AN APPLICATION FOR GENERALIZED APPROACH TO AUTOMATIC AND UNSUPERVISIONED FLOODS PREDICTION BASED ON DATA DRIVEN MODELS

Autors: Luigi Passariello (*), Michele Passariello (**), Fabiano Rinaldi (***)

*INGV: National Instiute di Geophisics and Volcanology, public scientific centre of research of Italian Ministry of Research
** Ma.Pa.COM: ICT Company committed to the development of high added value solutions based on 4.0 technologies
CRSLaghi: Lake research and studies center

Keywords: Deep Learning, Flood prediction, Models Comparison, Data Analytics.

Abstract

This work is aimed at a study of some parameters through the use of AI methods, with the aim of identifying dynamics that allow establishing a level of flooding risk. The choice of parameters is the result of observations obtained in India and Bangladesh which are notoriously the states with the greatest number of floods and the greatest number of losses of human lives. After processing the data for better reading and interpretation, we applied the following Machine Learning models: Logistic Regression (LR), k-nearest neighbors (KNN), and eXtreme Gradient Boosting (XGBoost). CatBoost, LightGBM. We subsequently evaluated their ability and accuracy to predict flood events. Overall The algorithms performed well in prediction.

INTRODUZIONE

Flood detection refers to the process of identifying, monitoring, and alerting authorities or individuals about the presence or likelihood of flooding in a particular area. It involves the use of various technologies and methods to detect, predict, and mitigate the impacts of floods. According to the Aqueduct Global Flood Analyzer created by the World Resources Institute (WRI) «More people are affected by floods than by any other type of natural disaster» The interactive web platform shows that «On average every year around 21 million people in around the world could be affected by river floods. A number that could increase to 54 million in 2030 due to climate change and socio-economic development." The study aims to identify datasets and applications of ML models capable of guaranteeing a good prediction of flood events and a good generalization of these models.

MATERIALS AND NODELS

Dataset

The choice of the dataset and the informative value of the parameters used can determine the success or failure of the application of machine learning methods for the prediction of flood events. Another important factor is the generalization of the data collected for their application even in contexts other than those in which they were acquired. It is important to use data relating to areas in which a certain flood periodicity exists as in much of the world flood phenomena are classified as HILP or rare events or HILP (literally High

Impact Low probability). Therefore, countries such as India, Bangladesh and Pakistan, which have always been the theaters of the greatest number of floods in the world and the loss of a significant number of human lives, provide a quantity of data on flood phenomena that can be used for scientific progress in prevention of floods, in the early warning to the populations and in the study of structural solutions for territorial planning that can reduce the impact of flood events following rainfall. The dataset used consists of 20 parameters:

Parameters	**Number of samples**	**Type of Samples**	**Desxcription**
MonsoonIntensity	50000 non-null	Int64	Intensity of monsoons (or other rainfall conditions)
TopographyDrainage	50000 non-null	Int64	Topographic drainage capacity of the affected area
RiverManagement	50000 non-null	Int64	Management of rivers in terms of cleaning the banks and riverbeds
Deforestation	50000 non-null	Int64	Presence of deforestation activities close to the area affected by floods
Urbanization	50000 non-null	Int64	Presence of urbanization activities
ClimateChange	50000 non-null	Int64	Influence of climate change
DamsQuality	50000 non-null	Int64	Quality and safety of river barriers
Siltation	50000 non-null	Int64	Landfill and water pollution with granular materials or sedimentation
AgriculturalPractices	50000 non-null	Int64	Practice of agricultural activities near flood sites
Encroachments	50000 non-null	Int64	Presence of tributaries or other sources of intrusion into river waters
IneffectiveDisasterPreparedness	50000 non-null	Int64	Ineffective preparation to deal with disasters both in the prevention and construction phases
DrainageSystems	50000 non-null	Int64	Level of presence of drainage systems
CoastalVulnerability	50000 non-null	Int64	Coastal vulnerability is a spatial concept that identifies people and places that are susceptible to disturbances resulting from coastal hazards [11], [12]
Landslides	50000 non-null	Int64	Presence or risk of landslides
Watersheds	50000 non-null	Int64	Presence of watershed
DeterioratingInfrastructure	50000 non-null	Int64	Level of deterioration of river control infrastructure (giga embankments, etc..)
PopulationScore	50000 non-null	Int64	Presence of population near rivers
WetlandLoss	50000 non-null	Int64	Water saturation level in the soil
InadequatePlanning	50000 non-null	Int64	Inadequate planning to deal with the flood event in advance and during its evolution phases
PoliticalFactors	50000 non-null	Int64	Political factors
FloodProbability	50000 non-null	float64	Binary variable that indicates whether, compared to the values of the previous parameters, there was (1) or not (0) a flood event

For each parameter, 50,000 (0 to 49,999) observations are available, represented with whole numbers. As can be seen, the dataset presents heterogeneous information relating to risk factors connected to the triggering of flood phenomena. The dataset is made up of parameters that international experience has identified as contributing causes of flood phenomena. This is a multidisciplinary approach to define the risk profile of territories on the basis of data that is easy to find in various contexts at an international level. Each parameter is assigned a value (weight) ranging from 1 to 10 in relation to the impact it had with respect to

the specific flood event. The FloodProbability variable instead takes on the values 0 or 1 to indicate whether, given the combination of the other variables, there was a flood or not.

	MonsoonIntensity	TopographyDrainage	RiverManagement	Deforestation	Urbanization	ClimateChange	DamsQuality	Siltation	AgriculturalPractices	Encroachments	IneffectiveDisasterPreparedness	DrainageSystems	CoastalVulnerability	Landslides	Watersheds	DeterioratingInfrastructure	PopulationScore	WetlandLoss	InadequatePlanning	PoliticalFactors	FloodProbability
0	3	8	6	6	4	4	6	2	3	2	5	10	7	4	2	3	4	3	2	6	0
1	8	4	5	7	7	9	1	5	5	4	6	9	2	6	2	1	1	9	1	3	0
2	3	10	4	1	7	5	4	7	4	9	2	7	4	4	8	6	1	8	3	6	1
3	4	4	2	7	3	4	1	4	6	4	9	4	2	6	6	8	8	6	6	10	1
4	3	7	5	2	5	8	5	2	7	5	7	7	6	5	3	3	4	4	3	4	0

Models

In our study we will use machine learning models such as Logistic Regression (LR), k-nearest neighbors (KNN), and eXtreme Gradient Boosting (XGBoost). CatBoost, LightGBM

The Logistic Regression statistical model (also known as the logit model) [1]. [2], [3], is often used for classification and predictive analytics. Logistic regression estimates the probability of the occurrence of an event, such as a vote cast or not cast, based on a specific dataset of independent variables. Since the outcome is a probability, the dependent variable is constrained between 0 and 1. In logistic regression, a logit transformation is applied to the probabilities – that is, the probability of success divided by the probability of failure. This is also commonly known as log probability, or the natural logarithm of probabilities.

The value of the dependent variable comes from a list of finite categories that use binary classification. These are called categorical variables. An example is the result of rolling a six-sided die. This relationship is known as a logistic relationship.

The formula for logistic regression applies a logit transformation, or the natural logarithm of the probabilities, to the probability of success or failure of a particular categorical variable.

$$y = e^{(\beta 0 + \beta 1X1 + \beta 2X2 + \ldots \beta nXn + \varepsilon)} / (1 + e^{(\beta 0 + \beta 1 \times 1 + \beta 2 \times 2 + \ldots \beta n \times n + \varepsilon)})$$

Here's what each variable means:

- Y gives the probability of success of the categorical variable y
- e(x) is the Euler number, the inverse of the natural logarithmic function or the sigmoid function, ln (x)
- β0 is the y-intercept when all independent input variables are equal to 0
- β1X1 is the regression coefficient (B1) of the first independent variable (X1), the impact value of the first independent variable on the dependent variable
- βnXn is the regression coefficient (BN) of the last independent variable (XN), when there are multiple input values.

The beta parameter, or coefficient, in this model is commonly estimated using maximum likelihood estimation (MLE). This method tests different beta values through multiple iterations to optimize for best fit log likelihood. All these iterations produce the log likelihood function, and logistic regression tries to maximize this function to find the best parameter estimate. Once the optimal coefficient (or coefficients, if there is more than one independent variable) is found, the conditional probabilities for each observation can be calculated, recorded, and added together to produce a predicted probability. For a binary classification, a probability less than 0.5 will predict 0 while a probability greater than 0 will predict 1. After the model has been calculated, it is best practice to evaluate how well the model predicts the dependent variable, the so-called goodness of fit. The Hosmer–Lemeshow test is a popular method for evaluating model fit.Complexity of training for logistic regression methods with gradient based optimization is:

$$O((f+1)cNE),$$

where:

- **f** - number of features (+1 because of bias). Multiplication of each feature times it's weight (f operations, +1 for bias). Another f + 1 operations for summing all of them (obtaining prediction). Using gradient method to improve weights counts for the same number of operations, so in total we get **4* (f+1)** (two for forward pass, two for backward), which is simply **O(f+1)**.
- **C** - number of classes (possible outputs) in your logistic regression. For binary classification it's one, so this term cancels out. Each class has it's corresponding set of weights.
- **N** - number of samples in your dataset, this one is quite intuitive I think.

- **E** - number of epochs you are willing to run the gradient descent (whole passes through dataset)

Because in our case C=1 computational complexity becomes

$$O((f+1)NE),$$

The k-nearest neighbors algorithm [4], [5], [6], also known as KNN or k-NN, is a non-parametric supervised learning classifier, which uses proximity to make classifications or predictions on the clustering of a single data point. Although it can be used for regression or classification problems, it is generally used as a classification algorithm, based on the assumption that similar points can be found close to each other. For classification problems, a class label is assigned based of a majority vote, e.g. the most frequently represented label around a given data point is used. While this is technically considered "plurality voting", the term "majority voting" is more commonly used in the literature. The distinction between these terminologies is that "majority voting" technically requires a majority greater than 50%, which works mostly when there are only two categories. When you have multiple classes, e.g. four categories, you don't necessarily need 50% of the votes to draw a conclusion about a class; you can assign a class label with a grade higher than 25%.The computational complexity of the KNN algorithm mainly depends on the size of the dataset, the number of features, and the value of k. The time complexity of the KNN algorithm for a single query point is

$$O(nd)$$

where

- n is the number of training examples and
- d is the number of features.

This is because for each query point, the algorithm needs to compute the distance between the query point and every other point in the dataset. As the number of features or the size of the dataset increases, the computational complexity of KNN also increases significantly, making it computationally expensive and impractical for large datasets. Additionally, finding the optimal value of k by brute force can also increase the time complexity of the algorithm.

XGBoost

XGBoost is a popular DT gradient boosting algorithm, designed for improving the speed and performance of GBDT [7]. XGBoost uses a second-order Taylor approximation and regularization term for the sum of squared errors to minimize the loss function and make it too tight. XGBoost has a low computational complexity of

$$O(Kd||x||\log n),$$

where

- K is the number of trees,
- d is the maximum tree depth,
- ||x|| is the number of non-missing samples and
- n is the data size [7].

Additionally, XGBoost supports parallel execution to save model learning time.

LightGBM

LightGBM is a fast and robust ensemble ML model built from multiple DTs [8]. The main advantage of LightGBM over other ML methods is its ability to efficiently handle large-scale and large-dimensional data. One-based Gradient Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) are the two main strategies of LightGBM [8].

- GOSS is a downsampling method that preserves only data samples with large gradients and randomly discards small gradient samples to speed up model training and reduce memory consumption.
- EFB is a feature engineering method that packages mutually exclusive features as single features to minimize feature size and improve model training efficiency.

Using GOSS and EFB, data size can be significantly reduced without the loss of critical information. The time and also spatial complexity of LightGBM has been reduced to:

$$O(N0F0),$$

where N0 is the reduced number of samples after using GOSS, F0 is the number of clustered features after using EFB [10].

CatBoost

CatBoost is another advanced gradient boosting algorithm designed to process categorical features more effectively [9]. CatBoost, compared to existing gradient boosting models, includes three significant model improvement components:

- symmetrical trees,
- orderly strengthening e
- native functionality support.

In symmetric trees, leaves are split under the same conditions as in previous trees, and the pair of feature splits with the lowest loss is applied to all nodes. Using symmetric trees can improve the model's prediction speed and reduce overfitting. Sorting boosting is a permutation-based technique that prevents overfitting on small datasets by training a model on one subset while computing residuals on another subset. CatBoost's native feature support means that it can directly process all types of features, such as numeric, textual, and categorical features, without the need for additional preprocessing. CatBoost is an ensemble model with low computational complexity:

$$O(SN),$$

where S is the number of subset permutations and N is the number of basic DT models [9].

RESULTS AND DISCUSSION

Correlation indicates the tendency that two variables (X and Y) have to vary together, that is, to covary. Bivariate correlation coefficients are one of the most used statistical indices to evaluate the relationship between variables. Within a research project, several are often calculated which are then summarized in a single table, called a correlation matrix. The correlation matrix is a square table (i.e. with the same number of rows and columns) which shows in the row and column headers the list of variables on which you want to evaluate the correlation. The individual bivariate correlation indices are indicated in the individual cells within the table. The bivariate correlation matrix relating to the data of the dataset considered is the following:

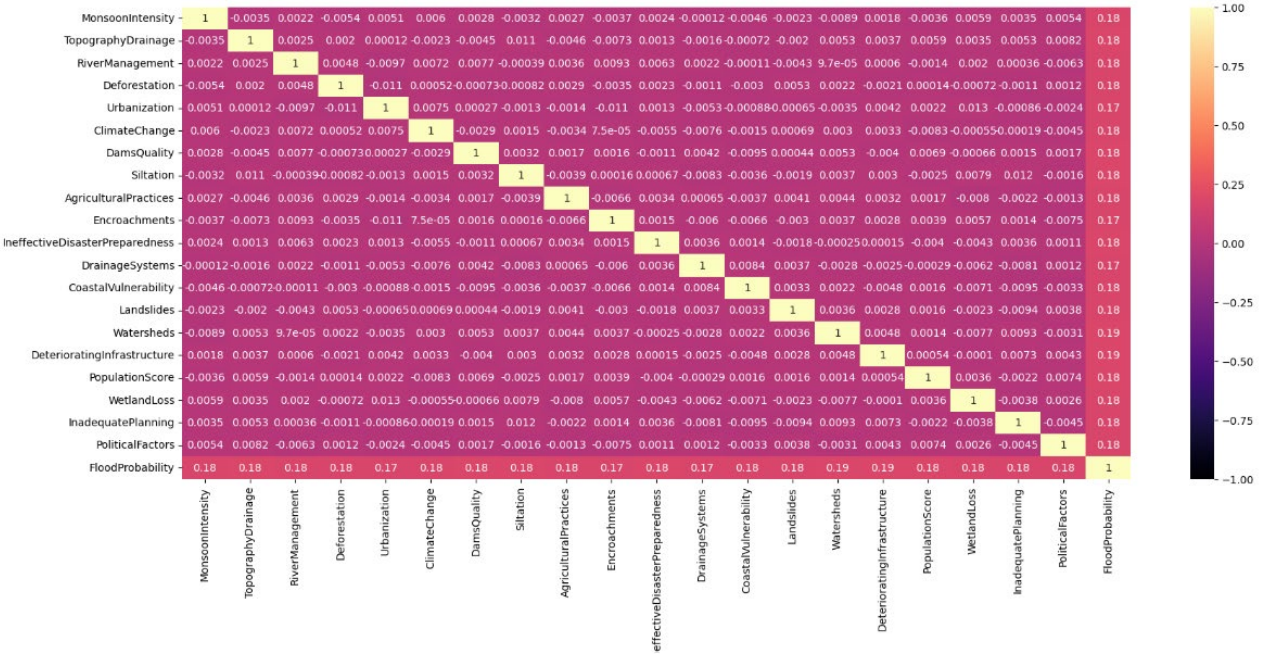

Figure 1 – Correlation matrix between all parameters

All the variables have a very low correlation index between them, either positively or negatively. We can assume that they are independent variables. The only variable most correlated with all the others is the Flood Probability with a value that varies in the range [0.17, 0.18]. This is encouraging as it correlates the occurrence of the flood event or not with the variables considered in various ways as contributing causes of the event

itself.. Let's now check for the presence of outliers, i.e. anomalous values are particular values assumed by a statistical variable which appear to be very different from the rest of the distribution. Identifying outliers in a statistical distribution is a very important activity that can have many practical applications. In some cases the outliers arise from simple manual errors in measurement or data imputation. In other cases, however, the outliers are "real values" that do not derive from errors. Precisely because of their particularity, identifying them can be very useful to verify the particular conditions that lead to the generation of these cases. We then calculate the boxplots:

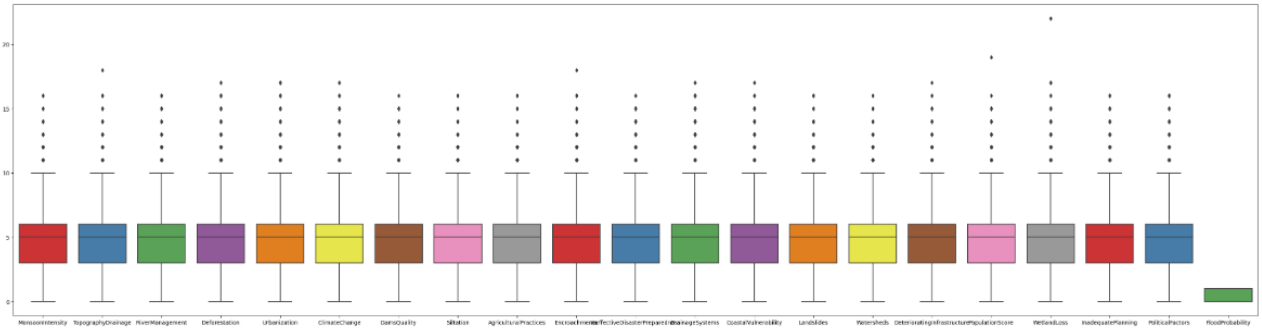

Figure 2 – Analysis of outliers by BoxPlot

We can notice the presence of numerous outliers. We assume the elimination of outliers by considering them to be the subject of incorrect attribution or alternatively of exceptional findings with low information content for our forecasting purposes. Since the data is normally distributed, we can use the z-score method to eliminate outliers. The Z-score is a statistical measure that indicates how many standard deviations a data point is from the mean. A Z-score greater than three or less than the negative three is considered an outlier.

Z-score is a reliable tool for detecting outliers, but requires an effective sample size. After removing the outliers our data looks like the following figure:

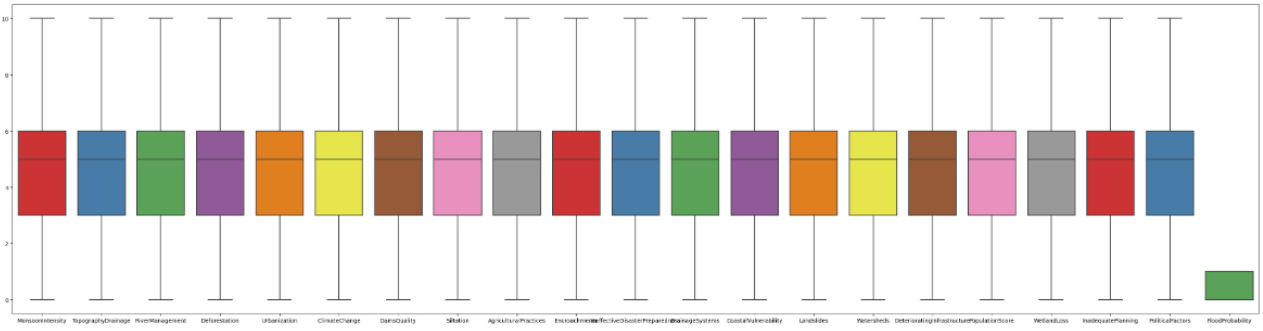

Figure 3 – Boxplot of dataset without outliers

We can note that the data relating to the FloodProbability variable are unbalanced, as the 0 or events in wh ich there was no flood are 22,895 and those in which there was a flood or the sequences with Flood Probabi lity 1 are 14,998. The accuracy of a learning model is greater if the data set is balanced. By removing 7897 e ntries that have the Flood Probability value at zero we obtain a balanced dataset of 14998 sequences that d id not generate floods and 14998 sequences that generated floods respectively; in other words 14998 sequ ences with value 0 of the FloodProbability variable and 14998 sequences with value 1 of the same variable.

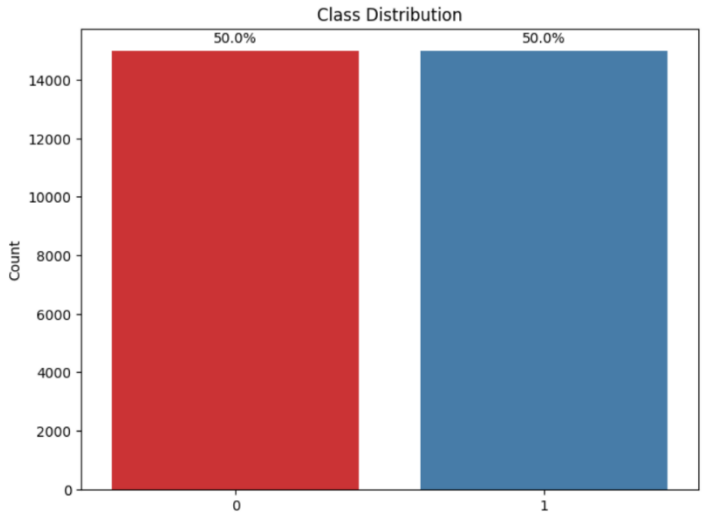

Figure 4 – Class observations distribution

Now we can apply the LG, KNN and XGBoost, CatBost and LightGBM models that we introduced previously. Using the features made available by Python and using the reduced dataset with the elimination of outliers and the balancing operation. We consider the precision, recall and F1 metrics to evaluate the accuracy of the three models. We obtained the following results forecasting flood events:

Modello	**Risultati delle metriche adottate**				**Accuratezza**
LG		precision	recall	f1-score	support	100%
	0	1.00	1.00	1.00	4477	
	1	1.00	1.00	1.00	4522	
	accuracy			1.00	8999	
	macro avg	1.00	1.00	1.00	8999	
	weighted avg	1.00	1.00	1.00	8999	
KNN		precision	recall	f1-score	support	83%
	0	0.79	0.93	0.86	4477	
	1	0.92	0.76	0.83	4522	
	accuracy			0.84	8999	
	macro avg	0.85	0.84	0.84	8999	
	weighted avg	0.86	0.84	0.84	8999	
XGBOOST		precision	recall	f1-score	support	92%
	0	0.92	0.92	0.92	4477	
	1	0.92	0.92	0.92	4522	
	accuracy			0.92	8999	
	macro avg	0.92	0.92	0.92	8999	
	weighted avg	0.92	0.92	0.92	8999	
CATBOOST		precision	recall	f1-score	support	96%
	0	0.98	0.95	0.96	4477	
	1	0.95	0.98	0.97	4522	
	accuracy			0.96	8999	
	macro avg	0.97	0.96	0.96	8999	
	weighted avg	0.97	0.96	0.96	8999	

LightGBM	precision recall f1-score support 0 0.92 0.89 0.90 4477 1 0.90 0.92 0.91 4522 accuracy 0.91 8999 macro avg 0.91 0.91 0.91 8999 weighted avg 0.91 0.91 0.91 8999	91%

If we plot the accuracy results of the various algorithms we obtain the following comparison graph

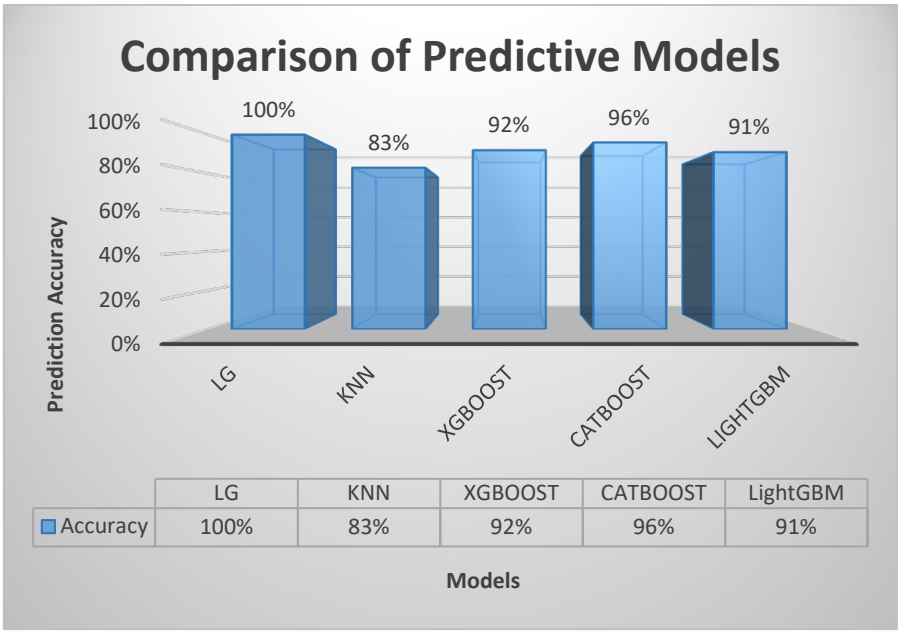

Figure 5 – Comoarison of accuracies (all predictive models)

We can see that using Logistic Regression we archive a flood prediction accuracy of 100%. For this type of weighted data it certainly represents the best performing method also considering the fact that the sample used for training is significant. A good prediction performance is also obtained with the CatBoost model which reaches an accuracy of 96%. Only the KNN algorithm achieves a performance lower than 90%, reaching an accuracy of 83%. The algorithms are very stable and converge regularly. The test set was constructed with 30% of the sequences.

If we evaluate the execution times, with the same number of samples, of the various flood prediction models we see that the LR is also confirmed in temporal terms as the optimal choice as highlighted in the following table:

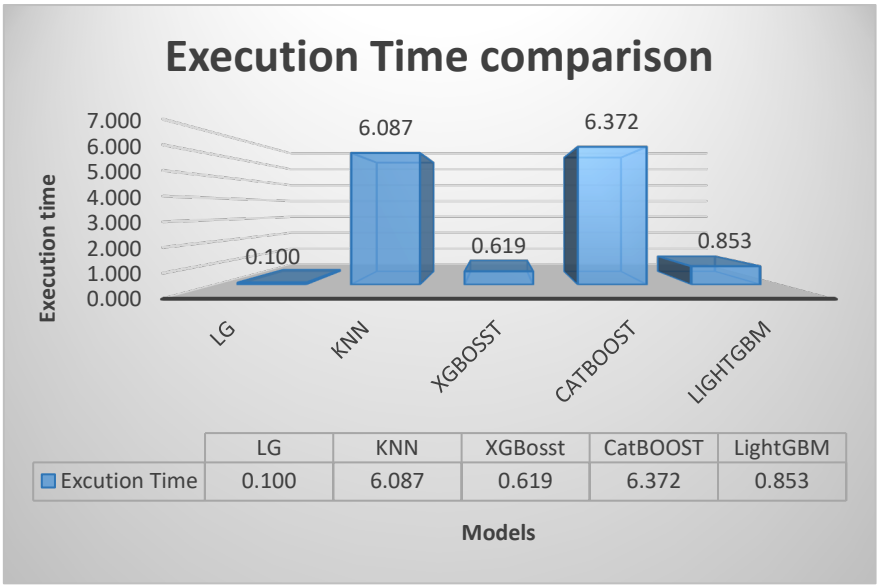

Figure 6 – Execution time comparison (all models)

For a best comparison we can use the following graph of Accuracy of prediction Vs Execution time:

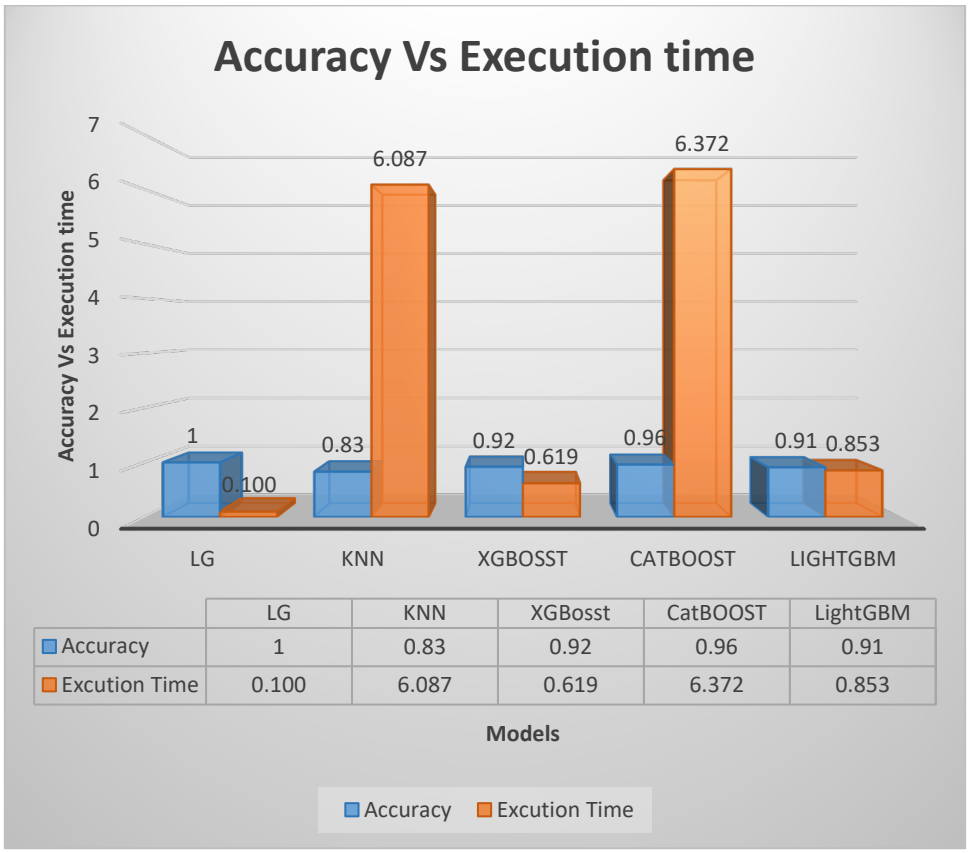

Figure 7 – Accuraacy vs execution time

In the previous graph we see how CATBoost, which seemed to have interesting performances in terms of accuracy, requires execution times greater than 640 times those of Logistic Regression. Therefore, for a data set like the one considered, the best performances are obtained with the logistic regression algorithm.

CONCLUSIONS AND FUTURE WORKS

This work started by considering a data set based on 20 parameters detected in the face of rainfall whether these rainfalls caused or did not cause floods. Each parameter represents a possible contributory cause to favor the triggering of a flood. The choice of parameters is the result of observations obtained in India and Bangladesh which are notoriously the states with the greatest number of floods and the greatest number of losses of human lives. After processing the data for better reading and interpretation, we applied the following Machine Learning models: Logistic Regression (LR), k-nearest neighbors (KNN), and eXtreme Gradient Boosting (XGBoost). CatBoost, LightGBM. We subsequently evaluated their ability and accuracy to predict flood events. Overall The algorithms performed well in prediction. In particular, the logistic regression (LR) obtained a prediction accuracy of 100% even by changing some boundary conditions such as a different subdivision of the quantity of sequences in the dataset intended for the test phase (30%, 20% 40%). another element that the study highlighted is that of execution time. From an evaluation carried out on an Acer I5 Kaptop, logistic regression was also found to be the fastest computationally.

For the future we plan to analyze different combinations of parameters and export this model to some areas of Italy such as Campania, Liguria and Emilia Romagna, which in Italy represent the territories affected by floods and inundations.

BIBLIOGRAPHY

[1] *Tolles, Juliana; Meurer, William J (2016). "Logistic Regression Relating Patient Characteristics to Outcomes". JAMA. 316 (5): 533–4.* doi:10.1001/jama.2016.7653. ISSN 0098-7484. OCLC 6823603312. PMID 27483067.

[2] k *Hosmer, David W.; Lemeshow, Stanley (2000). Applied Logistic Regression (2nd ed.). Wiley.* ISBN 978-0-471-35632-5.[page needed], ^ Jump up to:a b Cramer 2002, p. 10–11.

[3] Palei, S. K.; Das, S. K. (2009). "Logistic regression model for prediction of roof fall risks in bord and pillar workings in coal mines: An approach". *Safety Science*. **47**: 88–96. doi:10.1016/j.ssci.2008.01.002

[4] D. Wettschereck and D. Thomas G.,"Locally adaptive nearest neighbouralgorithms," Adv. Neural Inf. Process.Syst., pg. 184–186, 1994.

[5] Han EH.., Karypis G., Kumar V.(2001) "Text Categorization UsingWeight Adjusted k-Nearest NeighbourClassification". In: Cheung D., WilliamsG.J., Li Q. (eds) Advances inKnowledge Discovery and Data Mining.PAKDD 2001. Lecture Notes inComputer Science, vol 2035. Springer,Berlin, Heidelberg.

[6] Shengyi Jiang,GuansongPang,Meiling Wu,Limin Kuang, "Animproved K-nearest-neighbour algorithmfor text categorization",Expert Systemswith Applications,Elsevier(2012)

[7] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System,"in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[8] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," Adv. Neural Inf. Process. Syst., vol. 2017-December, no. Nips, pp. 3147–3155, 2017.

[9] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: Unbiased boosting with categorical features," Adv. Neural Inf. Process. Syst., vol. 2018-December, no. Section 4, pp. 6638–6648, 2018.

[10] L. Yang and A. Shami, "A Lightweight Concept Drift Detection and Adaptation Framework for IoT Data Streams," IEEE Internet Things Mag., vol. 4, no. 2, pp. 96–101, 2021.

[11] W.N. Adger, Successful adaptation to climate change across scales, Glob. Environ. Change, (2005)

[12] W.N. Adger, Social-ecological resilience to coastal disasters, Science, (2005)