Predicting the Burned Area of Forest Fires

Noah Vlodek

December 2024

Contents

1 Introduction **3**
- 1.1 Why Portugal? . . . 3
- 1.2 Wildfires Impact on World . . . 3

2 Forest Fire Dataset **4**
- 2.1 Data Preprocessing . . . 6

3 Method 1: Random Forest **6**
- 3.1 What are Random Forests? . . . 6
- 3.2 Running the Algorithm . . . 7
  - 3.2.1 Splitting the Data . . . 7
  - 3.2.2 Parameter Choice . . . 7
- 3.3 Final Results . . . 8

4 Method 2: Gradient Boosting **9**
- 4.1 What is Gradient Boosting? . . . 9
- 4.2 Parameter Choice . . . 9
- 4.3 Results . . . 9

5 Method 3: Adaboost **10**
- 5.1 What is Adaboost? . . . 10
- 5.2 Parameters . . . 11
- 5.3 Results . . . 11

6 Conclusions **11**
- 6.1 Comparison of Models . . . 11
  - 6.1.1 Random Forest . . . 12
  - 6.1.2 Gradient Boosting . . . 12
  - 6.1.3 AdaBoost Regression . . . 13
  - 6.1.4 Observations . . . 13
- 6.2 Final Reflections . . . 13

7 Works Cited **15**

List of Tables

1 Columns in the Dataset 5
2 Metrics, Equations, and Best Results 8
3 Metrics for Random Forest Model 9
4 Metrics for Gradient Boosting Model 10
5 Metrics for AdaBoost Model 11

List of Figures

1 Forest Fire Damage, September 2024 3
2 Structure of the FWI 4
3 One-Hot Encoding 6
4 Min-Max Scaler 6
5 Decision Tree 7
6 Final Random Forest Model 8
7 Gradient Boosting Algorithm 10
8 Adaboost Algorithm 10
9 Adaboost Regression 11

1 Introduction

The basic idea throughout this project is to try different machine learning algorithms to predict the burned area of forest fires in Portugal. With the rise of global climate change, forest fires are rising in frequency, leading to some serious consequences throughout the world. The specific data set I am going to be looking at for this project is Forest Fires dataset, available using the UCI Machine Learning Repository(1).

Code `https://colab.research.google.com/drive/1Uuaq-KeDfb2QpKNYoO6w5jl6vBy7d8-m?usp=sharing`

1.1 Why Portugal?

The data set for my project is based in Montesinho Natural Park, a national park in Northern Portugal. Portugal, like the rest of the world, has been dealing with an increase in forest fires over the past few years. Most recently, in late September, the community of Albergaria-a-Velha faced a massive wildfire. According to Sam Jones of the Guardian, these fires destroyed homes and cars and left behind puddles of black and acidic water (Jones)(2). The damage to these homes were unconscionable for many of the residents here. Below is an image to exemplify the serious damage that these wildfires cause.

Figure 1: Forest Fire Damage, September 2024

1.2 Wildfires Impact on World

According to the EPA, wildfires have the potential to harm property and human health (Environmental Protection Agency)(3). These fire-related threats are bigger for citizens that live near forests or grasslands. Also, of the 22 wildfires in the USA, between 1980 and 2023, that have caused serious damage, "18 of them have occurred since 2000 (EPA)." These statistics show the importance of this issue: these fires are becoming more and more frequent, due to climate change. The most prominent places where wildfires occur in the United States is on the west coast, like California. This topic is very important now more

than ever. As recently as December 11, 2024, according to Daniel Cole of The Guardian, there was a fire in Malibu, California (4). Because of this fire, 6300 citizens were evacuated from their homes. This is just one of the many examples of how wildfires impact communities.

As outlined above, these forest fires can have negative effects, including economical and ecological damage, as well as human suffering, as observed above. The main goal of this project is to predict the burned area of forest fires in Montesinho Natural Park, in northern Portugal. I will also show how this phenomenon is impacting the rest of the world.

2 Forest Fire Dataset

The dataset uses the Canadian Forest Fire Weather Index (FWI) System (CWFIS) (5), which consists of six components that account for the effects of fuel moisture and weather conditions on fire behavior. These components can be divided into two groups: **Fuel Moisture Codes** and **Fire Behavior Indices**.

- **Fuel Moisture Codes**: numeric ratings of the moisture content of the forest floor and other dead organic matter.
- **Fire Behavior Indices**: represent the rate of fire spread.

Here is a diagram that explains the FWI System:

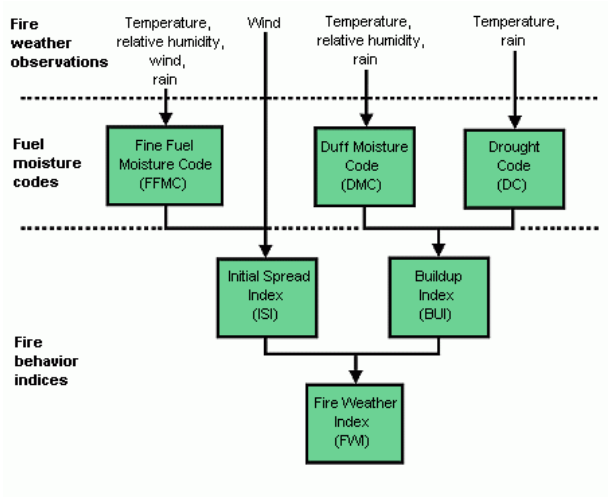

Figure 2: Structure of the FWI

Here is a quick rundown of the definitions of the columns for the data set:

Column	**Definition**
X	x-axis spatial coordinate within Montesino Park
Y	y-axis spatial coordinate within Montesino Park
month	Month of the year [Jan -¿ Dec]
day	Day of the week [Sun -¿ Sat]
FFMC (Fine Fuel Moisture Code)	Numeric rating of moisture content of litter and other cured fine fuels
DMC (Duff Moisture Code)	Numeric rating of the average moisture content of loosely compacted organic layers of moderate depth
DC (Drought Code)	Numeric rating of the average moisture content of deep, compact organic layers
ISI (Initial Spread Index)	Numeric rating of the expected rate of fire spread
temp	Temperature in degrees Celsius
RH (Relative Humidity)	Amount of water vapor present in air (as a percentage)
Wind Speed	Speed of wind in kilometers per hour (km/h)
Rain	Rain in mm/m^2
Area (*Target*)	Burned area of Montesinho Natural Park

Table 1: Columns in the Dataset

2.1 Data Preprocessing

The first task I completed in the pre-processing phase was to look for missing values in the data set: I was relieved to see that there was no data missing in the dataset. The main processing task in this project was encoding the categorical variables. Do encode the month and day columns, I used one-hot encoding, which converts all the categories in the column into their own binary column (0 for false, 1 for true).

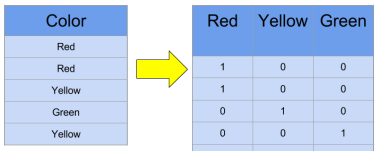

Figure 3: One-Hot Encoding

The next step in processing was to normalize the data. For simplicity, I used min-max scaling to scale all the numerical features:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Figure 4: Min-Max Scaler

Now, I am ready to run models to predict the burned area of a forest fire.

3 Method 1: Random Forest

3.1 What are Random Forests?

A random forest is a machine learning algorithm consisting of multiple *decision trees* to make classifications and predictions. A decision tree is basically a visual representation of all potential outcomes and/or consequences.

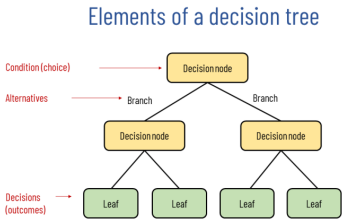

Figure 5: Decision Tree

3.2 Running the Algorithm

3.2.1 Splitting the Data

The first step to any machine learning algorithm is to split the data, and I chose to allocate 80% of the data for training, and the remaining 20% for testing. Splitting of the data is standard for machine learning processes.

3.2.2 Parameter Choice

For the parameters, I chose **1500 estimators**, which means there are 1500 trees in the forest, and I decided to use bootstrap sampling as well, which means that I drew samples from the data set and put them back in the data set before sampling again. I set the criterion to Friedman's'mse', which uses the mean squared error with Friedman's score for potential splits. The goal is to minimize the mean squared error at each stage (or level in the tree):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad \text{where} \quad y_i = \text{true value}, \quad \hat{y}_i = \text{predicted value}. \tag{1}$$

Predicting on Test Data After training the model on the training data, it was now time for me to use my model to predict on the testing data. In the next section, I will summarize my final results.

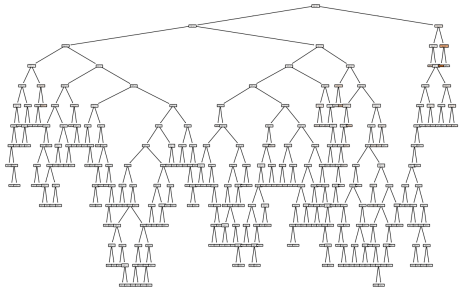

Figure 6: Final Random Forest Model

3.3 Final Results

Metrics to Evaluate Models For all the models that I ran, I used (1) Mean Absolute Error (MAE), (2) R-squared (R^2), (3) Mean Squared Error (MSE), and (4) Root Mean Squared Error (RMSE).

- **MAE**: measures the average magnitude of error between predicted and actual values.
- R^2: represents the proportion of the variance for a dependent variable that is explained by variables in a regression model.
- **MSE**: The average square difference between the observed value and the predicted values by the model.
- **RMSE**: Square root of MSE.

Metric	**Equation**	**Best Result (Goal)**
MAE	$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$	**close to 0**
R^2	$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$	**closer to 1, values range from 0-1**
MSE	$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	Lower
RMSE	$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	Lower

Table 2: Metrics, Equations, and Best Results

Metrics for the Random Forest Model Overall, when I ran the model on the normalized data, the values were actually pretty good. The only concerning metric I got was the R^2, because the value was pretty close to 0, meaning that the model did not pick up the variance in the target variable, Area.

Metric	Value
MAE	0.0248
R^2	0.0124
MSE	0.0098
RMSE	0.098

Table 3: Metrics for Random Forest Model

4 Method 2: Gradient Boosting

4.1 What is Gradient Boosting?

Gradient boosting is a machine learning model that is characterized as an ensemble model. Instead of trying to learn from the data independently, boosting combines the predictions of multiple weaker learners into a single stronger learner (Data Camp Gradient Boosting)(6). This algorithm is slightly different from the previous model, Random Forest: In the random forest algorithm, each decision tree is built independently. In gradient boosting, each new tree attempts to correct the errors in the previous trees.

4.2 Parameter Choice

Parameters To find the best parameters for this model, I used the GridSearchCV function avaliable from the sklearn library in Python. This built-in function is helpful because it searches a list of parameters and finds the optimal choice. For this model, I found the optimal learning rate from multiple choices: [0.01, 0.05,0.1, 0.2, and 0.3]. I used the optimal learning rate in my Gradient Boosting algorithm. Following are the results.

4.3 Results

After running the Gradient Boosting algorithm, my metrics were a little better. What is surprising to me is that the metrics did not change much at all, compared to the random forest model. They slightly improved, but there was not a huge jump. The R^2 value is arguably worse, because the closer it is to 1, the better the value is. So, this model is still not the best. Here are the metrics and the tree for the final model:

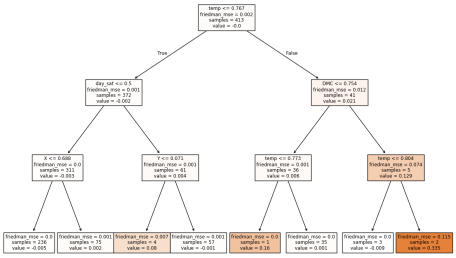

Figure 7: Gradient Boosting Algorithm

Metric	Value
MAE	0.0233
R^2	0.0008
MSE	0.0995
RMSE	0.098

Table 4: Metrics for Gradient Boosting Model

5 Method 3: Adaboost

5.1 What is Adaboost?

Adaboost is a boosting technique that aims at combining weak models into one strong classifier. According to Vihar Kurama, a single model may not be able to accurately predict a final outcome, but many models together may result in a better result (Kurama) (7). Adaboost could be described as a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error by the previous model (SKLearn)(8).

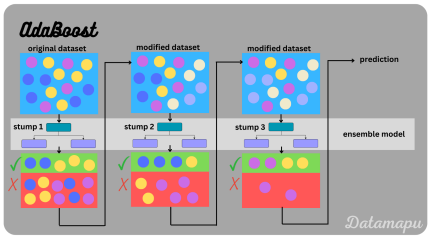

Figure 8: Adaboost Algorithm

5.2 Parameters

For the AdaBoost algorithm, I chose to do a decision tree as the base classifier. I used DecisionRegressor avaliable in sklearn to implement the algorithm. Another function I used is GridSearchCV, where I picked the best parameter for n-estimators from a list of options. I choose 50-950 estimators. This function helps pick the best estimator for the model, which is very important.

5.3 Results

The results ended up not changing that much: the Mean Squared Error and Root Mean Square error decreased a little bit. The R^2 value is still not great, but it did improve a little bit. Here are the final metrics for the AdaBoost algorithm:

Metric	Value
MAE	0.0265
R^2	0.0515
MSE	0.094
RMSE	0.0969

Table 5: Metrics for AdaBoost Model

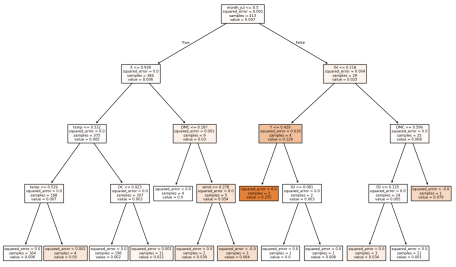

Figure 9: Adaboost Regression

6 Conclusions

6.1 Comparison of Models

The models I chose to implement in this project were: (1) Random Forest Regression, (2) Gradient Boosting Regression, and (3) Adaboost Regression.

6.1.1 Random Forest

Performance

- MAE: 0.0248
- R^2: 0.0124
- MSE: 0.0098
- RMSE: 0.098

Strengths

- Random forest performed decently in terms of MAE and RMSE, indicating that the model minimized the prediction errors.
- The model reduces over fitting, which is a big strength in machine learning projects.

Weaknesses

- The R^2 value suggests that this model struggles to explain the variance in the target variable, but this could be due to the distribution of the target, area.

6.1.2 Gradient Boosting

Performance

- MAE: 0.0233
- R^2: 0.0008
- MSE: 0.0995
- RMSE: 0.098

Strengths

- Slightly better MAE than Random Forest, indicating that the predictions were more accurate on average.
- Gradient Boosting improves iteratively, learning from previous errors, which can be helpful when trying to find complex patterns in data.

Weaknesses

- The R^2 value decreased further, indicating that this model is less likely to capture the underlying variance in the data.
- Computationally more intensive than Random Forest

6.1.3 AdaBoost Regression

Performance

- MAE: 0.0265
- R^2: 0.0515
- MSE: 0.0094
- RMSE: 0.0969

Strengths

- Delivered the highest R^2 value among all the models, indicating that the Adaboost handles the variance the best.
- The MSE and RMSE were also the lowest, among the models.

Weaknesses

- MAE was slightly higher for this model, compared to the other models.
- The Ada-boost algorithm is susceptible to noisy data, which could hinder consistency in the performance of the model.

6.1.4 Observations

- Adaboost performed the best in terms of the R^2, MSE, and RMSE metrics, suggesting that this model was the best for the dataset.
- Random Forest and Gradient Boosting were comparable in MAE and RMSE, with Gradient Boosting outperforming Random Forest in MAE. Both of these models had very low R^2 scores.
- **Biggest Challenge:** All three models struggled to achieve a high R^2 score, which implies the complexity of the data set and the potential need for feature engineering.

6.2 Final Reflections

Here is a summary of the techniques I used in this project:

- **Data Cleaning:** This project required me to perform three fundamental preprocessing tasks:

  Encoding Categorical Features: To encode the categorical features, I used one hot encoding. This process creates a new binary column for each category in a given column in a data set.

  Checking for Missing Data: Fortunately for this dataset, there were no missing features, but one always must check to be sure.

Data Normalization: To get all the data on the same scale, I used the Min-Max scaler, which converts all data points, in a continuous column, to numbers between 0 and 1, 0 being the minimum and 1 being the maximum.

- **Splitting the Data:** For this project, I split the data into training and testing data, using a split of 80-20. This is an imperative technique in machine learning because it is important to evaluate a model's performance on unseen data to prevent overfitting. If we don't have a model that can perform well on unseen data, the model is useless.

- **Tuning the Models** I had to tune each model's hyper parameters to make sure I ended with the best results. For example, if the R^2 value was negative (rather poor), I would update the parameters for the model until I get a decent score.

- **Data Visualization** I learned to graph decision trees and interpret each of the models.

7 Works Cited

References

[1] UC Irvine Machine Learning Repository, "Forest fires." Accessed: September 26, 2024. Dataset for regression tasks.

[2] S. Jones, "'absolute chaos': counting the cost of a deadly wildfire in northern portugal," 2024.

[3] U.S. Environmental Protection Agency, "Climate change indicators: Wildfires," 2024. Accessed: December 12, 2024.

[4] D. Cole, "Stubborn wildfire destroys several structures in malibu, california," 2024. Accessed: December 12, 2024.

[5] Canadian Wildland Fire Information System, "Fire weather index (fwi) system summary," 2024. Accessed: October 20, 2024.

[6] DataCamp Team, "A guide to the gradient boosting algorithm," 2024. Accessed: December 12, 2024.

[7] V. Kurama, "Understanding adaboost and its optimizer," 2024. Accessed: December 12, 2024.

[8] Scikit-learn Developers, "sklearn.ensemble.adaboostregressor," 2024. Accessed: December 12, 2024.