A DOCUMENTATION ON

FIRE WEATHER INDEX PREDICTOR

Project Duration

DEC 2025 – JAN 2026

Project Mentor

PRAVEEN

PROJECT SUBMITTED BY

Pranjali Kolawale

Infosys Springboard Internship Program

INDEX

5. Module 5: Model Evaluation

FIGURES

FIRE WEATHER INDEX PREDICTOR

INTRODUCTION

Wildfires have become more common in recent years and can cause serious damage to forests, animals, houses, and even people. Changes in climate, such as higher temperatures and uneven rainfall, make it easier for fires to start and spread. Because of this, it is important to know the fire danger level in advance so that forest officials and emergency teams can prepare and take action on time.

The Fire Weather Index (FWI) is a measure used to understand how risky the weather conditions are for a wildfire to occur. It depends on factors like temperature, humidity, wind speed, rainfall, and the dryness of forest fuel. Checking all these conditions manually can be slow and may not always give accurate predictions.

To solve this problem, this project aims to build a machine learning model that can predict the FWI using environmental data. By analysing the dataset and creating a predictive system, the project helps in giving early warnings and improving wildfire risk management.

PROBLEM STATEMENT

Wildfires can spread quickly and cause heavy damage, so it is important to know the fire risk in advance. The Fire Weather Index (FWI) helps measure how likely a fire is to start, but calculating it manually is not always accurate or fast. To solve this, the aim of this project is to use environmental factors like temperature, humidity, wind speed, and rainfall to build a machine learning model that can predict the FWI automatically. This will help in giving early warnings and improving wildfire preparedness.

OBJECTIVES

- To study the weather factors that affect the Fire Weather Index (FWI).
- To clean and prepare the dataset for accurate analysis.
- To explore the data and understand its patterns.
- To build a machine learning model that can predict FWI.
- To test the model and improve its accuracy.
- To create a simple web app for making FWI predictions.

MODULE 1: DATA COLLECTION AND INITIAL DATA PREPARATION

1.1 Loading Dataset and Explored Structure

Collected the Fire Weather Index (FWI) Dataset from the Kaggle. Then dataset was imported into a Pandas DataFrame. Basic exploration was done by checking the first few rows, the shape of the dataset, column names, and data types. This helped in understanding the kind of information available, such as weather parameters (*Temperature*, *RH*, *Ws*, *Rain*) and FWI-related indices (*FFMC*, *DMC*, *DC*, *ISI*, *BUI*, *FWI*), along with the *Classes* column representing fire risk levels.

1.2 Identifying Missing Values and Incorrect Values

The dataset was checked for missing entries and inconsistencies. Some columns, particularly *Classes*, *FWI*, and *DC*, contained missing or misformatted values, like spaces within numbers or text-based numeric entries. Detecting these issues early is important to ensure accurate analysis and modeling later.

```
missing_row = df[df.isnull().any(axis=1)]

print(missing_row)
```

1.3 Cleaning Column Names and Formats

Column names were standardized by removing extra spaces. Numeric columns that had text or unwanted spaces, such as *DC*, were cleaned by stripping spaces and converting them into proper numeric types. This step ensures that all calculations and analyses will be accurate and avoids errors during preprocessing.

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

df['DC'] = df['DC'].astype(str).str.strip()

df['DC'] = df['DC'].str.replace(' ', '')

df['DC'] = df['DC'].astype(float)
```

1.4 Handling Missing Values

Different strategies were applied based on column type. For the categorical *Classes* column, missing values were filled with the mode, as it represents the most common fire risk category. For numeric columns like *FWI*, invalid entries were replaced with the mean of the column after conversion to numeric. This ensures that the dataset remains complete and ready for exploratory analysis.

```
df['Classes'] = df['Classes'].fillna(df['Classes'].mode()[0])
```

1.5 Validating Data After Cleaning

After performing cleaning and handling missing values, the dataset was reviewed again using info() and sample rows. This confirmed that all corrections were applied properly, column types were consistent, and the dataset was ready for further analysis. This step helps prevent issues in downstream tasks like EDA or modeling.

```
print(df.info())
```

MODULE 2: DATA EXPLORATION AND PREPROCESSING

After collecting and cleaning the FWI dataset in Module 1, the next step was to explore the data and perform preprocessing to ensure it is ready for modeling. This module focused on understanding the data distribution, identifying patterns, handling categorical features, and preparing the dataset for machine learning.

2.1 Checking for Duplicates and Data Overview

The dataset was first examined for duplicate records, which could skew the analysis. Fortunately, no duplicates were found, confirming that each record was unique. Descriptive statistics were generated for numerical features to understand their distribution, central tendency, and variability. This helped in identifying potential anomalies, extreme values, or inconsistencies.

2.2 Handling Categorical Features

The dataset contained the categorical feature **Region**, which was converted into numerical values using label encoding. This transformation allowed the model to interpret region information effectively while maintaining the distinction between different regions.

```
df['Region'] = LabelEncoder().fit_transform(df['Region'])
```

2.3 Visualizing Data Distribution

Visualization played an important role in understanding the dataset:

- **Boxplots** provided insights into variations and outliers, particularly across regions, helping identify extreme values in feature like Temperature.

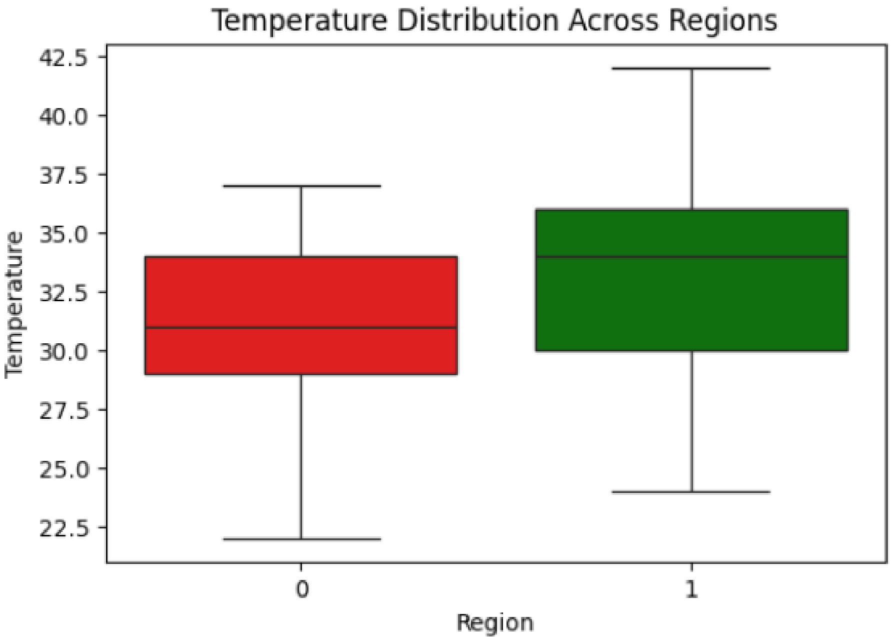


Figure 1

To understand how temperature values were spread across the dataset, I created a boxplot, which helped me quickly spot any unusual readings. After this, I applied the IQR method to calculate outliers. The first quartile (Q1) was 30.0 and the third quartile (Q3) was 35.0, giving an IQR of 5.0. Using these values, I calculated the lower limit as 22.5°C and the upper limit as 42.5°C. Any temperature outside this range is considered an outlier. Based on this, I found two outliers in the dataset, both with a temperature of 22°C at index numbers 93 and 105, which fall below the lower limit and indicate unusually low temperature values.

- **Histograms** were used to examine the distribution of numerical features in the data.

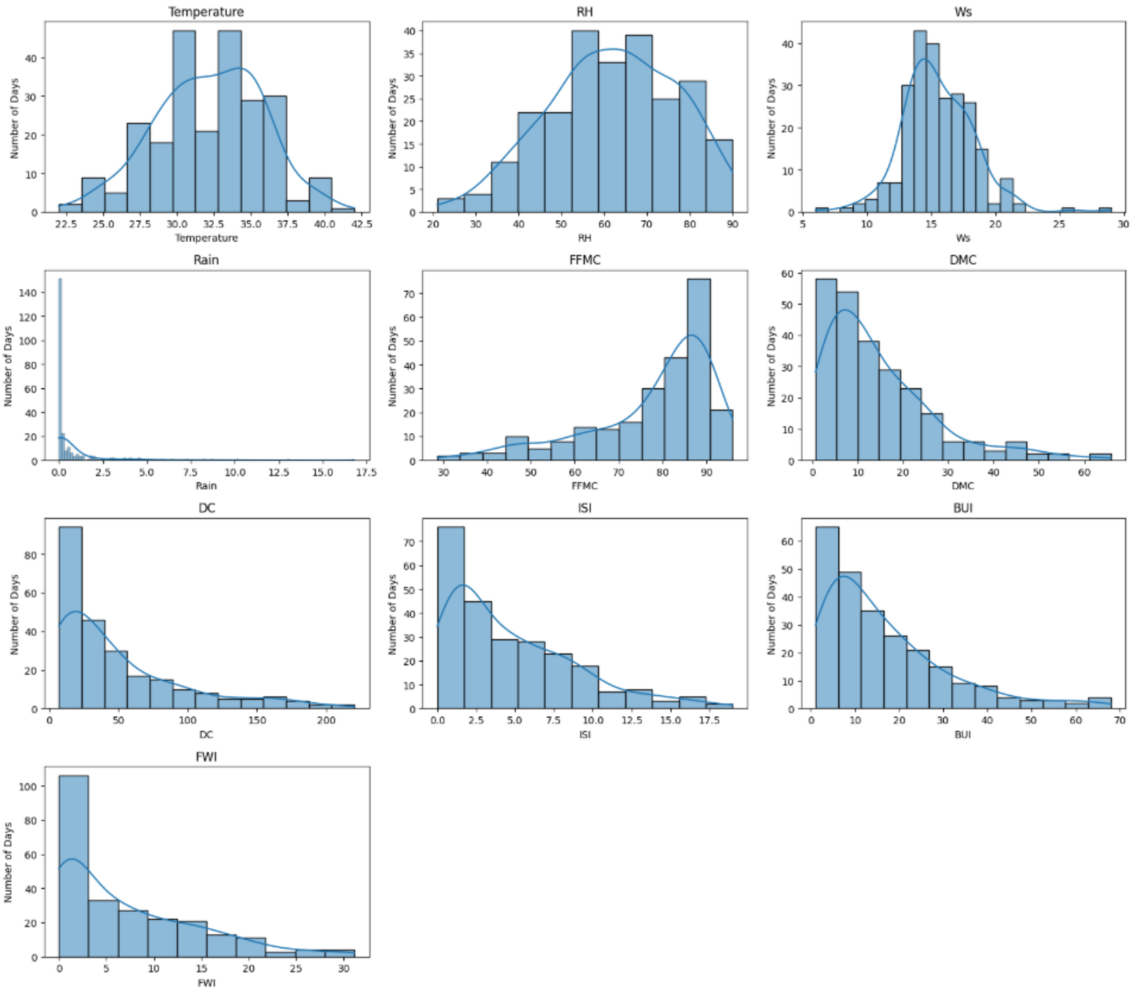


Figure 2

This figure shows histograms with smooth curves for all the main variables in the dataset, such as Temperature, Humidity, Wind Speed, Rainfall, and different fire-risk indexes (FFMC, DMC, DC, ISI, BUI, FWI). These plots help us understand how the data is spread out. Most variables like Rain, DC, ISI, BUI, and FWI are right-skewed, which means high values occur only on a few days. Temperature and RH appear more balanced, showing regular day-to-day patterns. Overall, the figure gives a clear idea of the typical value ranges, variations, and the occasional extreme conditions that can affect fire weather.

- **Scatter plots** give a visual idea of how different weather factors relate to the Fire Weather Index (FWI). They help us understand which variables have stronger or weaker influence on fire risk.

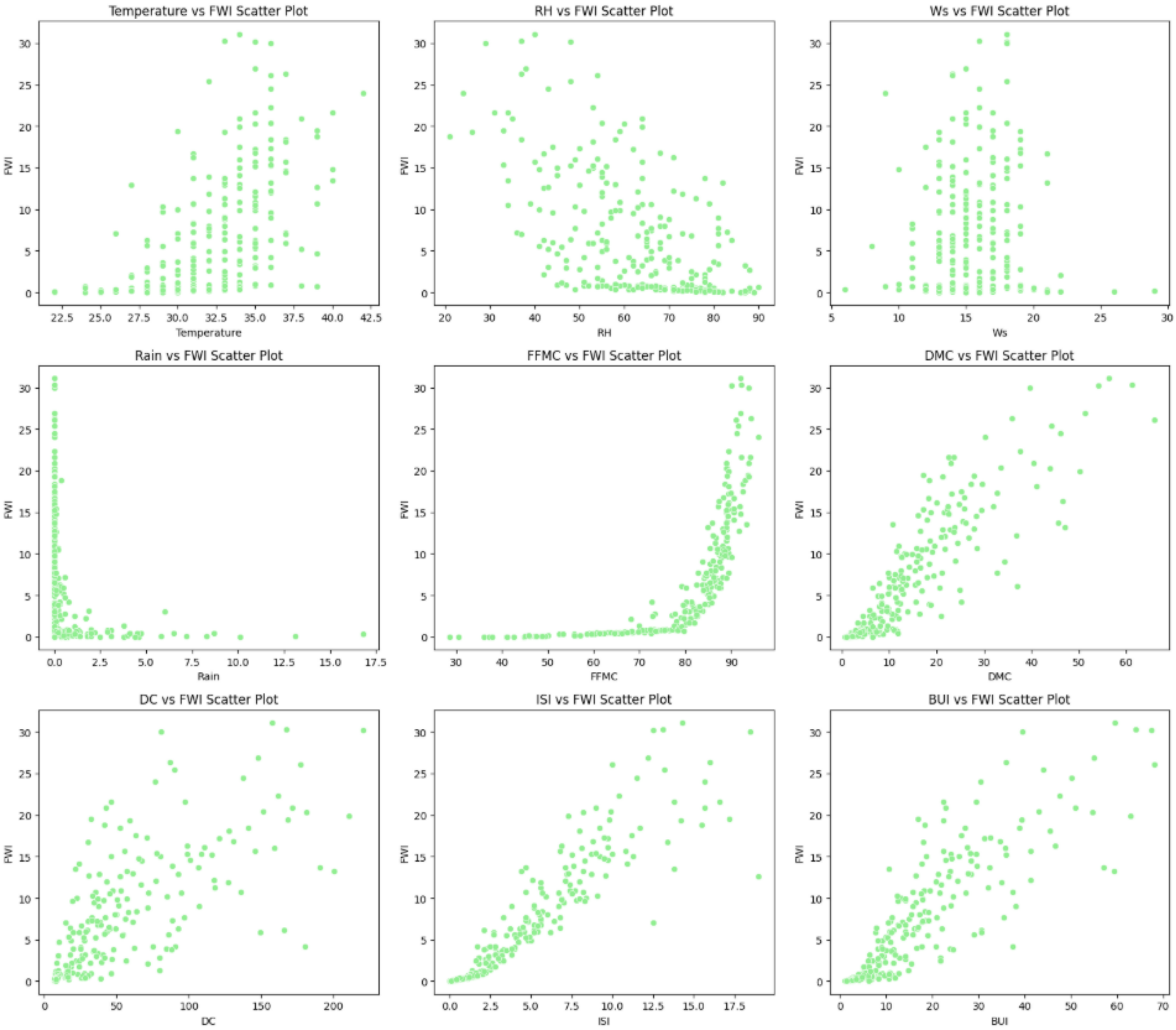
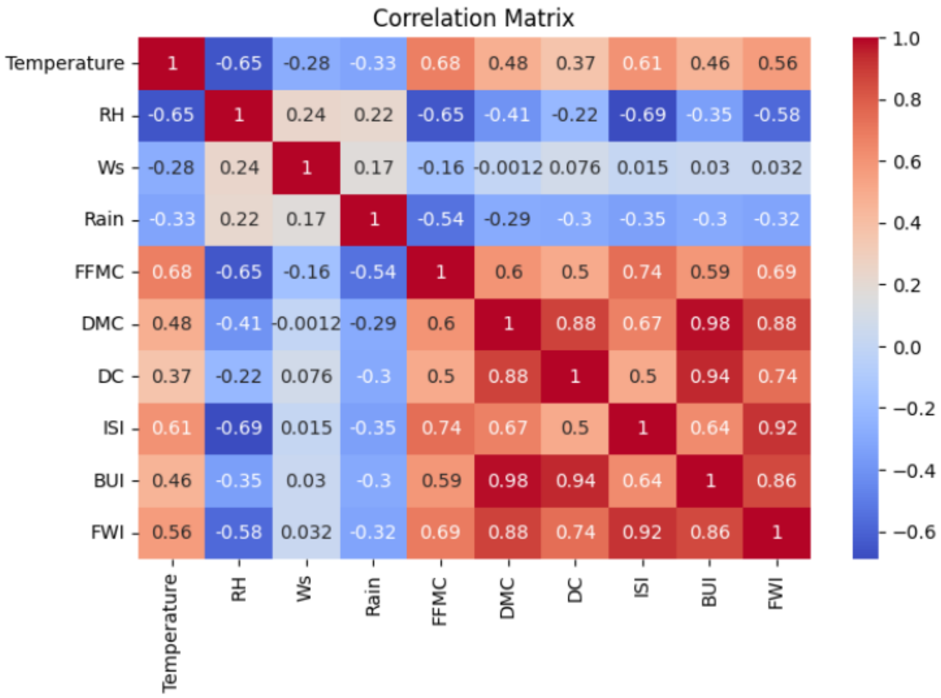


Figure 3

The scatter plots clearly show that temperature has a direct connection with FWI, as higher temperatures usually lead to an increase in fire risk. Relative humidity mostly stays linked with low FWI values, meaning more moisture in the air helps reduce the possibility of fire. Wind speed is spread out but still shows that stronger winds can slightly raise FWI. Rain has the opposite effect, as days with rainfall almost always show very low FWI. The fuel-related indices like FFMC, DMC, DC, ISI, and BUI display a strong upward trend, indicating that when these dryness and fuel buildup values rise, FWI also increases. Overall, the plots make it clear that dry weather, high temperature, low humidity, and accumulated dry fuel together lead to a higher chance of fire.

- **Correlation heatmaps** were generated to study relationships between features and the target variable FWI. Features showing strong correlation with FWI were noted as important for model input.



	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI
Temperature	1	-0.65	-0.28	-0.33	0.68	0.48	0.37	0.61	0.46	0.56
RH	-0.65	1	0.24	0.22	-0.65	-0.41	-0.22	-0.69	-0.35	-0.58
Ws	-0.28	0.24	1	0.17	-0.16	-0.0012	0.076	0.015	0.03	0.032
Rain	-0.33	0.22	0.17	1	-0.54	-0.29	-0.3	-0.35	-0.3	-0.32
FFMC	0.68	-0.65	-0.16	-0.54	1	0.6	0.5	0.74	0.59	0.69
DMC	0.48	-0.41	-0.0012	-0.29	0.6	1	0.88	0.67	0.98	0.88
DC	0.37	-0.22	0.076	-0.3	0.5	0.88	1	0.5	0.94	0.74
ISI	0.61	-0.69	0.015	-0.35	0.74	0.67	0.5	1	0.64	0.92
BUI	0.46	-0.35	0.03	-0.3	0.59	0.98	0.94	0.64	1	0.86
FWI	0.56	-0.58	0.032	-0.32	0.69	0.88	0.74	0.92	0.86	1

Figure 4

This correlation matrix shows how the main weather and fire-risk variables are related to each other. Fire-related indexes like DMC, DC, ISI, BUI, and FWI have strong positive correlations, meaning they increase together. Temperature also shows positive links with some fire indexes. In contrast, Relative Humidity has strong negative correlations with Temperature and several fire-risk variables. Rainfall has weak negative relationships overall. This figure helps identify which factors influence fire-weather conditions the most.

2.4 Preparing the Dataset

After visual exploration, unnecessary columns, such as **Classes**, were dropped since they were not relevant for FWI prediction. All numerical features were checked for consistency and corrected if needed, ensuring that the dataset was clean and formatted correctly for subsequent modeling.

```
df = df.drop('Classes', axis=1)
```

2.5 Saving the Processed Dataset

Finally, the fully cleaned and preprocessed dataset was saved as a new file. This ensured that the dataset could be easily used in the next modules for feature engineering, scaling, and model training.

```
df.to_csv("./datasets/FWI_Dataset_Cleaned.csv", index=False)
```

MODULE 3: FEATURE SELECTION AND SCALING

In this module, we prepared the FWI dataset for model training. Important features like temperature, humidity, wind, rain, and fire-fuel indices were selected as inputs, and FWI was set as the target. The data was split into training and testing sets, and all features were scaled to the same range to make the model training more effective. The scaling method was saved to use later during model deployment.

3.1 Importing Libraries and Loading Cleaned Dataset

First, we imported the necessary Python libraries for data handling, preprocessing, and saving objects. The cleaned Fire Weather Index (FWI) dataset was then loaded into a Pandas DataFrame from a CSV file, making it ready for further processing.

3.2 Feature Selection and Target Variable

Feature selection was performed to improve the accuracy and efficiency of the Fire Weather Index (FWI) prediction model. Initially, non-essential columns such as **Region, day, month, and year** were removed from the dataset, as they do not have a direct numerical impact on FWI. Correlation analysis was then applied to identify features that have a meaningful relationship with the target variable.

A correlation threshold of **0.3** was used to automatically select relevant features, while the target variable **FWI** was excluded to prevent data leakage. In addition to the correlation-based selection, **wind speed (Ws)** was manually included due to its significant role in influencing forest fire behavior. The final selected features were used as input variables (**X**), and **FWI** was taken as the output variable (**y**) for model training and evaluation.

```
correlation = df.corr()['FWI'].sort_values(ascending=False)
correlation_threshold = 0.3
selected_features = correlation[correlation
>correlation_threshold].index.tolist()
selected_features.remove('FWI')
selected_features.append('Ws’)
```

3.3 Splitting the Dataset

The dataset was split into training and testing sets to evaluate the model's performance. 80% of the data was used for training, and 20% was kept aside for testing. A fixed random state was used to ensure that the split can be reproduced in future runs.

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

3.4 Feature Scaling

The dataset was split into training and testing sets to evaluate the model's performance. 80% of the data was used for training, and 20% was kept aside for testing. A fixed random state was used to ensure that the split can be reproduced in future runs.

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

3.5 Saving the Scalar

The fitted StandardScaler was saved using pickle to ensure that the same scaling can be applied to new data during model deployment or future predictions. This helps maintain consistency in the input data for the trained model.

```
with open("scaler.pkl", "wb") as f:
    pickle.dump(scaler, f)
```

MODULE 4: MODEL TRAINING AND EVALUATION

Since the objective of this project is to **predict the Fire Weather Index (FWI)**, which is a continuous numerical value, regression-based machine learning models were selected. The following models were used to compare prediction performance and best model will choose for final predictions.

4.1 Selecting Regression Models

- **Linear Regression**
  Linear Regression was used to model the direct relationship between meteorological features and the Fire Weather Index. It serves as a baseline model for understanding how well simple linear relationships can predict fire risk.
- **Ridge Regression**
  Ridge Regression was chosen to handle multicollinearity among weather and fire-fuel features. By applying regularization, it helps improve prediction accuracy and reduces overfitting.
- **Lasso Regression**
  Lasso Regression was used to identify the most important features affecting FWI by shrinking less important feature coefficients to zero. This helps in simplifying the model while maintaining good performance.
- **ElasticNet Regression**
  ElasticNet Regression combines the strengths of both Ridge and Lasso regression. It is effective when multiple input features are correlated and provides a balanced regularization approach.

```
models = {
    "Linear Regression": LinearRegression(),
    "Ridge Regression": Ridge(alpha=1.0),
    "Lasso Regression": Lasso(alpha=0.01),
    "ElasticNet Regression":ElasticNet(alpha=0.01,l1_ratio=0.5)
}
```

4.2 Model Training

In this step, all selected regression models were trained using the scaled training data. After training, each model was used to predict the Fire Weather Index (FWI) values on the test dataset. This helped in understanding how well each model learns the relationship between the input features and the target variable.

```
results = []
for name, model in models.items():
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
```

4.3 Model Evaluation

After generating predictions, the performance of each regression model was evaluated using standard evaluation metrics. Mean Absolute Error (MAE) was used to measure the average prediction error, Root Mean Squared Error (RMSE) was used to penalize larger errors, and the R^2 score was used to understand how well the model explains the variation in Fire Weather Index values. These metrics helped in comparing the accuracy and reliability of different models.

```
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)
```

Model	**MAE**	**RMSE**	**R2 Score**
Linear Regression	0.500693	0.673129	0.988375
Ridge Regression	0.507991	0.685902	0.987929
Lasso Regression	0.498853	0.678988	0.988172
ElasticNet Regression	0.510224	0.688238	0.987847

4.4 Model Selection and Training

All the regression models were evaluated using MAE, RMSE, and R^2 score to compare their prediction performance. The results showed that Linear Regression, Ridge, Lasso, and ElasticNet achieved very similar accuracy with high R^2 values, indicating strong prediction capability for Fire Weather Index (FWI). Even though some models showed slightly lower error values, Ridge Regression was chosen as the final model because it provides more stable and reliable predictions. It handles correlated input features effectively through regularization, which reduces the chances of overfitting and improves generalization on unseen data.

```
model = Ridge(alpha=1.0)
model.fit(X_train_scaled, y_train)
```

4.5 Saving the Final Model

After selecting Ridge Regression as the final model, it was trained on the scaled training data and saved using the pickle library. Saving the model allows it to be reused later for deployment or future predictions without retraining. This ensures consistent and efficient prediction of Fire Weather Index values.

```
with open("ridge.pkl", "wb") as f:
    pickle.dump(model, f)
```

MODULE 5: MODEL EVALUATION

In this module, different regression models were evaluated to predict the Fire Weather Index (FWI). The main aim of this module is to tune model parameters and compare training and testing performance to understand how well the models generalize to unseen data.

5.1 Alpha Hyperparameter Tuning

Some regression models like **Ridge**, **Lasso**, and **ElasticNet** use a regularization parameter called **alpha (α)**. Alpha controls how much penalty is applied to the model coefficients. If alpha is too small, the model may overfit, and if it is too large, the model may underfit.

To find the best alpha value, **GridSearchCV** was used on the **training dataset only**. Multiple alpha values were tested, and the value that gave the best R^2 score was selected. Linear Regression does not use alpha, so it was trained directly.

5.2 R^2 Score

The **R^2 score** shows how well the model explains the variation in FWI values.

$$R^2 = 1 - \frac{\sum(y - \hat{y})^2}{\sum(y - \bar{y})^2}$$

- R^2 close to **1** means good prediction
- Lower R^2 means poor performance

Both training and testing R^2 scores were calculated to check if the model is overfitting or generalizing well.

```
train_r2 = r2_score(y_train, y_train_pred)
test_r2 = r2_score(y_test, y_test_pred)
```

5.3 Mean Square Error

The **Mean Squared Error (MSE)** measures the average of the squared difference between actual and predicted values.

$$MSE = \frac{1}{n} \sum(y - \hat{y})^2$$

MSE gives more weight to large errors, so it helps in identifying big prediction mistakes.

```
train_mae = mean_absolute_error(y_train, y_train_pred)
test_mae = mean_absolute_error(y_test, y_test_pred)
```

5.4 Root Mean Square Error

The **Root Mean Squared Error (RMSE)** is the square root of MSE and is in the same unit as FWI.

$$RMSE = \sqrt{\frac{1}{n}\sum(y - \hat{y})^2}$$

Lower RMSE values mean the model is making fewer prediction errors.

```
train_rmse = np.sqrt(mean_squared_error(y_train,
y_train_pred))
test_rmse = np.sqrt(mean_squared_error(y_test, y_test_pred))
```

5.5 Model Performance Visualization

Visualization was used to understand how well each regression model predicts the Fire Weather Index (FWI). For every trained model, an **Actual vs Predicted FWI scatter plot** was generated using the **test dataset**.

In these plots, the actual FWI values are shown on the x-axis and the predicted FWI values are shown on the y-axis. A diagonal reference line represents perfect prediction, where the predicted value is equal to the actual value. Data points closer to this line indicate better model performance, while points farther away represent higher prediction errors.

These visualizations help in comparing model behaviour on unseen data and make it easier to identify underfitting or overfitting. Using test data for visualization ensures that the model's real-world predictive ability is properly assessed.

- **Linear Regression**
  The graph shows the comparison between actual and predicted Fire Weather Index (FWI) values using the Linear Regression model. The x-axis represents actual FWI values, while the y-axis represents predicted FWI values, and the diagonal dashed line indicates perfect prediction. Most of the data points lie close to this line, showing that the model predicts FWI accurately on the test data, with only small deviations at higher FWI values. Overall, the graph indicates that the Linear Regression model captures the relationship between input features and FWI effectively.

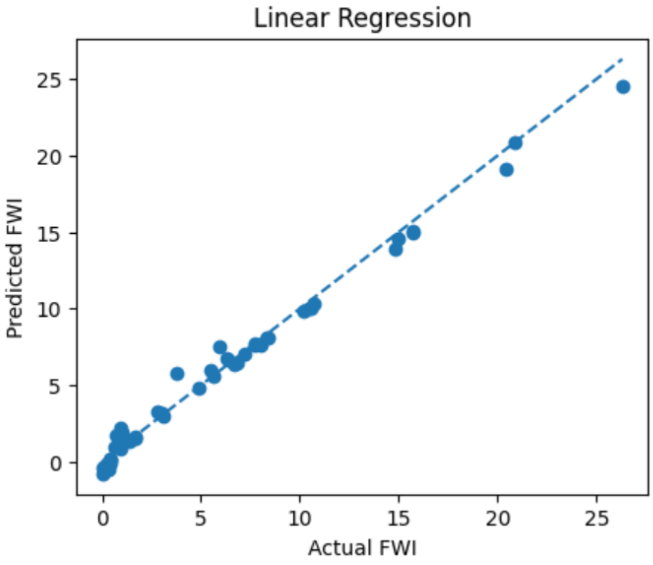


Figure 5

- **Ridge Regression**
  The graph shows the comparison between actual and predicted Fire Weather Index (FWI) values using the Ridge Regression model with α = 0.1. The x-axis represents actual FWI values, and the y-axis represents predicted FWI values, while the dashed diagonal line indicates perfect prediction. Most data points lie very close to this line, showing that Ridge Regression predicts FWI accurately on the test data. Compared to simple Linear Regression, Ridge Regression reduces small deviations at higher FWI values, indicating better stability and improved generalization due to regularization.

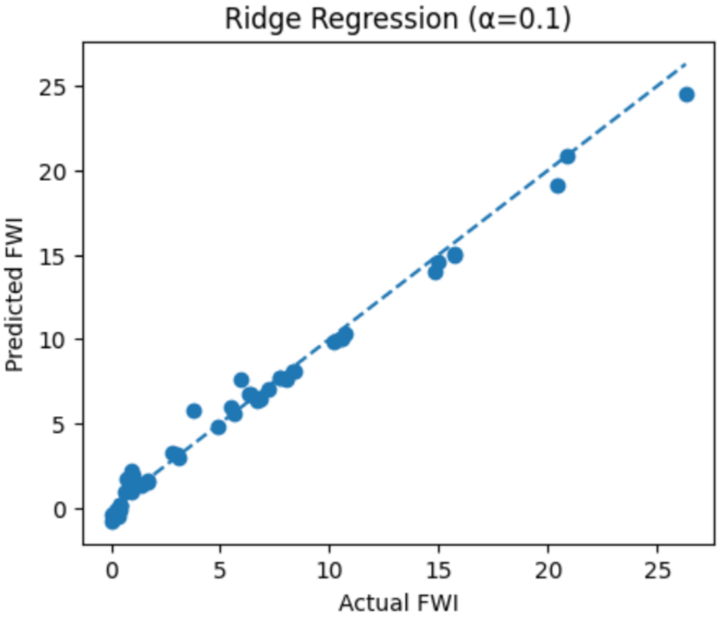


Figure 6

- **Lasso Regression**
  The graph shows the comparison between actual and predicted Fire Weather Index (FWI) values using the Lasso Regression model with α = 0.1. The x-axis represents the actual FWI values and the y-axis represents the predicted FWI values, while the dashed diagonal line indicates perfect prediction. Most points lie close to this line, showing that the model predicts FWI values reasonably well on the test data. Slight deviations at higher FWI values indicate some loss of accuracy due to feature shrinkage, but overall the model maintains good predictive performance while reducing the effect of less important features.

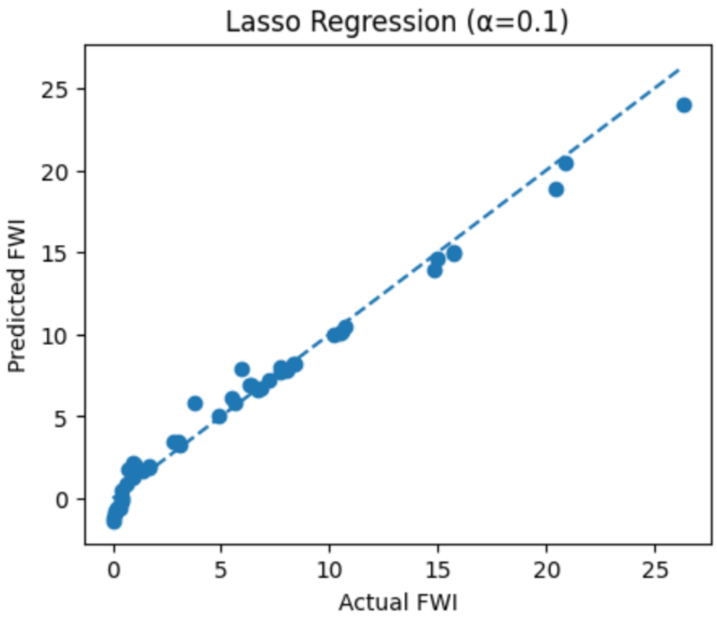


Figure 7

- **ElasticNet Regression**
  The graph shows the comparison between actual and predicted Fire Weather Index (FWI) values using the ElasticNet Regression model with α = 0.001. The x-axis represents the actual FWI values and the y-axis represents the predicted FWI values, while the dashed diagonal line indicates perfect prediction. Most of the data points lie close to this line, showing that ElasticNet predicts FWI values accurately on the test data. By combining both L1 and L2 regularization, the model provides stable predictions with reduced overfitting and good generalization performance.

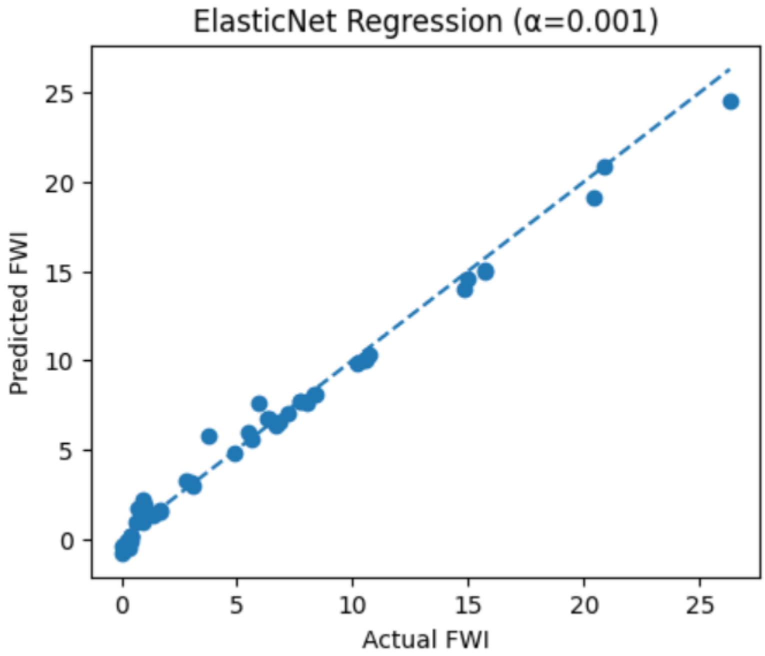


Figure 8

5.6 Training and Testing Performance Comparison

Performance of different regression models using **MAE, RMSE, and R²** for both training and testing data. **Ridge Regression** has low errors and high R², with Train and Test metrics very close, indicating good generalization and stability. **Lasso** shows higher errors due to shrinking some coefficients, while **ElasticNet** performs similar to Ridge but without significant improvement. Overall, Ridge Regression provides the best balance of accuracy and reliability for predicting FWI.

Model	**Train MAE**	**Train RMSE**	**Train R2**	**Test MAE**	**Test RMSE**	**Test R2**
Linear	0.723928	1.366687	0.968093	0.500693	0.673129	0.988375
Ridge	0.723977	1.367057	0.968075	0.499415	0.672402	0.988400
Lasso	0.809400	1.409568	0.966059	0.615034	0.815411	0.982941
ElasticNet	0.724390	1.367351	0.968062	0.499071	0.671670	0.988425

5.7 Best Model - Ridge

A. Handles Multicollinearity and Overfitting

Ridge Regression achieves **high R^2** (0.968 on Train, 0.988 on Test) and **low MAE and RMSE**, indicating accurate predictions across low and high FWI values. The **Actual vs Predicted FWI plot** confirms that Ridge predictions lie close to the diagonal line, further proving its accuracy. These metrics and visualizations make Ridge Regression the most suitable model for predicting the Fire Weather Index in this study.

B. Comparison With Other Regression Models

Unlike Lasso Regression, which has higher errors and can remove important features by shrinking coefficients to zero, and ElasticNet, which does not significantly improve performance here, Ridge retains all relevant features while controlling model complexity. This makes Ridge more reliable and consistent across datasets.

C. Better Performance Metrics and Visualizations

Ridge Regression achieves **high R^2** (0.968 on Train, 0.988 on Test) and **low MAE and RMSE**, indicating accurate predictions across low and high FWI values. The **Actual vs Predicted FWI plot** confirms that Ridge predictions lie close to the diagonal line, further proving its accuracy. These metrics and visualizations make Ridge Regression the most suitable model for predicting the Fire Weather Index in this study.