and data quality.

1. Load the dataset

```
df =
pd.read_csv('C:\\Users\\DELL\\OneDrive\\Desktop\\FWI_Predictor\\FWI
Dataset.csv')
print("Loaded the dataset using pandas")
```

2. Verify Datatypes

```
Data Types of Each Column:
day            int64
month          int64
year           int64
Temperature    int64
 RH            int64
 Ws            int64
Rain           float64
FFMC           float64
DMC            float64
DC             object
ISI            float64
BUI            float64
FWI            object
Classes        object
Region         object
dtype: object
```

3. Basic Information of the Dataset

```
print("\nFirst 5 Rows of Dataset:")
print(df.head())
print("\nLast 5 Rows of Dataset:")
print(df.tail())
✓ 0.0s

First 5 Rows of Dataset:
   day  month  year  Temperature  RH  Ws  Rain  FFMC  DMC    DC  ISI  BUI  \
0    1      6  2012           29  57  18   0.0  65.7  3.4   7.6  1.3  3.4
1    2      6  2012           29  61  13   1.3  64.4  4.1   7.6  1.0  3.9
2    3      6  2012           26  82  22  13.1  47.1  2.5   7.1  0.3  2.7
3    4      6  2012           25  89  13   2.5  28.6  1.3   6.9  0.0  1.7
4    5      6  2012           27  77  16   0.0  64.8  3.0  14.2  1.2  3.9

   FWI    Classes    Region
0  0.5  not fire     Bejaia
1  0.4  not fire     Bejaia
2  0.1  not fire     Bejaia
3    0  not fire     Bejaia
4  0.5  not fire     Bejaia

Last 5 Rows of Dataset:
     day  month  year  Temperature  RH  Ws  Rain  FFMC   DMC    DC  ISI  \
239   26      9  2012           30  65  14   0.0  85.4  16.0  44.5  4.5
240   27      9  2012           28  87  15   4.4  41.1   6.5     8  0.1
241   28      9  2012           27  87  29   0.5  45.9   3.5   7.9  0.4
242   29      9  2012           24  54  18   0.1  79.7   4.3  15.2  1.7
243   30      9  2012           24  64  15   0.2  67.3   3.8  16.5  1.2

      BUI  FWI     Classes            Region
...
240   6.2    0   not fire     Sidi-Bel Abbes
241   3.4  0.2   not fire     Sidi-Bel Abbes
242   5.1  0.7   not fire     Sidi-Bel Abbes
243   4.8  0.5  not fire      Sidi-Bel Abbes
```

FIG 1: Information of Dataset

The first and last five rows of the dataset were displayed to gain an initial understanding of the data structure, feature values, and overall formatting. This helped verify that all columns loaded correctly, values are aligned, and the dataset is ready for further inspection and preprocessing.

4. Statistical Summary

```
#stats for numerical features
print("\nStatistical Summary of Numerical Features:")
print(df.describe())
```

✓ 0.3s

```
Statistical Summary of Numerical Features:
              day       month    year  Temperature          RH          Ws  \
count  244.000000  244.000000   244.0   244.000000  244.000000  244.000000
mean    15.754098    7.500000  2012.0    32.172131   61.938525   15.504098
std      8.825059    1.112961     0.0     3.633843   14.884200    2.810178
min      1.000000    6.000000  2012.0    22.000000   21.000000    6.000000
25%      8.000000    7.000000  2012.0    30.000000   52.000000   14.000000
50%     16.000000    7.500000  2012.0    32.000000   63.000000   15.000000
75%     23.000000    8.000000  2012.0    35.000000   73.250000   17.000000
max     31.000000    9.000000  2012.0    42.000000   90.000000   29.000000

             Rain        FFMC         DMC         ISI         BUI
count  244.000000  244.000000  244.000000  244.000000  244.000000
mean     0.760656   77.887705   14.673361    4.774180   16.664754
std      1.999406   14.337571   12.368039    4.175318   14.204824
min      0.000000   28.600000    0.700000    0.000000    1.100000
25%      0.000000   72.075000    5.800000    1.400000    6.000000
50%      0.000000   83.500000   11.300000    3.500000   12.250000
75%      0.500000   88.300000   20.750000    7.300000   22.525000
max     16.800000   96.000000   65.900000   19.000000   68.000000
```

FIG 2: Statistical summary of the Dataset

A statistical summary was generated to examine key numerical characteristics such as mean, median, minimum, maximum, and standard deviation for each feature. This helped assess data variability, detect potential outliers, and understand the overall distribution of numerical values in the dataset.

5. Duplicate Values

```
print("\nChecking for Duplicate Rows:")
print(df.duplicated().sum())
```

2. Module 2 (Data Exploration (EDA) and Data Preprocessing)

During the preprocessing stage, the dataset was first examined for missing or null values, and appropriate handling techniques were applied to ensure completeness. Outlier detection was then performed using boxplots and statistical thresholds to identify abnormal values that could affect model performance. To better understand feature behaviour, data distributions were visualised through histograms and density plots, while correlation matrices and scatterplots were used to explore relationships between variables. Categorical features, such as *Region,* were encoded using label encoding to make them suitable for machine learning algorithms. Finally, the cleaned and processed dataset was saved for use in building and evaluating predictive models.

1. Handle missing values

```
Checking for Missing Values:
day            0
month          0
year           0
Temperature    0
 RH            0
 Ws            0
Rain           0
FFMC           0
DMC            0
DC             0
ISI            0
BUI            0
FWI            0
Classes        1
Region         0
dtype: int64
Rows with missing values:
      day  month  year  Temperature   RH   Ws  Rain   FFMC   DMC      DC   ISI  \
165   14      7  2012           37   37   18    0.2  88.9  12.9  14.6 9  12.5

      BUI      FWI Classes            Region
165  10.4  fire          NaN  Sidi-Bel Abbes

Total number of rows with missing values:
1
```

FIG 3: Missing Values

A missing-value check was performed to identify incomplete records, revealing that only one row contained a null value in the *Classes* column. This validation helped ensure data completeness and supported appropriate handling before further preprocessing.

2. **Boxplot and statistical threshold using the IQR method**

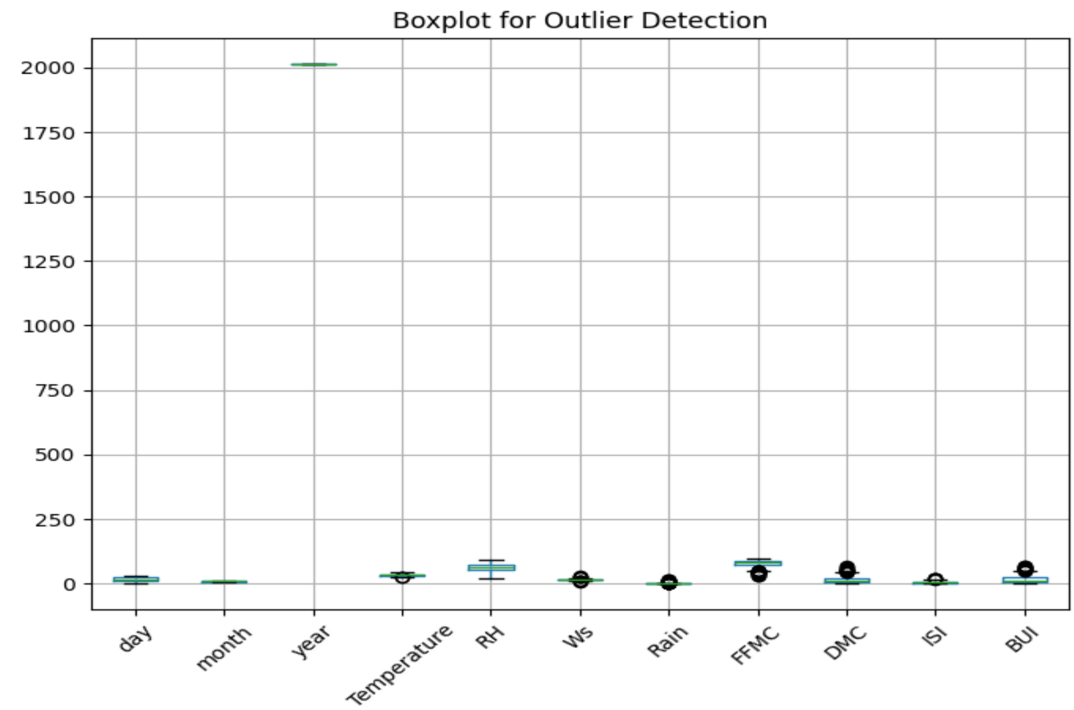

FIG 4: Outlier Detection

Outlier detection was performed using both boxplots and the Interquartile Range (IQR) statistical method to identify unusually high or low values across numerical features. The IQR approach calculated lower and upper bounds for each feature, flagging values that fell outside the 1.5 × IQR range as potential outliers. While some features like day, month, year, RH, and DC showed no outliers, others such as Temperature, Wind Speed, Rain, FFMC, DMC, ISI, and BUI contained multiple extreme values. These findings helped reveal variability patterns within the dataset and informed decisions for further cleaning and preprocessing.

3. Density Plots and histogram for each feature

Histograms with KDE density curves were generated for all numerical features to visualize their distributions and identify underlying patterns. These plots helped reveal whether features were normally distributed, skewed, or contained extreme values. Variables like Temperature, RH, and Wind Speed showed smoother, more symmetric distributions, while features such as Rain, DMC, ISI, and BUI displayed strong right-skewness. This visual exploration provided valuable insights for understanding variability and preparing the data for modeling.

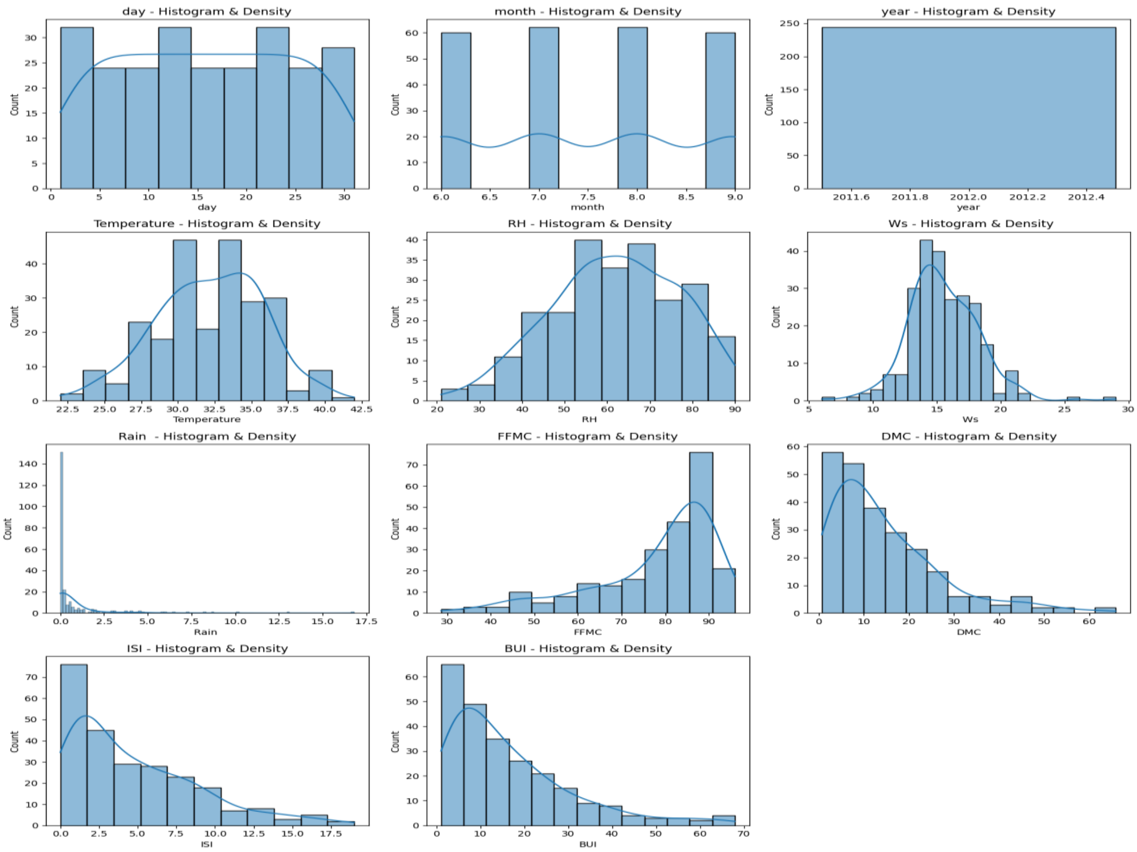

FIG 4: Histogram with KDE curves for features

4. Compute the correlation matrix for numerical features

A correlation matrix was generated to explore relationships among numerical features and identify which variables influence each other most strongly. The heatmap revealed that fire danger indices such as BUI, DMC, FFMC, and ISI have high positive correlations with each other and with Temperature, indicating shared patterns in fire behavior. Relative Humidity showed strong negative correlations with several indices, reflecting its inverse effect on fire risk. These insights helped highlight key predictive features for the Fire Weather Index and guided feature selection for modeling.

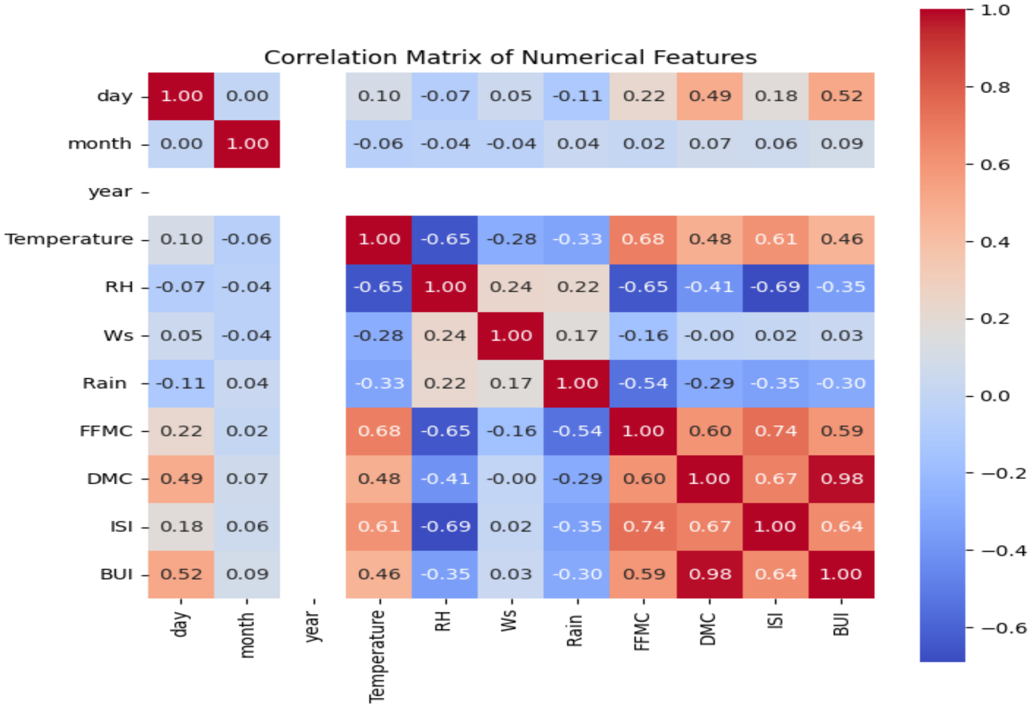

FIG 5: Correlation matrix

5. **Scatter plots for Features vs FWI**
Scatterplots were generated to examine how each key environmental feature relates to the Fire Weather Index (FWI). These visualizations helped reveal whether variables like Temperature, RH, Wind Speed, Rain, FFMC, DMC, and ISI show increasing, decreasing, or nonlinear trends with FWI. Features such as FFMC, DMC, and ISI displayed clearer upward patterns, indicating stronger influence on fire danger levels. In contrast, variables like Rain and Wind Speed showed weaker or more scattered relationships, helping identify which predictors contribute most to FWI modeling.

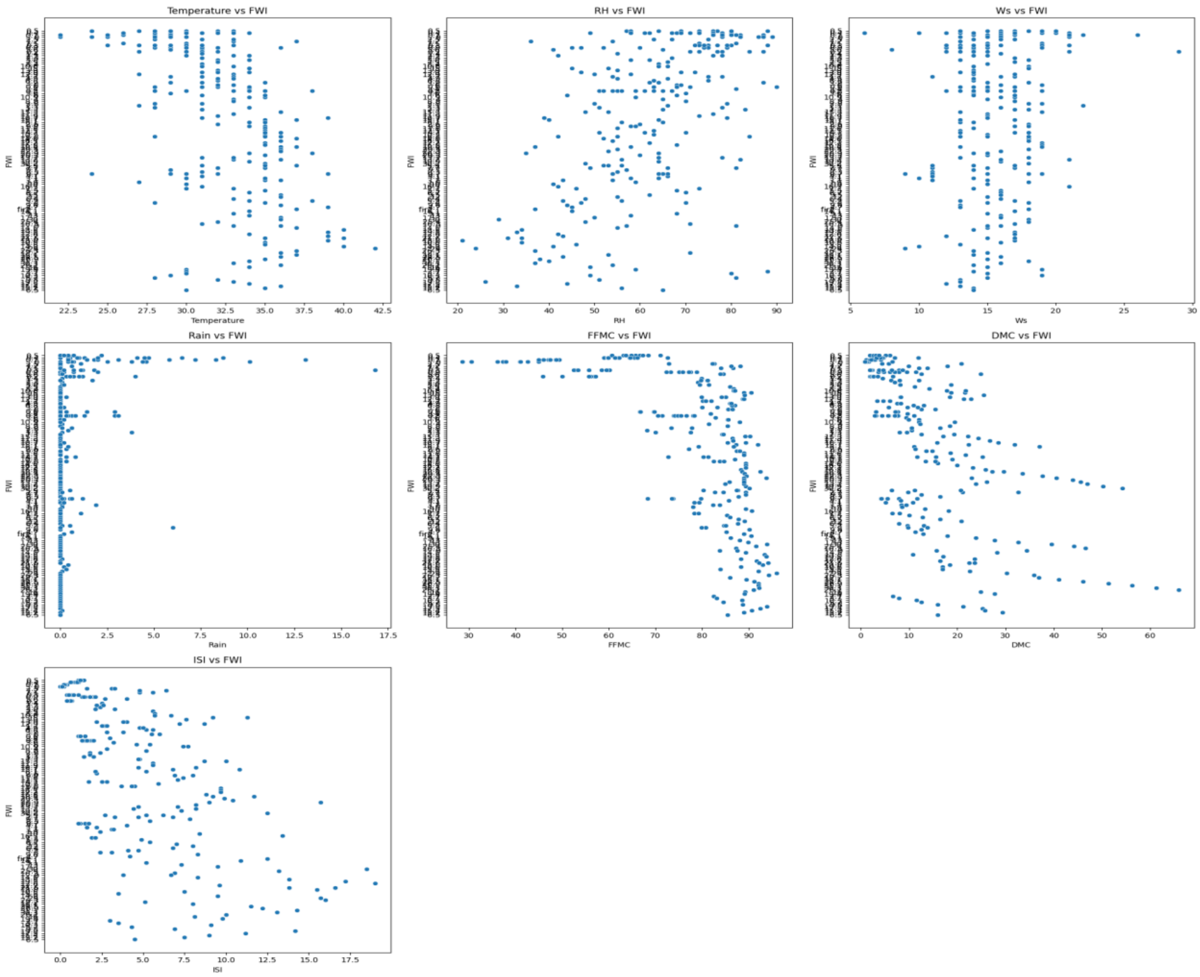

FIG 6: Scatterplot for features and FWI

6. Encoding region

```
Encoded Values:
  Region  Region_encoded
0  Bejaia              0
1  Bejaia              0
2  Bejaia              0
3  Bejaia              0
4  Bejaia              0
Region Mapping:
{'Bejaia': np.int64(0), 'Sidi-Bel Abbes': np.int64(1)}
Region              object
Region_encoded    category
dtype: object
```

The categorical feature *Region* was converted into numerical form using label encoding to make it suitable for machine learning models. Each unique region was assigned a numeric label, with "Bejaia" encoded as 0 and "Sidi-Bel Abbes" encoded as 1. A new column, *Region_encoded*, was created to store these encoded values. The encoded column was then converted to a categorical datatype to maintain consistency with the original feature.