Regression Algorithm Comparison

Nevil Patel, 000892482

# Data Description

**Dataset Name:** Algerian Forest Fires Dataset

**Source:** Abid, (2019). Algerian Forest Fires [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5KW4N.

**Dataset URL:** <https://archive.ics.uci.edu/dataset/547/algerian+forest+fires+dataset>

**Description of Features:**

The dataset contains 244 instances of meteorological data from 2 regions of Algeria: Bejaia and Sidi Bel-Abbes located in the northeast and northwest respectively. It includes 13 features such as temperature, humidity, wind speed, and other environmental factors that influence the occurrence of forest fires. The target variable is binary, indicating whether a fire occurred as (1) or not as (0).

**Regression Task:**

The task is to predict the likelihood of a forest fire based on the meteorological features. This is treated as a regression problem, where the target variable is transformed into a continuous value for regression modeling.

**Statistics:**

* Number of features: 13
* Number of Samples: 244
* Training set size: 195 (80%)
* Testing set size: 49(20%)
* Ranges of features:
  + Temperature: 22C to 42C
  + Humidity: 15% to 100%
  + Wind Speed: 6 km/h to 29km/h
* Target variable: Binary (0 or 1)

# The Tests

Explain the tests you did here, very briefly. What kind of cross-validation did you use? For each algorithm, what parameters did you alter and what values of each parameter did you try in your grid search before landing on the best set of parameters?

**Algorithm Tests**:

1. Linear Regression: No Hyperparameters
2. Polynomial Regression:
   1. order = [2, 3, 4, 5]
3. K-Nearest Neighbors (K-NN) Regressor:
   1. n\_neighbours = [3,5,7,9]
   2. weights = [‘uniform’, distance]
   3. p = [1,2]
4. Decision Tree Regressors:
   1. max\_depth = [3, 4, 5, 6]
   2. min\_samples\_split = [2, 5, 10]
   3. criterion = [spuared\_error, friedman\_mse]
5. Random Forest Regressor:
   1. n\_estimators = [50, 100, 200]
   2. max\_depth = [None, 4, 6, 8]
   3. max\_features = [sqrt, log2]
   4. min\_samples\_split = [ 2, 5, 10]
6. Support Vector Regressor (SVR):
   1. kernel = [‘linear’, ‘poly’]
   2. C = [ 0.1, 1, 10, 100]
   3. Gamma = [ ‘scale’, ‘auto’, 0.1, 1]
   4. Epsilon = [0.1, 0.2, 0.5]

# The Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Description | Mean MSE | Min MSE | MAX MSE |
| Linear | N.A. | 0.0826 | 0.0644 | 0.1052 |
| Polynomial | Order = 2 | 0.5900 | 0.0972 | 1.7168 |
| K-NN | n\_neighbors = 3, p = 1, weights = distance | 0.0538 | 0.0395 | 0.0838 |
| Decision Tree | criterion = squared\_error, max\_depth = 3, min\_samples\_split = 10 | 0.0149 | 0.0016 | 0.0408 |
| Random Forest | max\_depth = 8, max\_features = log 2, min\_samples\_split = 5, n\_estimators = 100 | 0.0167 | 0.0033 | 0.0279 |
| Support Vectors | C = 1, gamma = scale, epsilon = 0.2,  kernel = linear | 0.0836 | 0.0644 | 0.1077 |

# Discussion

## Polynomial and Linear Regression

**Performance Comparison:** Linear Regression (order = 1) performed significantly better, achieving an average MSE of 0.0826, while Polynomial Regression (degree = 2) had a much higher MSE of 0.5900.

**Optimal Order and Overfitting:** Even though degree = 2 was selected during tuning, it still underperformed. Higher-degree polynomials (3–5) worsened the issue, leading to overfitting. For instance, degree = 2 reached a max MSE of 1.7168, likely due to outliers or noise.

**Key Takeaway:** The data doesn’t exhibit a strong polynomial relationship with fire risk. While Linear Regression captures some correlations—such as higher temperatures increasing fire likelihood—it’s clear that more sophisticated non-linear models would be more effective.

## K-NN Regression

## **Best Configuration:** The optimal model used k = 3, p = 1 (Manhattan distance), and weights = 'distance'.

## **Why It Performed Well:** Lower values of k (around 3–5) helped in identifying local patterns without excessive smoothing. Additionally, Manhattan distance (p = 1) outperformed Euclidean distance (p = 2), possibly because it’s less sensitive to outliers in features like wind speed.

## **Data Insights:** The model achieved a low MSE of 0.0538, suggesting that fire risk is influenced by localized weather patterns, where clusters of high temperatures and low humidity play a major role.

## Decision Tree Regression

## **Best Parameters:** The optimal setup involved max\_depth = 4, min\_samples\_split = 10, and criterion = 'squared\_error'.

**Why It Excelled:** Limiting tree depth to 4 while requiring at least 10 samples per split helped balance interpretability and performance, preventing overfitting.

**Key Insight:** Decision Trees achieved the lowest MSE of 0.0152, indicating that fire risk follows a rule-based, hierarchical structure—for example, "If temperature > X and humidity < Y, then fire risk is high." The model effectively captured threshold-based relationships in key features like FFMC (Fine Fuel Moisture Code).

## Random Forest Regression

## **Best Configuration:** The most effective model used n\_estimators = 100, max\_depth = 8, and max\_features = 'log2'.

## **Why It Worked:** The ensemble of 100 trees, with controlled depth and feature selection (log2), helped reduce overfitting while maintaining predictive power.

## **Comparison to Decision Tree:** The MSE was slightly higher at 0.0172, suggesting that while Random Forest reduces overfitting, it might also oversimplify certain rules that a single Decision Tree captures more precisely. This implies limited feature interactions in the dataset.

## Support Vector Regression

# Best Parameters: A linear kernel with default C = 1 performed best.

# Why It Struggled: Unlike tree-based models, SVR assumes a smoother relationship between features and fire risk. The linear kernel outperformed RBF and polynomial kernels, hinting at some linear patterns, but overall, fire risk relationships are more complex.

# Performance Review: The model’s MSE was 0.0836, confirming that SVR is not well-suited for capturing the threshold-based, non-linear nature of fire risk.

# Final Recommendation

Among all models tested, Decision Tree Regression clearly outperformed the rest, achieving the lowest MSE (0.0152).

**Why It Works Best:**

The dataset follows hierarchical, rule-based relationships rather than smooth linear or polynomial patterns.

Decision Trees naturally capture critical environmental thresholds affecting fire risk, such as specific temperature and humidity levels.

**Why Other Models Fell Short**

Linear & SVR: Assumed a linear trend, which isn’t sufficient for fire risk prediction.

Polynomial Regression: Overfit to noise without improving accuracy.

K-NN: Worked well but struggled with irrelevant features.

Random Forest: Regularization helped, but it slightly oversimplified key patterns.

**Conclusion**

For fire risk prediction, a Decision Tree Regressor is the best choice. It offers the highest accuracy, is easy to interpret, and runs efficiently—making it the ideal model for deployment.