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:
    a. order = [2, 3, 4, 5]
3. K-Nearest Neighbors (K-NN) Regressor:
    a. n_neighbours = [3,5,7,9]
    b. weights = ['uniform', distance]
    c. p = [1,2]
4. Decision Tree Regressors:
    a. max_depth = [3, 4, 5, 6]
    b. min_samples_split = [2, 5, 10]
    c. criterion = [squared_error, friedman_mse]
5. Random Forest Regressor:
    a. n_estimators = [50, 100, 200]
    b. max_depth = [None, 4, 6, 8]
    c. max_features = [sqrt, log2]
    d. min_samples_split = [2, 5, 10]
6. Support Vector Regressor (SVR):
    a. kernel = ['linear', 'rbf', 'poly']
    b. C = [0.1, 1, 10, 100]
    c. Gamma = ['scale', 'auto', 0.1, 1]
    d. 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 = log2, min_samples_split = 5, n_estimators = 100	0.0167	0.0033	0.0279
SUPPORT VECTORS	C = 1, gamma = scale, epsilon = 0.1, kernel = rbf	0.0602	0.0497	0.0702

Discussion

Polynomial and Linear Regression

Best Order:

- Linear Regression (order=1) outperformed Polynomial Regression (best: order=2).

Why Order=2 Was Optimal:

- Higher degrees (3–5) caused severe overfitting and lacking consistent predictions (max MSE=1.71).
- Order=2 slightly improved over Linear Regression in some folds but averaged 7× worse MSE (0.59 vs. 0.08).

Data Insight:

- The data doesn’t show a strong polynomial relationship with fire risk. While Linear Regression captures some correlations for features such as higher temperatures, wind speed etc. for increasing fire likelihood, it’s helping to make clear and effective predictions using sophisticated non-linear models.

K-NN Regression

Best Parameters:

- 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.
- Manhattan distance (p = 1) outperformed Euclidean distance (p = 2), because it’s handles outliers (like wind speed variations) better.

Winner:

- Small k=3 focused on local patterns (e.g., heatwaves).
- Manhattan distance (p=1) ignored outlier noise better than Euclidean.
- Weighted voting prioritized nearby neighbors which MSE is 0.0538.

Loser:

- **Large k values** (7–9), has on an average too many neighbors and missing local trends (MSE=0.061).
- It treat all the neighbors equally, even distant ones (MSE=0.059).
- Euclidean Distance is overreacted to outliers like rare wind gusts (MSE=0.057).

Note: All the MSE values are retrieved while tuning hyperparameters.

Data Insights: The model achieved a low MSE of 0.0538, suggesting that fire risk is influenced by localized weather patterns, such as clusters of high temperatures and low humidity.

Decision Tree Regression

Best Configuration: The optimal setup used max_depth = 4, min_samples_split = 10, and criterion = 'squared_error'.

Why It Performed Well: Keeping the tree depth at 4 and requiring at least **10 samples per split** helped strike a balance between accuracy and avoiding overfitting.

Winner

- IT has limited depth to 3 splits (e.g., "Is temp > 35°C? → Yes/No").
- It avoids splits on tiny groups (it needs 10+ samples to split).
- It has Achieved lowest MSE (0.0149).

Losers:

- Deep trees of depth 6, overfit to noise (MSE=0.0219).
- Created rules like "If temp=34.2°C AND humidity=47%...", which are too specific and doesn't help to predict.
- Also because of small min_samples_split (2) split on tiny/noisy groups (MSE=0.018).

Note: All the MSE values are retrieved while tuning hyperparameters.

Data Insight: Decision Trees achieved the **lowest MSE** of **0.0152**, indicating that fire risk follows a rule-based, 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 Performed Well: Using an ensemble of 100 trees, with controlled depth and feature selection, helped **reduce overfitting** while maintaining strong appropriate prediction.

Winner:

- 100 trees reduced overfitting while capturing appropriate trends.
- Deeper trees such as depth=8 found more patterns than single trees.
- log2 features improved diversity among trees to achieve MSE=0.0167.

Losers:

- Shallow trees of depth 4, miss key thresholds (MSE=0.0201).
- Few trees such as 50, has less reliable predictions (MSE=0.019).
- sqrt features has limited features diversity (MSE=0.0184).

Note: All the MSE values are retrieved while tuning hyperparameters.

Comparison to Decision Tree: The MSE was slightly higher at **0.0172**, suggesting that while Random Forest helps with generalization, it may oversimplify some relationships that a single Decision Tree captures more accurately, that's why Random Forest reduces overfitting.

Support Vector Regression

Best Configuration: kernel= cbf, C = 1, gamma = 'scale', epsilon = '0.1' performed best.

Why It Struggled: Unlike tree-based models, **SVR assumes a smoother relationship** between features and fire risk. The linear kernel outperformed Random Forest Regression and polynomial kernels, but overall, fire risk relationships appear to be more complex relationship.

Winner:

- RBF kernel detected non-linear patterns (MSE=0.0602).
- gamma=scale adjusted to mixed feature scales (e.g., temperature vs. wind).

Losers:

- Linear kernel is assumed the fire risk was a straight-line trend (MES=0.0836).
- Poly kernel overfit to the noise of the data (MSE=0.12).
- High C=100, overfit the rare possibilities in the dataset (MSE=0.683).

Note: All the MSE values are retrieved while tuning hyperparameters.

Performance Review: SVR had an **MSE of 0.0836**, confirming that it's **not ideal** for capturing fire risk, which follows more **threshold-based, non-linear** patterns.

Final Recommendation

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

Why Decision Tree Work Best:

- The Algerian Forest Fires 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 temperature and humidity levels.

Why Other Models Not Performed Well:

- **Linear & SVR**: It assume the dataset as a linear trend, which isn't sufficient for fire risk prediction.
- **Polynomial Regression**: It overfit to noise without improving accuracy.
- **K-NN**: It somewhat worked well but it is struggling with irrelevant features and outliers.
- **Random Forest**: It reduced overfitting but 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.