Predicting the burned area in forest fires

Final Project Report

Pietro Dellino
Centrale Paris
2nd year
pietro.dellino@student.ecp.fr

Lucie Donne
Centrale Paris
2nd year
lucie.donne@student.ecp.fr

Valentin Gonsolin
Centrale Paris
2nd year
valentin.gonsolin@student.ecp.fr

Baptiste Marvaldi
Centrale Paris
2nd year
baptiste.marvaldi@student.ecp.fr

ABSTRACT

Forest fires are huge threats to both human lives and a country economy as well as ecological disasters, and disastrous ones keep happening in the recent years. To counter them, our reaction-time has to be minimized, and it can be done through allocating fire forces in the right place at the right time which is done by knowing the prioritized targets for firefighting. In order to do so, we have to be able to predict the forest fire risk of a certain location. In this work, we use a Machine Learning approach to predict the burned area of forest fires. Four different classification techniques and four different regression techniques were tested on data collected from a natural park of Portugal, subject to forest fires. Classification was done in order to predict whether or not a fire would start, and regression was done to estimate the burned area. The best configuration found was Bayes estimator with all features for the classification task, and Support Vector Machines with few features selected for the regression task (which corresponds to the results of previous works on this dataset). However, the complexity of the task yields to accuracy that may not be satisfying enough. A future work would focus on improving this accuracy by working with a dataset containing relevant features which weren't in this one.

KEYWORDS

Forest fire, Regression, Classification, Nearest neighbors, SVMs

1 Introduction – Motivation

Forest fires are environmental and economical catastrophes and moreover can be extremely dangerous for the population. Each year, 60 000 forest fires happen in Europe and in the world, the total surface hit by these phenomena represents 350 million hectares, that's 6 times the surface of France. Last year in 2017, the largest fireforest in Portugal's history happened, leaving behind 66 dead and 204 injured, as well as 45 000 ha of pure devastation.

That's why it's really important to be able to prevent these disasters, by determining the places more likely to be hit by forest fires and also by knowing how spread a forest fire can become when it starts. By doing so, we can settle firefighter units near risky areas and forecast the human and logistical resources needed for a given forest fire. We want to concentrate our study on Portugal, one of the country the most affected by forest fires in the recent years.

Hence our goal here is to predict the total surface that could be burnt by fire in the Montesinho natural park, northeast region of Portugal, by using the tools of Machine Learning, in order to predict the means needed to stop its expansion, taking into account various parameters of the surrounding environment such as meteorological conditions.

Our work here concentrates on a given dataset, recording data about forest fires in the Mintesinho natural park of the northeast region of Portugal in the last few years. For an actual fire, the total burned area of the fire was recorded, as well as meteorological conditions at the moment of the fire and the 16 hours before, all of this summarized in meteorological indexes FFMC, DMC, DC, ISI.

This dataset is known as a difficult regression task; hence we aim to compare different Data Mining algorithms and find one with a satisfying performance overall. To make this project an interesting learning project, we divided the work in two tasks: On the one hand a classification task, predicting whether for given conditions a fire is likely to happen; and on the other hand, a regression task, predicting the burned area in the case a fire is happening.

2 Problem definition

2.1 Description of the dataset

Our aim is to find a regression function between the burned area and the dataset information:

- Position X
- Position Y
- Month
- Day of the week
- FFMC (Fine Fire Moisture Code): a numeric rating of the moisture content of litter and other cured fine fuels. Indicator of the relative ease of ignition and the flammability of fine fuel.
- DMC (Duff Moisture Code): a numeric rating of the average moisture content of loosely compacted organic layers of moderate depth. Indication of fuel consumption in moderate duff layers and medium-size woody material.
- DC (Drought Code): a numeric rating of the average moisture content of deep, compact organic layers. Useful indicator of seasonal drought effects on forest fuels and the amount of smoldering in deep duff layers and large logs.
- ISI (Initial Spread Index) (ISI): a numeric rating of the expected rate of fire spread. It combines the effects of wind and the FFMC on rate of spread without the influence of variable quantities of fuel.
- Temperature (°C)
- Relative humidity (%)
- Wind speed (km/h)
- Rain (mm/m^2)
- Burned area (ha)

2.2 Formal definitions

The dataset is made of $N = 517$ input vectors. For $k \in [\![1, N]\!]$, the input vector x^k has $A = 12$ components:

$$\boldsymbol{x}^k = (x_1^k, \dots, x_A^k)$$

$X \in \mathbb{R}^{N \times A}$ is the dataset matrix: its lines are the input vectors.

The thirteenth column of the dataset is the burned area, which is the label y_k. Our aim is to find a regression function which can predict the burned area:

$$f: \boldsymbol{x} \in \mathbb{R}^A \longmapsto y \in \mathbb{R}^+$$

Note that the burned area can be zero, in this case there is no fire.

To evaluate this regression function, we are using three types of errors: mean squared error (MSE), mean absolute error (MAE) and mean squared log error (MSLE).

$$MSE(f) = \frac{1}{N}\sum_{k=1}^{N}(f(\boldsymbol{x}_k) - y_k)^2$$

$$MAE(f) = \frac{1}{N}\sum_{k=1}^{N}|f(\boldsymbol{x}_k) - y_k|$$

$$MSLE(f) = \frac{1}{N}\sum_{k=1}^{N}\left(\ln\left(1 + f(\boldsymbol{x}_k)\right) - \ln(1 + y_k)\right)^2$$

We are especially interested in the last one, because it penalizes more an under-predicted estimate than an over-predicted estimate. Indeed, it is worse to underestimate a fire, than to overestimate it.

For example, for a simple linear regression, the expression of the regression function is:

$$f(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x}$$

Where θ is the parameter that we are looking for. In this case, the error functions are:

$$MSE(f) = \frac{1}{N}\|X\boldsymbol{\theta} - \boldsymbol{y}\|_2^2$$

$$MAE(f) = \frac{1}{N}\|X\boldsymbol{\theta} - \boldsymbol{y}\|_1$$

We will try different types of algorithms and compare them with these error functions: nearest neighbors regression and SVM regression with different types of dimensionality reduction and feature selection processes.

To evaluate the algorithms, we use cross-validation. It consists in splitting the dataset in M parts: the regression algorithm is run M times, and each time one part of the dataset is used for testing, and the M-1 other are used for training. This method allows us to calculate the error with many different sets, to evaluate the algorithm more precisely.

3 Related work

Our work register itself in a continuity of work about fire prediction. Most past works use Support Vector Machines SVM to find satisfying results on fire prediction.

One work [7] applied two SVM consequently to classify fire risk into four degrees of hazard, using data from Lebanon to train the model and also using the weather data of the past day to predict the forest fire risk of the following day, in order to erase the error caused by weather condition prediction and increase generalization of the algorithm. We wanted to but couldn't apply this condition as our data set didn't have the weather conditions of every single day, but only the one that mattered.

Another worked focused [1] on the statistical analysis of the data used in Machine Learning algorithms for forest fire and found Bayesian method did the best approximation.

Furthermore, even k-clustering was used [5] in order to find the location where forest fires are likely to happen based on the fire history of a specific region.

All these works show the novelty of our approach, the majority of work has been focusing on classification and not on the task of regression regarding the prediction of burned area for forest fires.

Finally, P. Cortez and A. Morais [2] already worked on our dataset and tried to predict the burned area with regression. Hence our work is deeply connected to theirs. They used Multiple Regression, Neural Networks and Support Vector Machines algorithm for this purpose, and compared their performances using the Median Absolute Deviation metric which is less sensitive to outliers than Root Mean Squared Error. After selecting the features by hand (whereas we will try to implement feature selection algorithms) and fitting the algorithms on the whole dataset, they obtained a 60% accurately predicted with an admissible error of 2ha on the dataset.

Taking into account their results, we aim to study this dataset by ourselves and explore Machine Learning algorithms for this task and test their limitations; in order to see how relevant their results were.

4 Methodology

4.1 Strategy and pipeline

As it was explained in the previous section, algorithms to predict if a fire is going to take place already exists [1], as well as regression algorithms to predict the burned area [2].

When we studied our dataset, we saw that a big part of the burned areas was equal to zero, and that the data was very concentrated around zero. Thus, we decided to use a mixed pipeline: firstly, a classification algorithm to predict if a fire will happen, and secondly, if the fire is predicted, a regression algorithm to predict the burned area. By doing that, within our regression function, we can apply a logarithm to the burned area, because it will always be strictly positive.

Starting from our dataset, we create two training sets: one for classification, the other for regression. In the first set, each input vector with a strictly positive burned area is labelled as positive. In the second dataset, only input vectors with a strictly positive burned area are kept. We apply a logarithmic function to the burned area (which is always strictly positive) to make the data less concentrated around zero.

Hence, our final regression function is the result of the application of the classification and the regression:

$$f(\boldsymbol{x}) = \begin{cases} f_r(\boldsymbol{x}), & f_c(\boldsymbol{x}) = 1 \\ 0, & f_c(\boldsymbol{x}) = 0 \end{cases}$$

With:

$$f_c : \boldsymbol{x} \in \mathbb{R}^A \longmapsto y_c \in \{0,1\}$$

and

$$f_r : \boldsymbol{x} \in \mathbb{R}^A \longmapsto y_r \in \mathbb{R}_+^*$$

For regression, we tested five algorithms:

- Nearest Neighbors Regression: our dataset is not huge (only 517 inputs), that is why we firstly thought to use kNN.
- Support Vector Regression: according to [4] and [6] it is a very good algorithm.
- Linear Regression: it is one of the simplest algorithms, and we think that it is always should be tried. Indeed, if it gives good results, training and test have very low complexity compared to other algorithms.
- Stochastic Gradient Descent
- Neural Networks

For classification, we tested four algorithms:

- Nearest Neighbors: as well as for regression, we tried this algorithm because the dataset is quite small.
- Support Vector Classification
- Adaboost
- Decision tree

4.2 Pre-processing

4.2.1 Regression

For regression, the first processing we applicated to the output data was a logarithmic function. Indeed, it improved the variance and made the data easier to discriminate, as we can see on the following diagram:

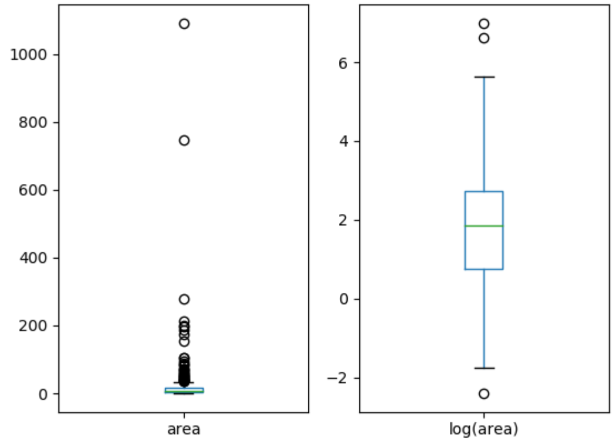

Figure 1. Box diagram of the burned area

There also was the day and the month to process: initially, they were not numerical data, but strings. Our first reaction was to associate a numerical value for each possibility (a number between 1 and 12 for the month, and between 1 and 7 for the day). However, this is not a good idea because it implies that there is an order in the categories: for example, we would logically associate January to 1, February to 2, and so on; but why would December be represented by a greater attribute than January? To counter this, instead of using one feature for the month, we create twelve features, one for each month, whose value is 1 is the fire has happened during the concerned month, 0 otherwise. We did the same with the day of the week.

We also tried creating new features: the season, which groups the months by three, and the weekend, which is 1 if the day is Saturday or Sunday, because we thought that forest fires are more likely to appear in summer and on the weekend when there are more tourists in the park. We also created a feature that we called hot, which is 1 if the month is May, June, July, August or September, and 0 otherwise. Finally, we removed the position X and Y from the dataset because they appeared useless to us, and in fact, when we tested, they appeared to be. Furthermore, it helps for the generalization of our algorithm, as it can now be applied to another dataset more easily since the location data is not taken into account.

Then, we created four training sets to compare the algorithms:

- X: all the numeric features (except X and Y) plus all the days and months features as described before: totally 27 features
- X1: all the numeric features (except X and Y) plus weekend and hot: totally 10 features
- X2: all the numeric features (except X and Y) plus weekend and the season: totally 13 features
- X3: only meteorological features: temperature, humidity, wind and rain

The data was the scaled, to make each attribute lie between 0 and 1. Moreover, we removed the features that had the same value for each input.

4.2.2 Classification

First, we created the label vector, with ones if the burned area was strictly positive and zeros otherwise.

We took the same training sets as regression to compare the algorithms, except that we did not remove the input vectors with a burned area equal to zero.

4.2.2 Feature selection

We tried to use algorithms for feature selection, but it was not convincing. This diagram shows the mean absolute error of the regression algorithms given the number of features, selected with an algorithm from scikit-learn library. The training set was the one with more features.

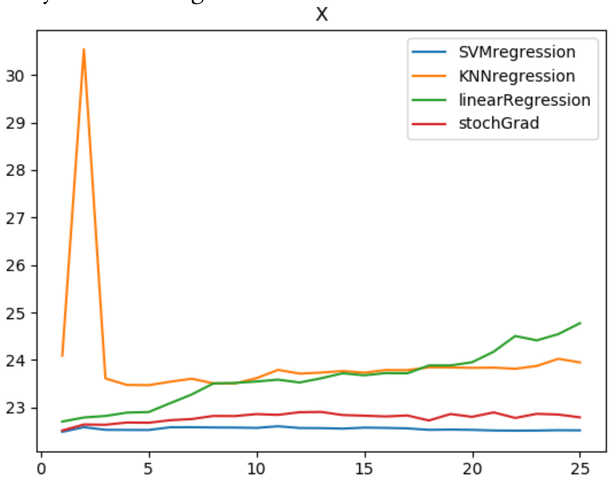

Figure 2. Error for different regression algorithms

We can see that there are no significant changes in the error when we change the number of features. Indeed, in the papers we read, the authors chose the features themselves. We tried to do the same when we create our four training sets.

5 Evaluation and results

To evaluate our algorithms, we used cross validation. For regression, we calculated the mean absolute error and the mean squared log error for the total of the tests. For classification, we did the same with accuracy, and we have also been mindful of the false negative rate, because for an algorithm that could be used to prevent forest fires, it is important that the number of undetected fires is as low as possible.

5.1 Regression

The following table shows the absolute mean error of each algorithm with each dataset.

	X	X1	X2	X3
SVM	22.51	22.53	22.55	22.44
kNN	23.94	23.63	23.52	23.23
Linear	24.7	23.36	23.72	22.67
Stochastic Gradient	22.83	22.6	22.69	22.58

Table 1. Mean absolute error

As we can see, the best algorithm is the SVM with the fourth dataset, which is the one with less features (temperature, humidity, wind and rain), as found in [2].
This table shows the mean squared logarithmic error of each algorithm for each dataset. We can see that SVM with the fourth dataset is the best again.

	X	X1	X2	X3
SVM	1.77	1.79	1.80	1.75
kNN	2.26	2.13	2.14	2.08
Linear	2.34	2.03	2.12	1.87
Stochastic Gradient	1.98	1.87	1.83	1.82

Table 2. Mean squared log error

5.2 Classification

These tables show the accuracy and the false negative rate of each algorithm.

	X	X1	X2	X3
Bayes	49.32%	42.75%	41.00%	51.13%
Adaboost	41.39%	41.00%	39.85%	47.00%
Tree	46.62%	46.23%	46.04%	48.36%
SVM	34.62	42.36%	42.75%	31.53%

Table 3. Accuracy

	X	X1	X2	X3
Bayes	5.60%	38.69%	37.33%	45.84%
Adaboost	26.89%	29.98%	29.79%	25.53%
Tree	28.05%	26.50%	27.85%	27.47%
SVM	21.08%	12.96%	17.80%	25.34%

Table 4. False Negative Rate

To have the best accuracy, we should choose Naive Bayes with the fourth dataset, but we would have a very high false negative rate. Indeed, the best solution is to take Naive Bayes with the first dataset: the accuracy is almost the same and the false negative rate is very low compared to any other possibility.

5.2 Whole algorithm

We plotted the REC curve of our whole algorithm, which is the union of the classification and the regression algorithms that gave the best results, as seen before.
It is clear that even for the best algorithms, the error is still high and the accuracy low. However, we should take into account the difficulty of the task, as well as the small size of the dataset. We should compare it to previous works [2], where only 60% accuracy was found having the model trained on the whole dataset.

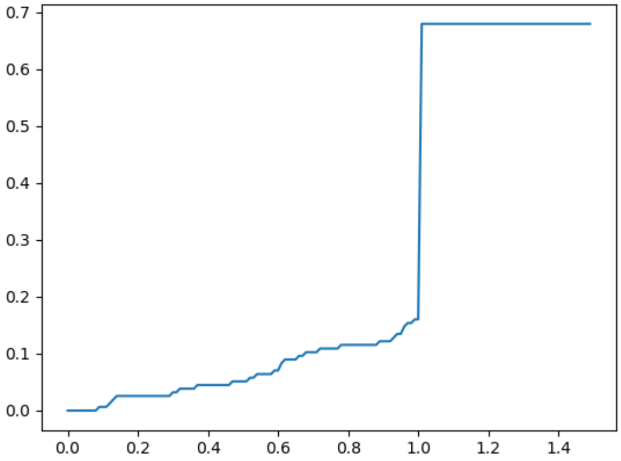

Figure 3. REC curve of the algorithm

CONCLUSION

Forest fires are one of nature most destructor hazards and seem to happen more frequently in recent years due to global warming. We tried to apply Machine Learning science to forest fire prediction, and we found that one of the best algorithm to predict the burned area of a forest fire was Support Vector Machines using only four features, which are corresponding to previous work's results.

However, the accuracy we had is not satisfying enough if we want it to help correctly firefighter departments prioritizing their targets when extinguishing fires. A further work would try to improve this accuracy working with a bigger dataset.

REFERENCES

[1] V. Iyer, S. S. Iyengar, N. Paramesh, G. R. Murthy, M. B. Srinivas. SENSORCOMM 2011. *Machine Learning and Datamining Algorithms for Predicting Accidental Small Forest Fires.* International Institute of Information Technology, Hyderabad, India. Louisiana State University, Baton Rouge, USA. University of New South Wales, Sidney, Australia. Brila Institute of Technology & Science, Hyderabad Campus, India.

[2] P. Cortez, A. Morais. 2007. *A Data Mining Approach to Predict Forest Fires using Meteorological Data.* University of Minho, Guimarães, Portugal.

[3] Government of Canada. 2019. *Canadian Wildland Fire Information System.* http://cwfis.cfs.nrcan.gc.ca/background/summary/fwi

[4] Pedregosa et al. 2011. *Scikit-learn: Machine Learning in Python.* JMLR 12, pp. 2825-2830. https://scikit-learn.org/stable/index.html

[5] I. Downard. 2018. *Predicting forest fires with spark machine learning.* https://mapr.com/blog/predicting-forest-fires-with-spark-machine-learning/

[6] M. Fragkiskos. 2018. *MA2823 Machine Learning course.* CentraleSupélec, Gif-sur-Yvette, France.

[7] G. Sakr, I. Elhajj, G. Mitri, U. Wejinya. 2010. *Artificial intelligence for forest fire prediction.* IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM. 1311-1316. 10.1109/AIM.2010.5695809.