Fire Weather Index Predictor

[A Machine Learning Model to Predict Fire Weather Index]

Infosys SpringBoard Virtual Internship Program

Submitted by

U P Mahendra

Under the guidance of Mentor **Praveen**

Project Statement

Project Statement: Wildfires pose a significant threat to ecosystems, human life, and property. The Fire Weather Index (FWI) is a crucial tool used by meteorological and environmental agencies worldwide to estimate wildfire potential. This project aims to build a machine learning model that predicts FWI based on real-time environmental data, enabling proactive wildfire risk management. The model is trained using Ridge Regression, deployed via a Flask web application, and supports early warning systems for wildfire hazards.

Expected Outcomes

- A predictive ML model trained using Ridge Regression to forecast FWI.
- A pre-processing pipeline using StandardScaler for normalization.
- 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 Optimization
- Deployment via Flask App
- Presentation and Documentation

Requirement tools

Numpy == 2.2.6

pandas==2.3.3

pytz==2025.2

six==1.17.0

tzdata==2025.

MODULE 1: DATA COLLECTION & INITIAL DATA INSPECTION

This module focuses on collecting the dataset, loading it into Python, cleaning column names, mapping region values, inspecting random records, checking missing values, converting data types, and understanding the structure of the dataset using info and null counts.

Step-by-Step Code – Module 1

```
# Step 1: Import Pandas
import pandas as pd

# Step 2: Load the dataset
file_path = "FWI Dataset.csv"
df = pd.read_csv(file_path)

# Step 3: Clean column names
df.columns = df.columns.str.strip()

# Step 4: Map Region values (if encoded)
region_mapping = {0: "Bejaia", 1: "Sidi-Bel Abbes"}
if "Region" in df.columns:
    if df["Region"].dtype != "object":
        df["Region"] = df["Region"].map(region_mapping)

# Step 5: Display random records
print(df.sample(5))

# Step 6: Show rows containing missing values
print(df[df.isnull().any(axis=1)])

# Step 7: Convert DC and FWI to numeric
df["DC"] = pd.to_numeric(df["DC"], errors="coerce")
df["FWI"] = pd.to_numeric(df["FWI"], errors="coerce")

# Step 8: Dataset information
df.info()

# Step 9: Missing values count
print(df.isnull().sum())
```

MODULE 2: DATA EXPLORATION & DATA PREPROCESSING

This module focuses on handling missing values, detecting outliers using IQR, visualizing feature distributions using histograms and boxplots, performing correlation analysis, encoding categorical variables, removing unnecessary columns, and saving the final cleaned dataset.

Step-by-Step Code – Module 2

```
# Step 1: Import required libraries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from sklearn.preprocessing import LabelEncoder

sns.set(style="whitegrid")

# Step 2: Handle missing values
if "Classes" in df.columns:
    df["Classes"] = df["Classes"].fillna(method="ffill")

numeric_cols = df.select_dtypes(include=["int64","float64"]).columns
for col in numeric_cols:
    df[col] = df[col].fillna(df[col].mean())

print(df.isnull().sum())

# Step 3: Outlier detection using IQR and boxplots
num_cols = df.select_dtypes(include=["int64","float64"]).columns
for col in num_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1

    outliers = df[(df[col] < Q1 - 1.5 * IQR) | (df[col] > Q3 + 1.5 * IQR)]
    print(f"{col}: {outliers.shape[0]} outliers")

    plt.figure(figsize=(6,4))
    df.boxplot(column=[col])
    plt.title(f"Boxplot of {col}")
    plt.tight_layout()
    plt.show()

# Step 4: Correlation matrix and heatmap
cont_cols = ['Temperature','RH','Ws','Rain','FFMC','DMC','DC','ISI','BUI','FWI']
corr = df[cont_cols].corr()

plt.figure(figsize=(10,8))
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Matrix Heatmap")
plt.tight_layout()
plt.show()

# Step 5: Label Encoding for Region
df["Region"] = df["Region"].astype(str).str.lower()
le = LabelEncoder()
df["Region_encoded"] = le.fit_transform(df["Region"])

# Step 6: Remove Classes column completely
if "Classes" in df.columns:
    df.drop(columns=["Classes"], inplace=True)

if "Classes_encoded" in df.columns:
    df.drop(columns=["Classes_encoded"], inplace=True)

# Step 7: Final datatype conversion
df["FWI"] = df["FWI"].round().astype("int64")
df["DC"] = df["DC"].round().astype("int64")

# Step 8: Save final cleaned dataset
```

```
df.to_csv("FWI_Dataset_Final.csv", index=False)

print("FINAL CLEANED DATASET SAVED AS: FWI_Dataset_Final.csv")
```

Final Conclusion

Module 1 and Module 2 together ensured that the Fire Weather Index dataset was properly collected, inspected, cleaned, explored, and prepared for feature engineering and machine learning model training.

FWI Exploratory Data Analysis – Figures

Figure 1

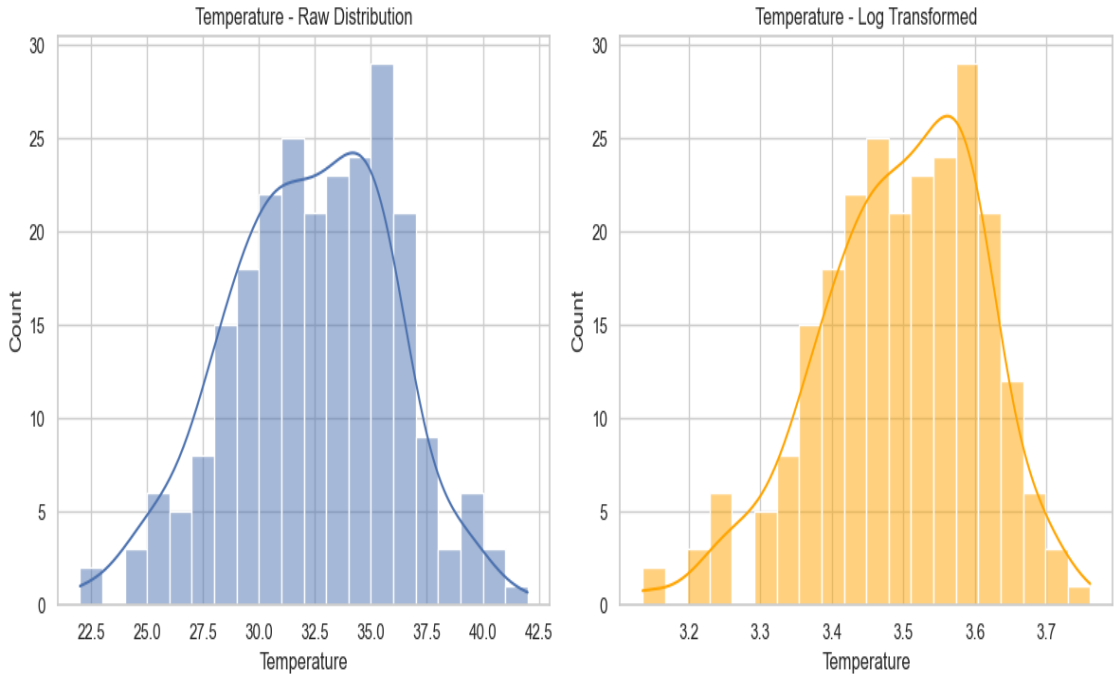

Figure 2

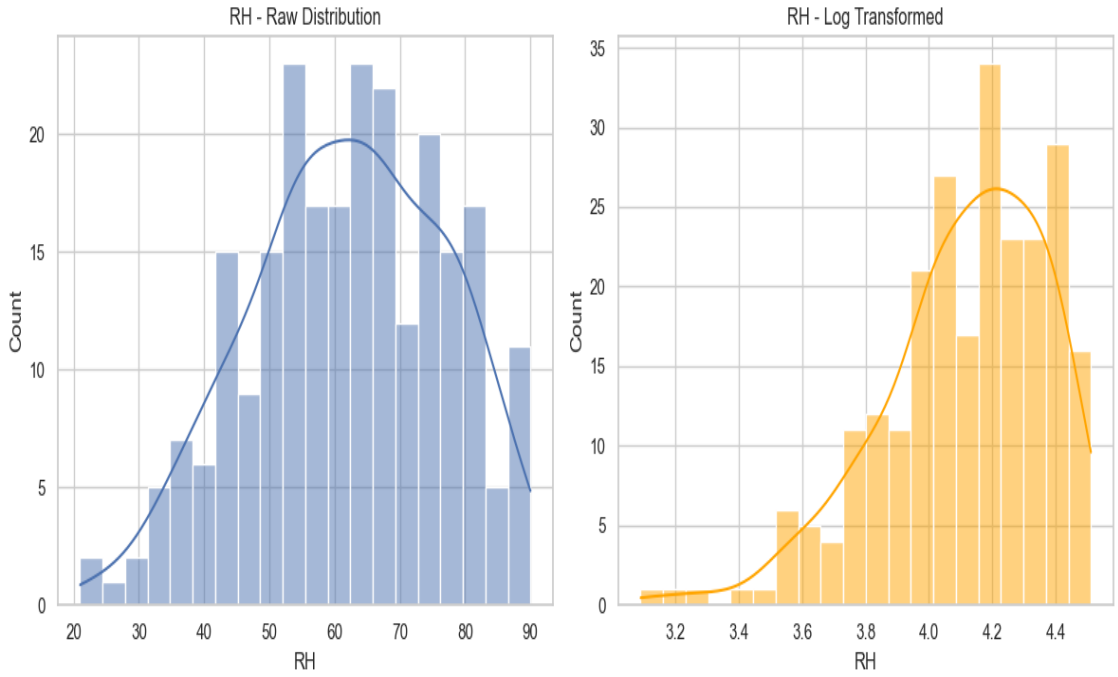
RH - Raw Distribution
Count
RH
0
5
10
15
20
20
30
40
50
60
70
80
90
RH - Log Transformed
Count
RH
0
5
10
15
20
25
30
35
3.2
3.4
3.6
3.8
4.0
4.2
4.4

Figure 3

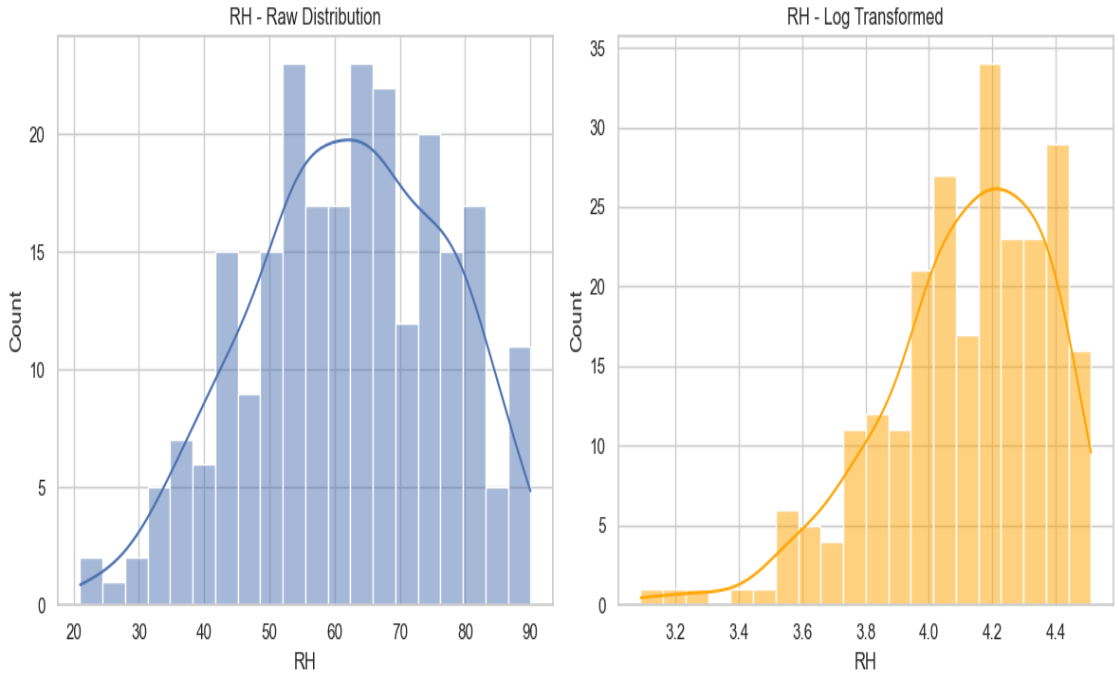
RH - Raw Distribution
Count
RH
0
5
10
15
20
20
30
40
50
60
70
80
90
RH - Log Transformed
Count
RH
0
5
10
15
20
25
30
35
3.2
3.4
3.6
3.8
4.0
4.2
4.4

Figure 4

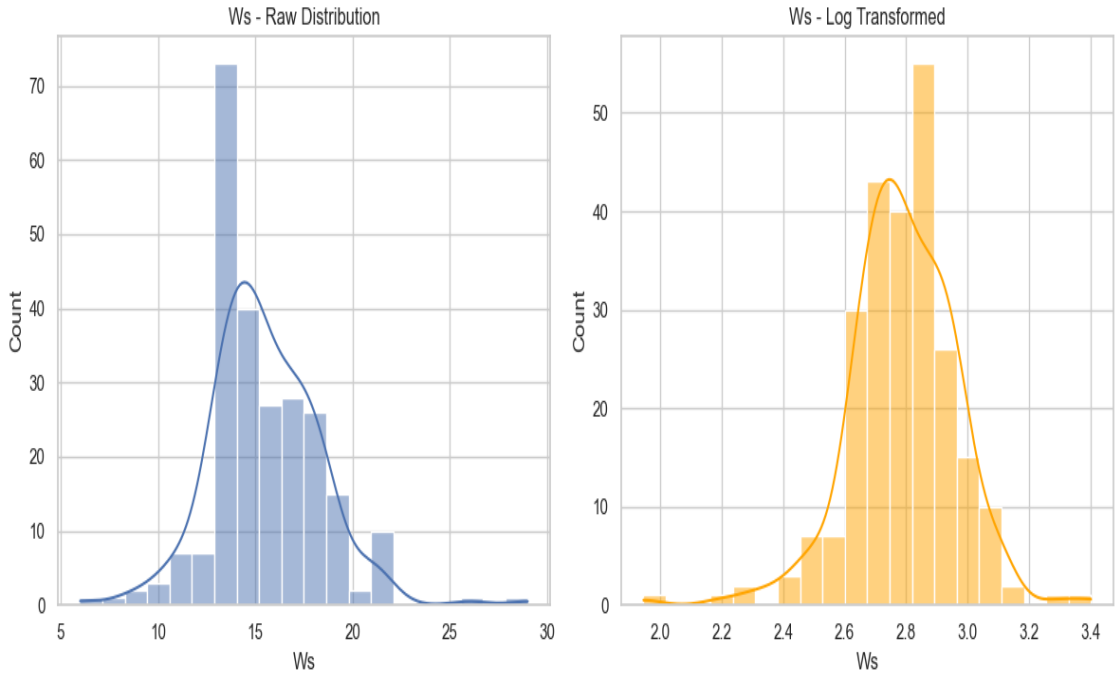
Ws - Raw Distribution
Count
Ws
0
10
20
30
40
50
60
70
5
10
15
20
25
30
Ws - Log Transformed
Count
Ws
0
10
20
30
40
50
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4

Figure 5

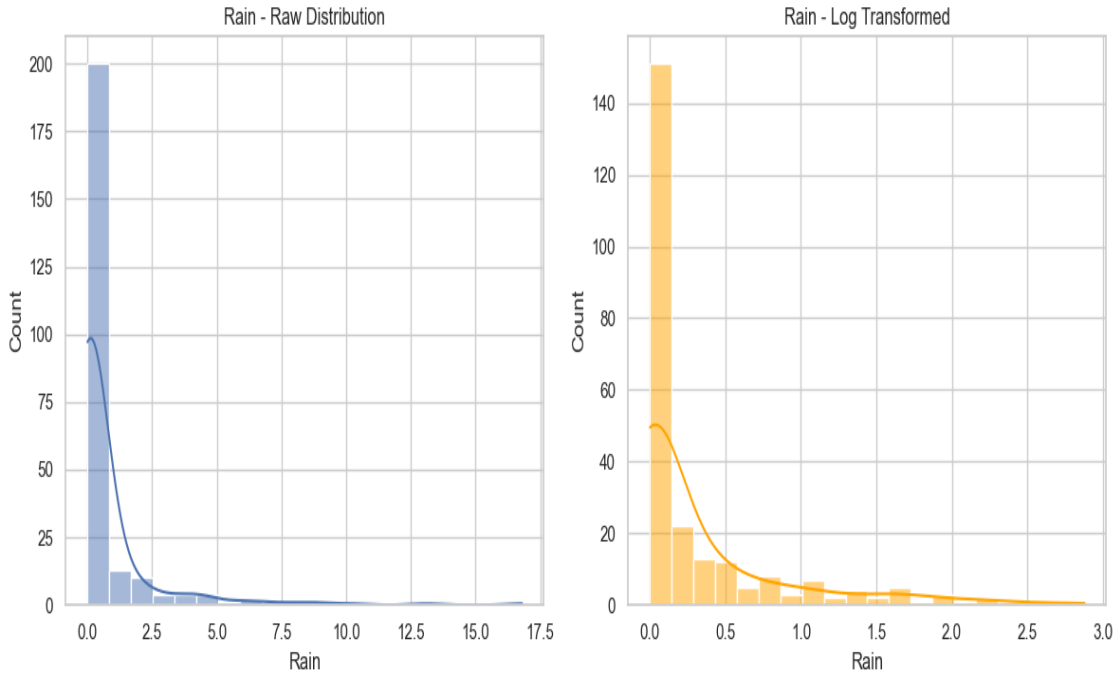
Rain - Raw Distribution
Count
Rain
0
25
50
75
100
125
150
175
200
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Rain - Log Transformed
Count
Rain
0
20
40
60
80
100
120
140
0.0
0.5
1.0
1.5
2.0
2.5
3.0

Figure 6

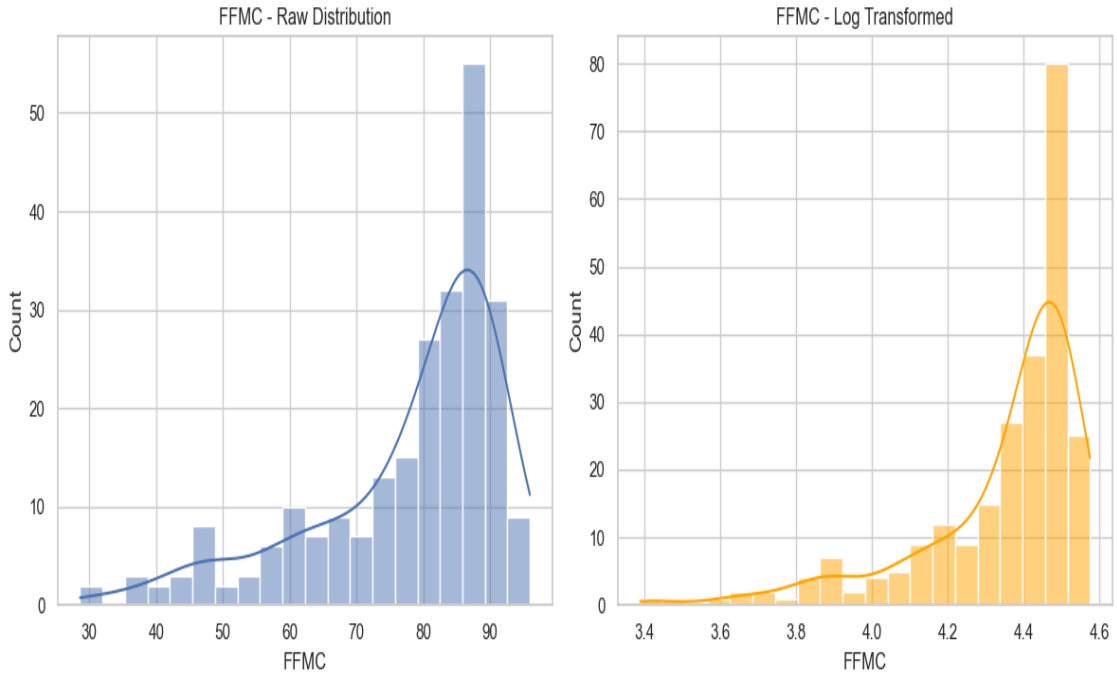
FFMC - Raw Distribution
Count
FFMC
0
10
20
30
40
50
30
40
50
60
70
80
90
FFMC - Log Transformed
Count
FFMC
0
10
20
30
40
50
60
70
80
3.4
3.6
3.8
4.0
4.2
4.4
4.6

Figure 7

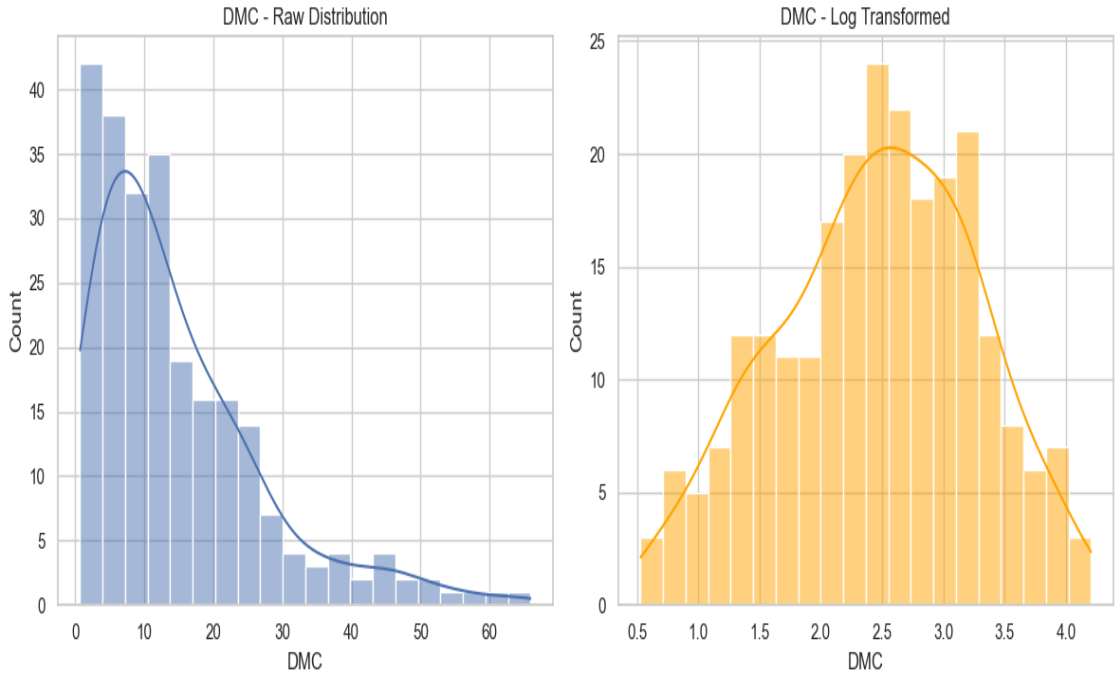
DMC - Raw Distribution
Count
DMC
0
5
10
15
20
25
30
35
40
0
10
20
30
40
50
60
DMC - Log Transformed
Count
DMC
0
5
10
15
20
25
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Figure 8

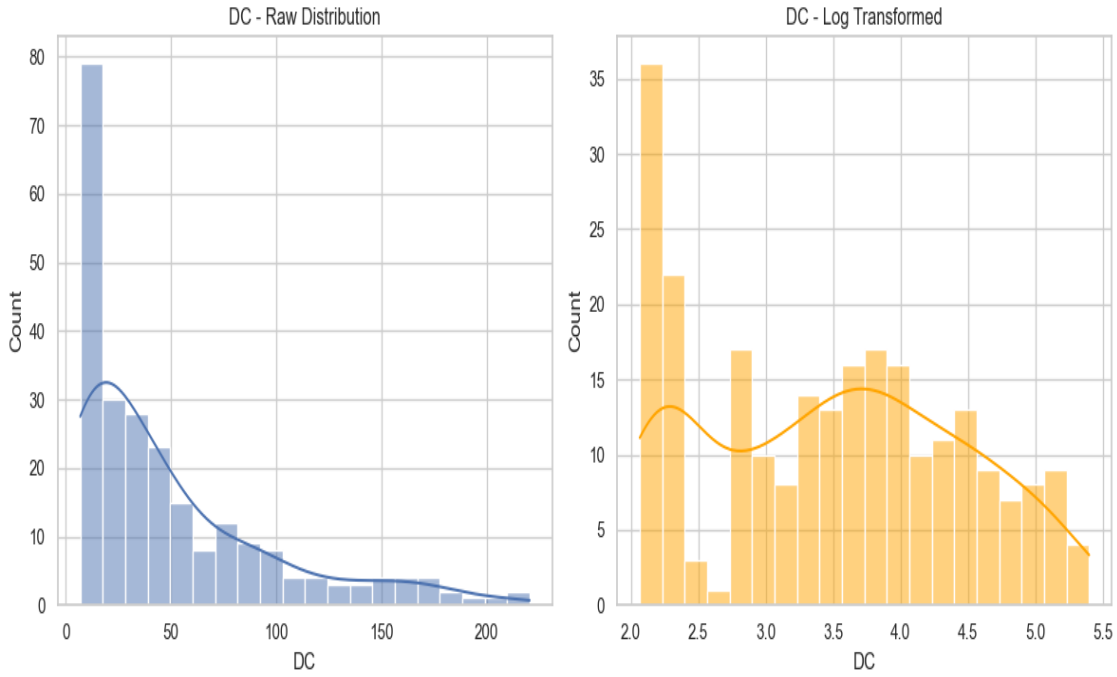
DC - Raw Distribution
Count
DC
0
10
20
30
40
50
60
70
80
0
50
100
150
200
DC - Log Transformed
Count
DC
0
5
10
15
20
25
30
35
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5

Figure 9

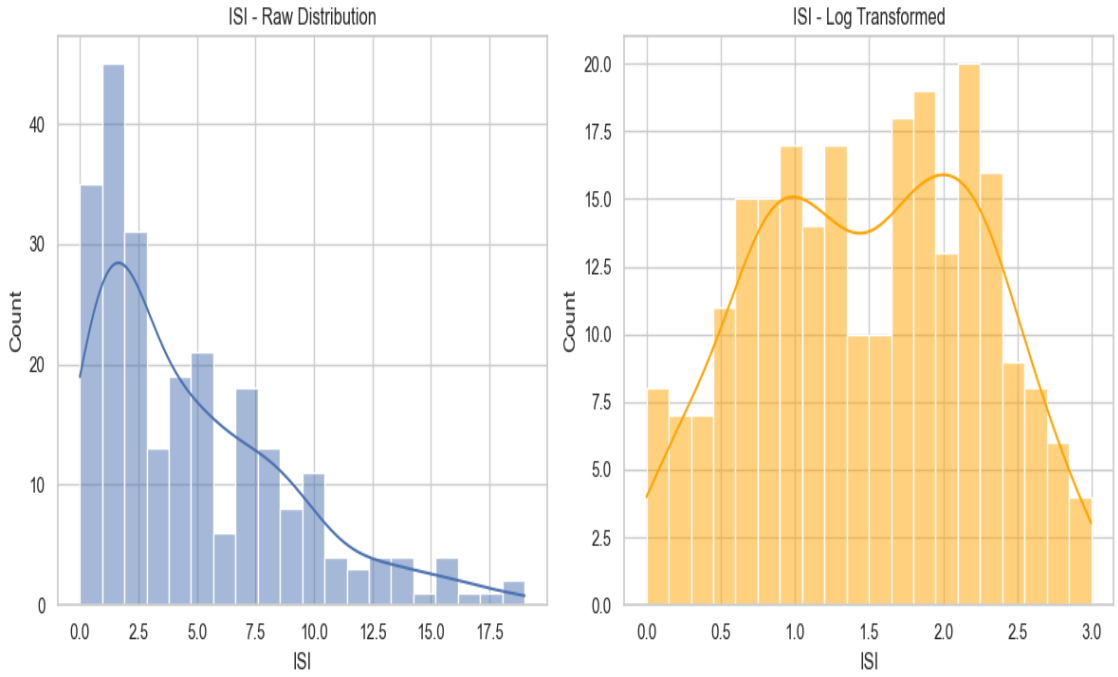
ISI - Raw Distribution
Count
ISI
0
10
20
30
40
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
ISI - Log Transformed
Count
ISI
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0

Figure 10

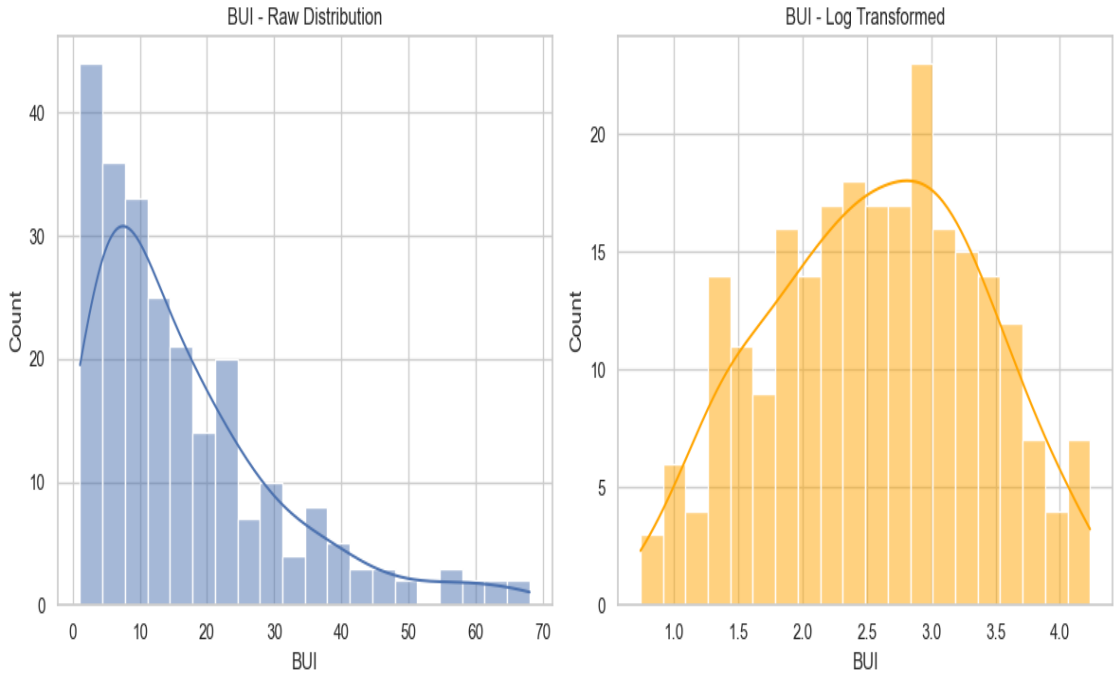
BUI - Raw Distribution
Count
BUI
0
10
20
30
40
50
60
70
BUI - Log Transformed
Count
BUI
0
5
10
15
20
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Figure 11

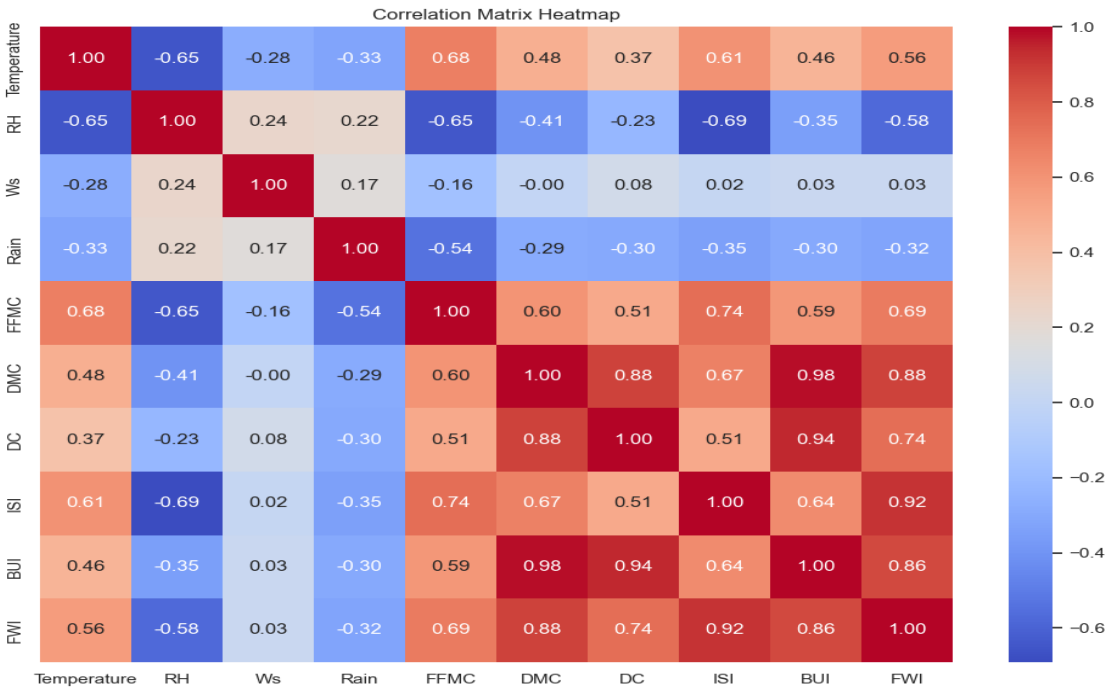
Correlation Matrix Heatmap
Temperature
RH
Ws
Rain
FFMC
DMC
DC
ISI
BUI
FWI
1.00 -0.65 -0.28 -0.33 0.68 0.48 0.37 0.61 0.46 0.56
-0.65 1.00 0.24 0.22 -0.65 -0.41 -0.23 -0.69 -0.35 -0.58
-0.28 0.24 1.00 0.17 -0.16 -0.00 0.08 0.02 0.03 0.03
-0.33 0.22 0.17 1.00 -0.54 -0.29 -0.30 -0.35 -0.30 -0.32
0.68 -0.65 -0.16 -0.54 1.00 0.60 0.51 0.74 0.59 0.69
0.48 -0.41 -0.00 -0.29 0.60 1.00 0.88 0.67 0.98 0.88
0.37 -0.23 0.08 -0.30 0.51 0.88 1.00 0.51 0.94 0.74
0.61 -0.69 0.02 -0.35 0.74 0.67 0.51 1.00 0.64 0.92
0.46 -0.35 0.03 -0.30 0.59 0.98 0.94 0.64 1.00 0.86
0.56 -0.58 0.03 -0.32 0.69 0.88 0.74 0.92 0.86 1.00
Temperature RH Ws Rain FFMC DMC DC ISI BUI FWI
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Figure 12

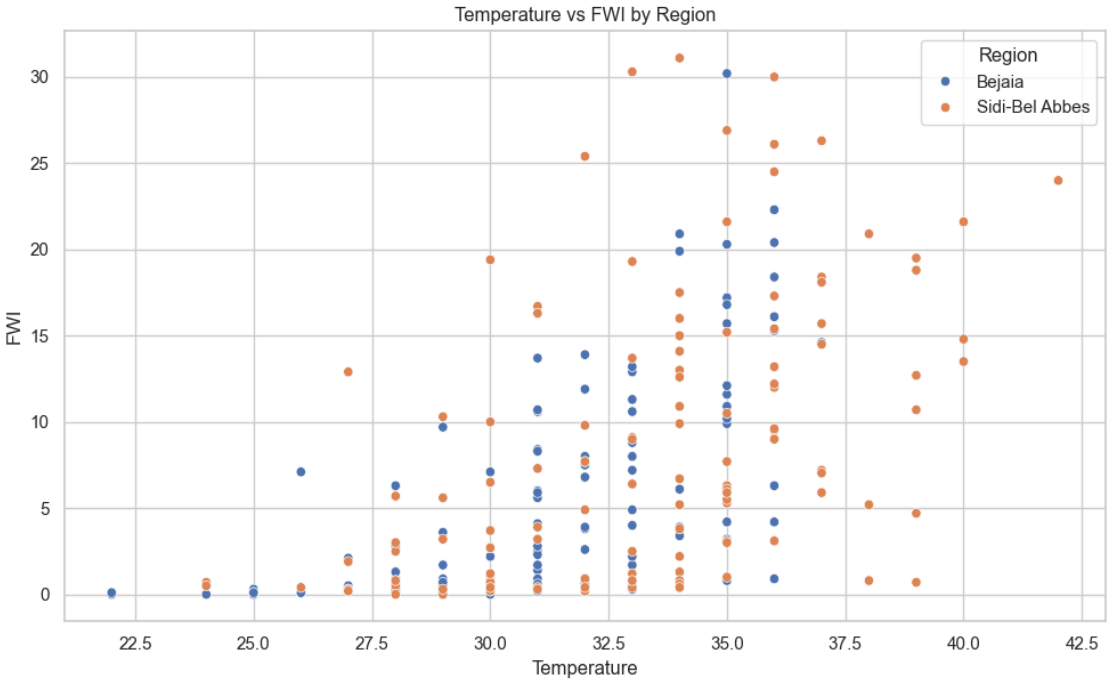
Temperature vs FWI by Region
Region
Bejaia
Sidi-Bel Abbes
FWI
Temperature
0
5
10
15
20
25
30
22.5
25.0
27.5
30.0
32.5
35.0
37.5
40.0
42.5

Figure 13

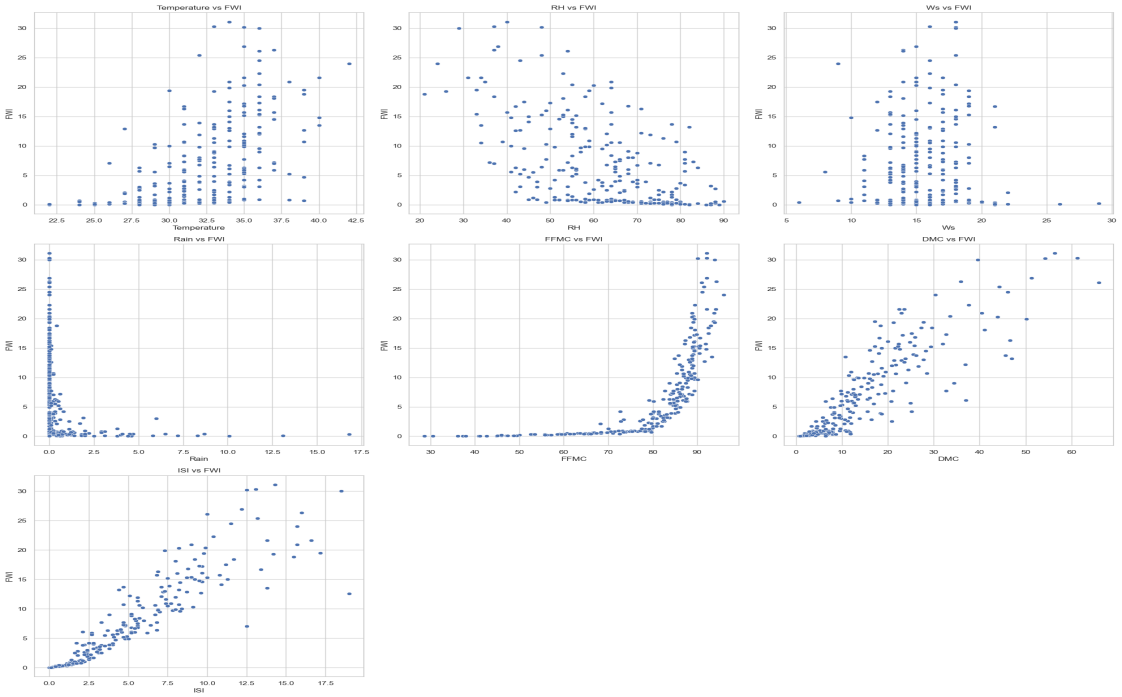
Temperature vs FWI
FWI
Temperature
RH vs FWI
FWI
RH
Ws vs FWI
FWI
Ws
Rain vs FWI
FWI
Rain
FFMC vs FWI
FWI
FFMC
DMC vs FWI
FWI
DMC
ISI vs FWI
FWI
ISI

Figure 14


Pair Plot of Key Numeric Features
RH
DMC
DC
BUI
FWI
Temperature
RH
DMC
DC
BUI
FWI

Execution Output

```
Output 1:
Random rows from the dataset:
 day month year Temperature RH Ws Rain FFMC DMC DC ISI
191 9 8 2012 39 43 12 0.0 91.7 16.5 30.9 9.6
82 22 8 2012 36 55 18 0.0 89.1 33.5 151.3 9.9
149 28 6 2012 37 37 13 0.0 92.5 27.2 52.4 11.7
39 10 7 2012 33 69 13 0.7 66.6 6.0 9.3 1.1
4 5 6 2012 27 77 16 0.0 64.8 3.0 14.2 1.2
 BUI FWI Classes Region
191 16.4 12.7 fire Sidi-Bel Abbes
82 43.1 20.4 fire Bejaia
149 27.1 18.4 fire Sidi-Bel Abbes
39 5.8 0.5 not fire Bejaia
4 3.9 0.5 not fire Bejaia


Output 2:
Rows containing 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


Output 3:
DataFrame Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 15 columns):
# Column Non-Null Count Dtype
0 day 244 non-null int64
1 month 244 non-null int64
2 year 244 non-null int64
3 Temperature 244 non-null int64
4 RH 244 non-null int64
5 Ws 244 non-null int64
6 Rain 244 non-null float64
7 FFMC 244 non-null float64
8 DMC 244 non-null float64
9 DC 243 non-null float64
10 ISI 244 non-null float64
11 BUI 244 non-null float64
12 FWI 243 non-null float64
13 Classes 243 non-null object
14 Region 244 non-null object
dtypes: float64(7), int64(6), object(2)
memory usage: 28.7+ KB


Output 4–6:
Missing Values Handling Summary:

Missing Values BEFORE Handling:
day 0
month 0
year 0
Temperature 0
RH 0
Ws 0
Rain 0
FFMC 0
DMC 0
DC 1
ISI 0
BUI 0
FWI 1
Classes 1
Region 0
dtype: int64

Missing Values AFTER Handling:
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 0
Region 0
dtype: int64
```

```
Output 7:
Outlier Detection Using Boxplots and IQR:
day: 0 outliers
month: 0 outliers
year: 0 outliers
Temperature: 2 outliers
RH: 0 outliers
Ws: 8 outliers
Rain: 35 outliers
FFMC: 16 outliers
DMC: 12 outliers
DC: 15 outliers
ISI: 4 outliers
BUI: 12 outliers
FWI: 4 outliers
```

```
Output 8:
FWI Correlation Summary:
FWI 1.000000
ISI 0.916343
DMC 0.875827
BUI 0.857628
DC 0.739521
FFMC 0.690289
Temperature 0.564599
Ws 0.032315
Rain -0.324369
RH -0.577577
```

```
Output 9:
Data Types After Final Casting:
FWI int64
DC int64
dtype: object
```