Predicting forest fire in Algeria

1. Introduction

“Across Earth’s ecosystems, wildfires are growing in intensity and spreading in range. From Australia to Canada, the United States to China, across Europe and the Amazon, wildfires are wreaking havoc on the environment, wildlife, human health, and infrastructure.” [1] For a more effective approach, not only do emergency services need to be improved but a bigger focus on fire prevention should be done. Indeed prevention is crucial for the protection of human lifes and ecosystems. In many cases, certain weather and climate conditions combine to make wildfire events more likely affecting how they spread and how much land they burn through within short timeframes. Machine learning can be used to identify the conditions under which there are fire threats and thus improve fire prevention.

In this report an attempt of predicting the presence of fire in a region of Algeria based on the features of the given region is made. The problem is more clearly formulated in section 2 and the two machine learning methods that are applied to solve it are presented in section 3. The results are discussed in section 4 and the conclusions that arise from them can be found in section 5.The code is attached as an appendix in section 7, below the references (section 6).

2. Problem Formulation

The dataset used was collected by Faroudja ABID [2] and downloaded from the UCI machine learning repository [3].The data was collected between June 2012 and September 2012 in two regions: the Bejaia region (northeast of Algeria) and Sidi Bel-abbes (northwest of Algeria). Each data point represents one of the two regions on a given day. There are 247 data points in total and each point is characterized by 12 attributes that present the features (such as the temperature at noon, the wind speed and the relative humidity registered that day) of the region on the given day, and whether there is fire or not. The label is the class of the datapoint (column 12) : 0 = not fire, 1 = fire ; thus it is a binary classification problem. The other 11 columns are potential features. For a detailed list of the features and their possible values, see Table 1. In Appendix.To solve this problem we will apply two supervised binary classification methods.

3. Methods

3.1. Features engineering

3.1.1. Features selection

We keep all the features except the date (index 1 in Table 1 Appendix), because it seems reasonable to consider that the date itself (day and month from june to september) does not impact the probability to have fire. All the data is normalized to increase its visibility and reduce the effect of the units, which are different among the features.

3.1.2. Principal Component Analysis (PCA)

Our dataset has numerous features and given that they are strongly dependent on the environment, it seems reasonable to suppose that some of them are redundant. For example we can easily image a correlation between the rain and the relative humidity (RH). In order to reduce the dimensionality of our model we use Principal component analysis (PCA), which is a feature learning algorithm. It solves a series of optimization problems of linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space. The PCA plot shows that three dimensions have a cumulative explained variance of 90%, so we decide to keep three dimensions. The model is implemented in two datasets (with and without PCA reduction) to analyze the impact of PCA on the prediction accuracy.

3.1.3. K-fold

Our model should explain the fire for the two regions where the data have been collected so we should train, validate and test it with data points coming from both regions. To be sure that we have a homogenous dataset and to increase the validity of the accuracy we use k-fold cross-validation. We first create a testing set by keeping 10% of the data of each region (for a total of 24 points), then we apply the k-fold method on the resting data to create the validation and the training set (respectively 44 and 176 points).
In this method we apply the following steps : Dividing the data randomly into k parts, building the training set with the k-1 part of the whole set and keeping the last one for each of the combinations.

3.2. Logistic regression model

We apply logistic regression on each set combination generated by the k-fold method. We choose this model because it is a simple linear binary classification algorithm, adapted to our initial hypothesis that there exists a linear dependence between the features and the presence of fire. The chosen loss function for learning the hypothesis is logistic loss as it allows the use of a ready-made library for logistic (*sklearn.linear_model.LogisticRegression*). Logistic loss is used during the regression to optimize at each step the weights.

Formula for logistic loss :

$$L[(X,Y),h(.)] = -\frac{1}{m}\sum_{i=1}^{m} y_i(logh(x_i)) + (1-y_i)(log(1-h(x_i))) + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

3.3. Decision Tree classifier

Given logistic regression gives quite low accuracy, we now want to use a non-linear classification method. Among the non-linear methods we choose the decision tree classifier method because it is quite simple and easy to implement. Several values are tested for the depth of the tree on the set with PCA (see the section “max_depth optimization”). A depth of 3 is chosen because it gives the best accuracy (around 0.881818 in the max_depth test and 0.940909 in the general result phase). This parameter can be empirically chosen because we want a trade-off between a high accuracy and a low number of depth, in order to keep

the model simple and avoid overfitting. This means that we are not looking for the global maximum accuracy but rather the first local maximum accuracy.In addition the accuracy doesn't seem to vary much depending on the number of depth, so we can suppose that this local maximum is not far from the global one. The chosen loss function is entropy and it helps to determine how the features of the dataset should be split at each node of the tree. Entropy is a number representing the level of disorder of a dataset, ranging from 0 (homogeneous set) to 1 (very heterogeneous set). At each split the loss of entropy is maximized in order to maximize the information gain at each node. We chose entropy over gini impurity because it empirically gave us better results (an accuracy of 0.92 with gini impurity compared to a 0.94 accuracy on the validation set with PCA).

4. Results

All the results discussed below are available in the Result and Testing section of the Notebook joined in part 7.Appendix of the report.

4.1 Effectiveness of PCA

The final accuracy means of the samples with and without PCA implementation are the same (see the results table) showing that the PCA procedure has reduced the dimensionality of the model without removing any important elements from the data and thus proving its effectiveness.

4.2 Comparison between Logistic Regression and Decision Tree Classification Accuracy

The Logistic Regression and Decision Tree Classification model give an accuracy equal to 62% and 94% respectively (see the results table).The confusion matrix of the logistic regression model (section 'Evaluation Metrics for Classification Model') suggests that the low accuracy is due to the high number of false positive errors made with this model.The confusion matrix of the Decision tree classification method is consistent with the accuracy, with a low number of false positive and false negative errors. Because the accuracy is way better with the decision tree classification method, we chose it as our final method and can assume that the correlation between the features and the class of our problem is not linear.

4.3 Test set result

We apply our final chosen method on our test set (see part 3.1.3. of this report for the composition of the test set).The accuracy of the test set is 95.83 % (section 'Decision tree classification testing phase') which is similar to the one obtained on the validation set, equal to 94% (see results table).This shows that there is no overfitting on the validation set.

Out of the 24 data points of the test set only one has a wrong prediction, given with a confidence of ⅗ (which means that 3 of the 5 models built with k-fold crossing make the wrong prediction). All the other data points have the right prediction with a 5/5 confidence, meaning that the k-fold crossing builds homogenous samples and gives coherent models. (see Table in 'Decision tree classification testing phase'). This test set is small but these results are positive for the further applications of the model on bigger sets.

5. Conclusions

To summarize, we apply the PCA method to the environment features to extract the 3 more significant components of our features in order to simplify the problem without losing crucial information. Then we apply two supervised machine learning methods, Logistic Regression and Decision Tree Classification, to classify the data into two classes: if the given conditions lead to a fire (1) or not (0). We implement a k-fold crossing procedure with k = 5 to homogenize our dataset. The results of the validation set with Logistic regression is not conclusive (64% of accuracy and a lot of false positive errors), but decision tree classification fits the problem very well (94% of accuracy) so we choose it as our final model. To do a last verification on our final chosen model we use a test sample never used in the training phase and we observe an accuracy of 95%, which seems optimal. However we can still improve the model by focusing on the erronate predictions and looking at the influence of the out-layer on the training phase.To improve our model we consider applying it to more training data, coming from different regions and periods. To push the project one step further, the model should be able to predict the presence of wildfire with a greater advance using meteorological prediction. Being one step ahead will undoubtedly improve the emergency response and wildfire control and thus reduce the extent of damage.

6. Bibliography/References

6.1 . References

[1] report “Spreading like Wildfire: The Rising Threat of Extraordinary Landscape Fires” by UNEP and GRID-Arendal

link : Spreading like Wildfire: The Rising Threat of Extraordinary Landscape Fires | UNEP - UN Environment Programme

[2] Faroudja ABID et al. , Predicting Forest Fire in Algeria using Data Mining Techniques: Case Study of the Decision Tree Algorithm, International Conference on Advanced Intelligent Systems for Sustainable Development (AI2SD 2019) , 08 - 11 July , 2019, Marrakech, Morocco.

[3] link : UCI Machine Learning Repository: Algerian Forest Fires Dataset Data Set

https://archive.ics.uci.edu/ml/datasets/Algerian+Forest+Fires+Dataset++

6.2. Tables

Table1. List of the features of the dataset

index	name	meaning	units	range
1	Date	Date	(DD/MM/YYYY) Day, month, year	month : from june to september year : 2012
2	Temp	temperature noon (temperature max)	°C	22 to 42

3	RH	relative humidity	%	21 to 90
4	Ws	Wind speed	km/h	6 to 29
5	Rain	total rain in day	mm	0 to 16.9
6	FFMC	Fine Fuel Moisture Code (index from FWI system)	X	28.6 to 92.5
7	DMC	Duff Moisture Code (index from FWI system)	X	1.1 to 65.9
8	DC	Drought Code (index from FWI system)	X	7 to 220.4
9	ISI	Initial Spread Index (index from FWI system)	X	0 to 18.5
10	BUI	Buildup Index (index from FWI system)	X	1.1 to 68
11	FWI	Fire Weather Index	X	0 to 31.1
12	Classes	X	X	2 classes : ‘fire’ and ‘not fire’

7. Appendix