Tempest FWI Predictor – A Machine Learning Model to Predict Fire Weather Index

Submitted by: Mishti Kalra
Date: 29 November 2025

Week 3 Learnings Summary

In this week I focused on feature engineering, scaling, model training using Ridge Regression, and evaluating performance metrics to develop a stable and optimized Fire Weather Index (FWI) prediction model. I focused on handling **multicollinearity** and implementing **Ridge Regression (L2 Regularization)** to build a stable, accurate, and deployment-ready model for predicting the Fire Weather Index (FWI).

Module 3: Feature Engineering and Feature Scaling

In this module, key preprocessing steps were performed to prepare the dataset for ridge regression while ensuring numerical stability and consistent model behaviour during deployment.

1. Feature Selection Using Correlation Analysis

- Computed the correlation matrix for all input variables against the target variable FWI.
- Selected the most relevant features by identifying strong positive and negative correlations.
- This ensured meaningful features were passed to the model while preserving the real-world relationships among meteorological variables.

2. Scaling Numerical Features

- Applied StandardScaler to all input features to normalize the data.
- Standardization ensures each feature has:
    - Mean = 0
    - Standard deviation = 1
- This step is *mandatory* for regularized models (like Ridge) because without scaling, features with large magnitude would dominate the L2 penalty.

3. Separating Input (X) and Output (y)

- Extracted all selected features into matrix X.
- Set FWI as the target vector y.

4. Train–Test Split

- Split the prepared data into:
    - Training set (80%)
    - Testing set (20%)
- Ensures unbiased evaluation of model performance on unseen data.

5. Saving the Scaler for Deployment

- Exported the trained StandardScaler as scaler.pkl.
- This ensures that during deployment, incoming data is processed with the *exact same scaling parameters*, maintaining inference consistency.

Module 4: Model Training Using Ridge Regression

This module focused on training a robust machine learning model capable of handling multicollinearity, which is common in meteorological datasets.

1. Why Ridge Regression?

- Ridge Regression adds L2 regularization to control large coefficients.
- Especially useful in this dataset because:
    - Fire weather variables (e.g., DMC, ISI) are highly correlated.
    - Ordinary Linear Regression becomes unstable in such cases.
- Ridge stabilizes the solution and improves generalization.

2. Hyperparameter Tuning of Alpha

- The alpha (λ) parameter controls the strength of regularization.
- Tested multiple alpha values to observe:
    - Training MSE
    - Testing MSE
    - Testing RMSE
    - Testing MAE
- Optimal alpha selected based on best test performance and balanced bias-variance tradeoff.

3. Model Training

- Trained a Ridge Regression model using the optimal alpha.
- Ensured convergence and validated the model's behavior through error trends.

4. Saving the Model

- Stored the trained model as ridge.pkl using pickle.
- This serialized model can be integrated into a production pipeline for real-time FWI prediction.

M**odule 5: Model Evaluation and Optimization**

After training, performance metrics were computed to understand the model's predictive quality and identify opportunities for improvement.

1. Mean Absolute Error (MAE)

- Measures average magnitude of errors.
- Useful for interpreting real-world prediction differences.

2. Root Mean Squared Error (RMSE)

- Penalizes large deviations more than MAE.
- Helps evaluate sensitivity to outliers.

3. R^2 Score

- Indicates how much variance in FWI is explained by the model.
- Higher values represent better predictive capability.

4. Predicted vs Actual Plot

- Visualized how closely the model's predictions align with true FWI values.
- A near-diagonal distribution indicates strong predictive accuracy.

5. Hyperparameter Re-tuning

- If performance metrics were suboptimal, alpha was re-tuned.
- The model was retrained until achieving improved MAE, RMSE, and R^2 values.

Conclusion

So, I have successfully implemented a complete and production-ready machine learning pipeline for Fire Weather Index prediction. The workflow included:

- Correlation-based feature selection
- Robust feature scaling
- Ridge Regression training with proper regularization
- Systematic alpha tuning
- Comprehensive metric evaluation
- Saving scaler and model for deployment