FIRE WEATHER INDEX (FWI) PREDICTION MODEL

Milestone 1:

Data Collection & Data Exploration and Preprocessing

Submitted by:

Himanshu Ramole

Infosys Springboard Mentor:

Praveen

Date: 11-12-2025

1. Introduction

This report summarizes the work completed as part of **Milestone 1 (Week 1–2)** of the Infosys Springboard Virtual Internship project: *Fire Weather Index Prediction*. The objective for this milestone was to complete **Module 1: Data Collection** and **Module 2: Data Exploration & Preprocessing**, ensuring the dataset is cleaned, analyzed, and prepared for further modeling in Milestone 2.

The dataset contains meteorological and fire-index parameters for two regions in Algeria, which form the basis of the FWI prediction system.

Module 1 – Data Collection

The goal of Module 1 was to load the dataset, validate its structure, check data types, and perform an initial inspection to understand data quality.

1.1 Data Loading

The dataset was loaded into a Pandas DataFrame from the local machine.

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import os


raw_path = r"C:\Infosys Springboard 6.0 Internship\Datasets\FWI Dataset.csv"

df = pd.read_csv(raw_path)


print("Dataset loaded successfully.")

df.head()
```

1.2 Structural Inspection & Column Cleaning

This step verified column names, data types, and formatting consistency.

```
print("Info before cleaning:")
```

```
df.info()


print("\nOriginal columns:")
print(df.columns.tolist())


df.columns = df.columns.str.strip()


print("\nColumns after stripping spaces:")
print(df.columns.tolist())
```

1.3 Descriptive Statistics & Missing Value Overview

```
print("Descriptive statistics:")
display(df.describe())


print("\nMissing values before handling:")
print(df.isnull().sum())
```

This provided a high-level understanding of numerical distributions and identified missing entries requiring preprocessing.

Module 2 – Data Exploration & Preprocessing

The objective of Module 2 was to prepare the dataset for machine learning by handling missing values, correcting data formats, detecting outliers, visualizing feature distributions, and encoding categorical fields.

2.1 Missing Value Handling & Data Type Corrections

The dataset contained minor inconsistencies, especially in the *DC*, *FWI*, and *Classes* fields.

```
if df['Classes'].isnull().sum() > 0:
    mode_classes = df['Classes'].mode()[0]
    df['Classes'].fillna(mode_classes, inplace=True)
    print(f"Filled missing 'Classes' with mode: {mode_classes}")

for col in ['DC', 'FWI']:
    df[col] = df[col].astype(str).str.replace(" ", "", regex=False)
    df[col] = pd.to_numeric(df[col], errors='coerce')

print("\nMissing values in DC & FWI after conversion:")
print(df[['DC', 'FWI']].isnull().sum())

for col in ['DC', 'FWI']:
    if df[col].isnull().sum() > 0:
        mean_val = df[col].mean()
        df[col].fillna(mean_val, inplace=True)
        print(f"Filled missing '{col}' with mean: {mean_val:.4f}")

print("\nMissing values after handling:")
print(df.isnull().sum())
```

2.2 Cleaning Categorical Fields

```
df['Classes'] = df['Classes'].astype(str).str.strip()
df['Region'] = df['Region'].astype(str).str.strip()

print("\nUnique values in Classes:")
print(df['Classes'].value_counts())

print("\nUnique values in Region:")
print(df['Region'].value_counts())
```

2.3 Distribution Analysis (Histograms)

```
numeric_cols = ['Temperature', 'RH', 'Ws', 'Rain',
            'FFMC', 'DMC', 'DC', 'ISI', 'BUI', 'FWI']

df[numeric_cols].hist(figsize=(12, 10))

plt.tight_layout()

plt.show()
```

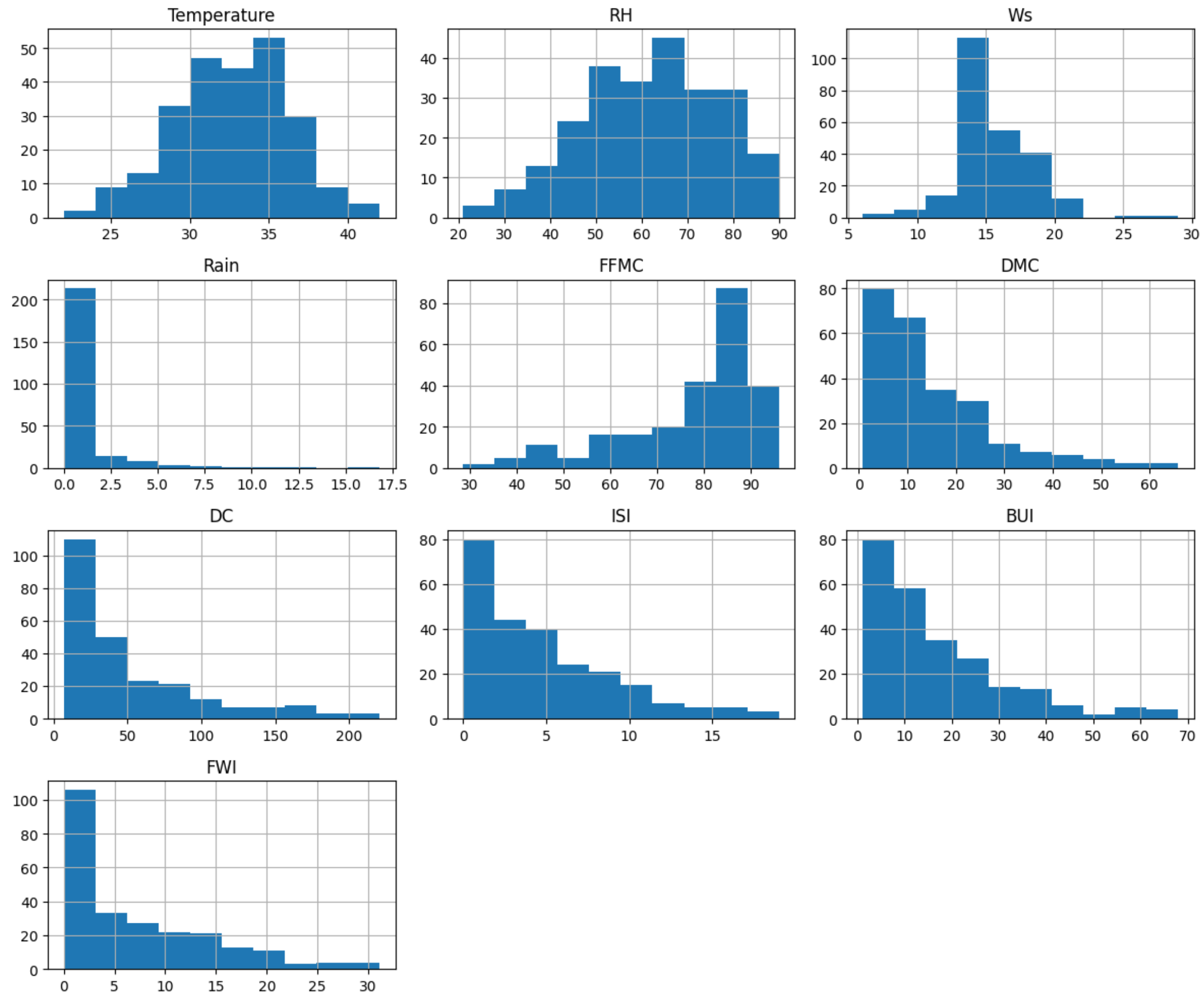


Fig: Histrograms

2.4 Outlier Detection (Boxplots)

```
plt.figure(figsize=(12, 8))

for i, col in enumerate(numeric_cols, 1):

    plt.subplot(3, 4, i)
    sns.boxplot(y=df[col])
    plt.title(col)
plt.tight_layout()
plt.show()
```

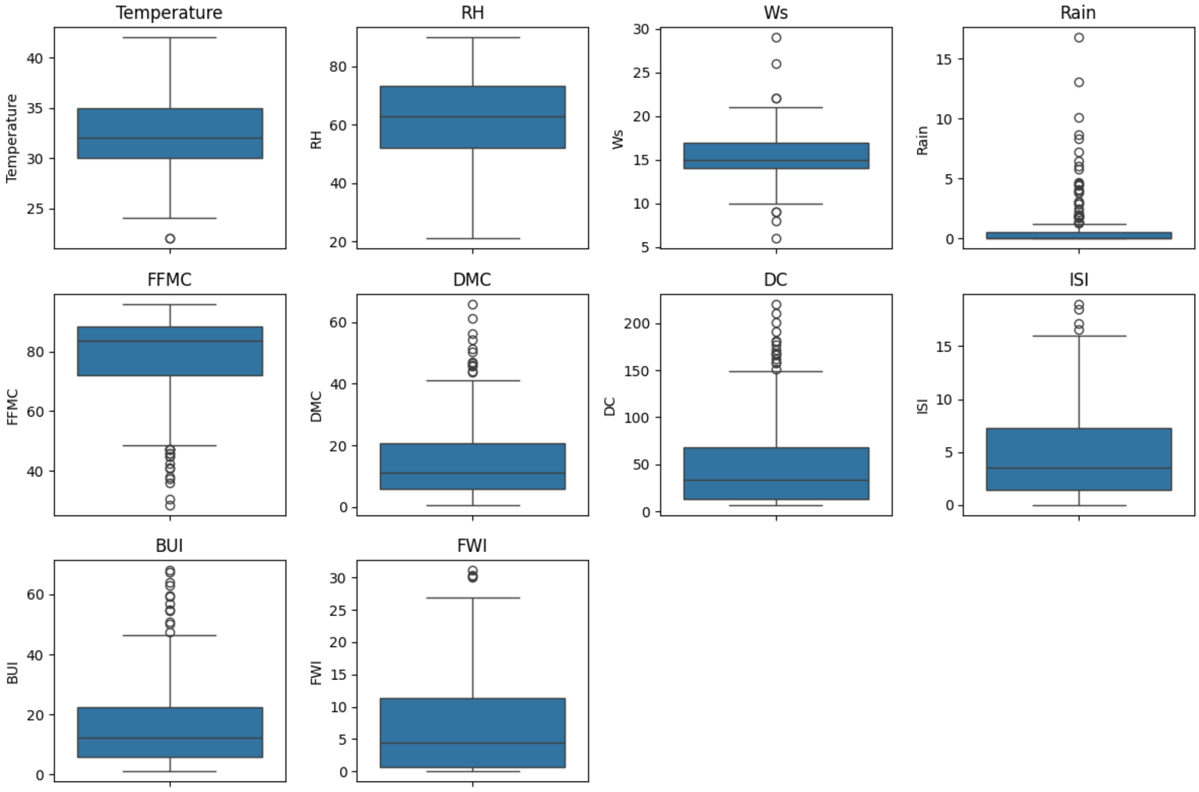


Fig: Boxplots

2.5 Correlation Analysis

```
numeric_df = df.select_dtypes(include='number')


plt.figure(figsize=(10, 6))
corr = numeric_df.corr()
sns.heatmap(corr, annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap of Numerical Features")
plt.show()
```

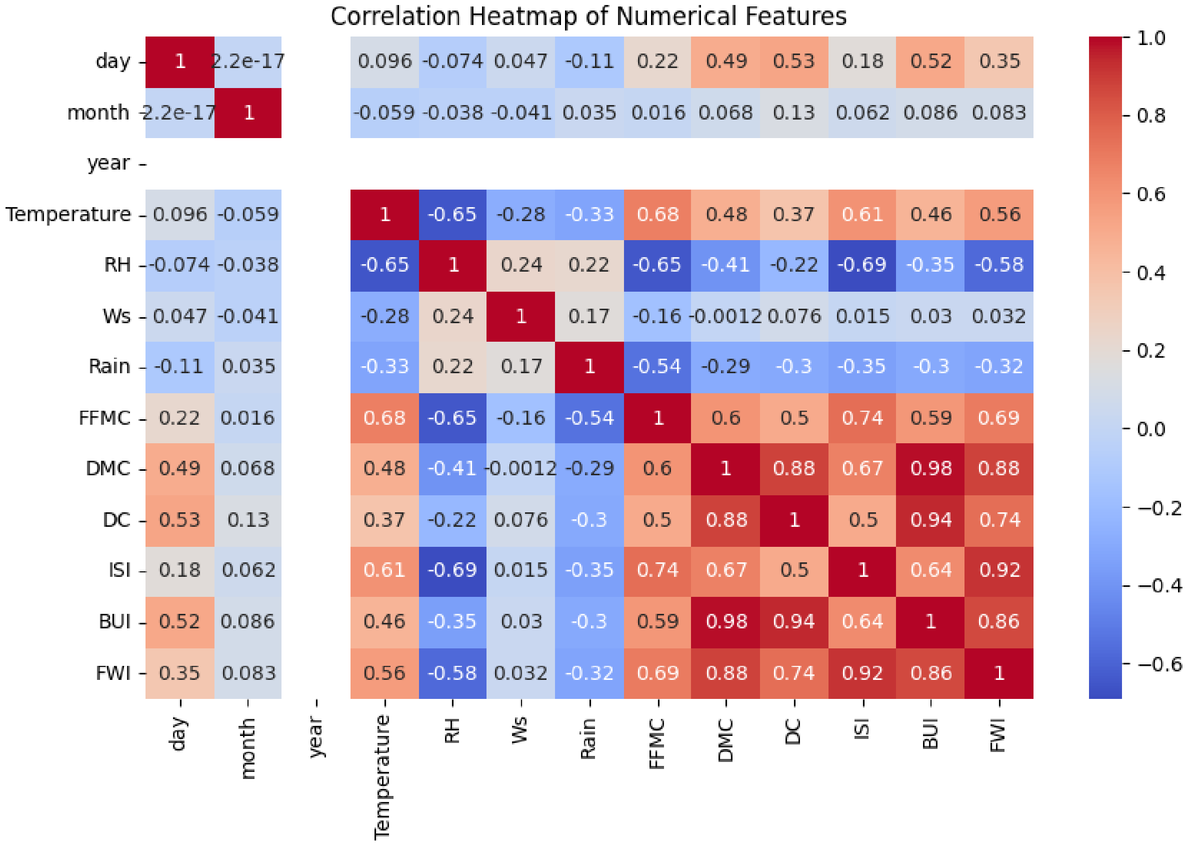
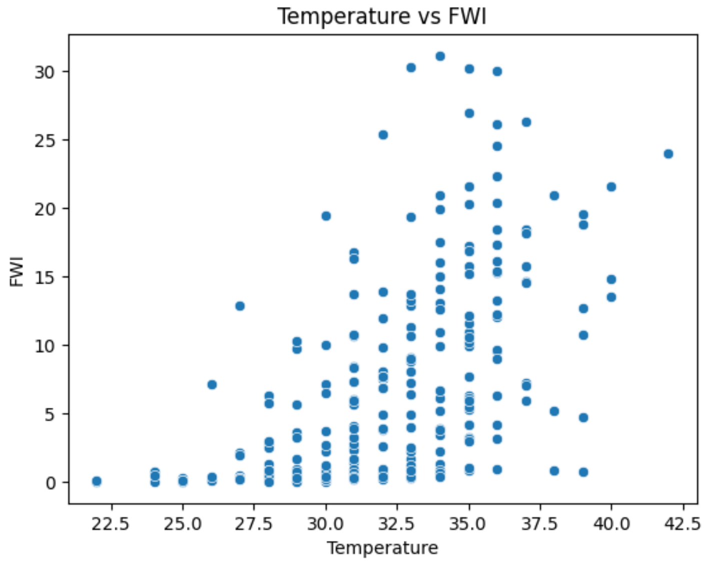


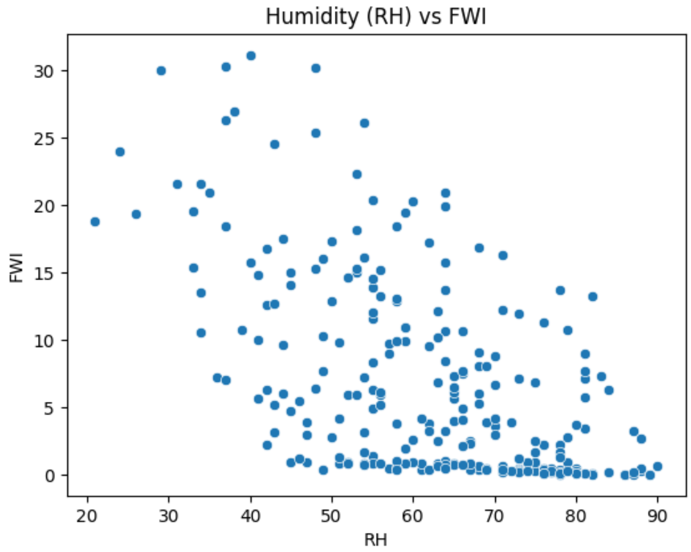
Fig: Correlation Heatmap of Numerical Features

2.6 Feature Relationship Visualizations (Scatterplots)

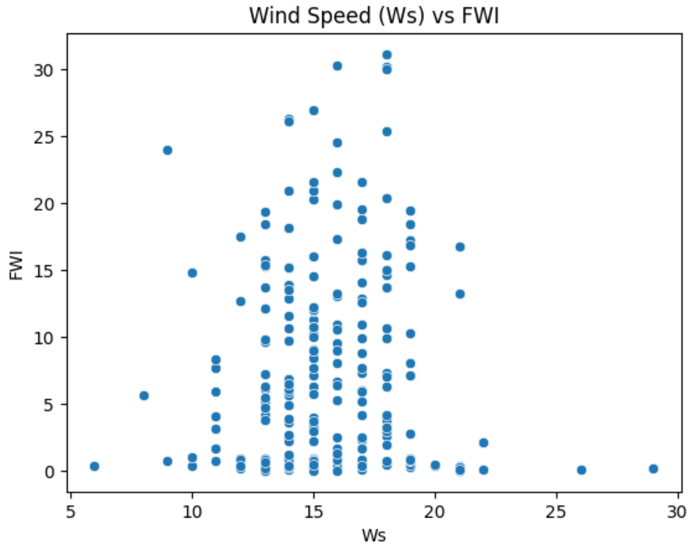
```
sns.scatterplot(data=df, x='Temperature', y='FWI')

plt.title("Temperature vs FWI")

plt.show()
```



```
sns.scatterplot(data=df, x='RH', y='FWI')

plt.title("Humidity (RH) vs FWI")

plt.show()
```



```
sns.scatterplot(data=df, x='Ws', y='FWI')

plt.title("Wind Speed (Ws) vs FWI")

plt.show()
```



These visualizations highlight linear and nonlinear dependencies.

2.7 Region Encoding

For model training, categorical fields must be converted to numerical labels.

```
region_mapping = {'Bejaia': 0, 'Sidi-Bel Abbes': 1}
df['Region'] = df['Region'].map(region_mapping)

print("Region after encoding:")
print(df['Region'].unique())

print("\nFinal dtypes after preprocessing:")
print(df.dtypes)
```

2.8 Creating the Final Cleaned Dataset

The *Classes* column is dropped for regression model training.

```
df_clean = df.drop(columns=['Classes'])

print("Columns in cleaned dataset (Classes dropped):")
print(df_clean.columns.tolist())
```

2.9 Saving the Cleaned Dataset

```
save_dir = r"C:\Infosys Springboard 6.0 Internship\Datasets"
os.makedirs(save_dir, exist_ok=True)

save_path = os.path.join(save_dir, "FWI_Cleaned.csv")
df_clean.to_csv(save_path, index=False)

print(f"Cleaned dataset saved to: {save_path}")
```

The cleaned dataset is now ready for **Milestone 2 (Feature Engineering & Modeling)**.

Conclusion

Milestone 1 successfully established a solid foundation for the Fire Weather Index Prediction system. The dataset was fully validated, cleaned, analyzed, and transformed into a high-quality, model-ready format. All deliverables for **Module 1** and **Module 2** have been completed as required.