Classification of extreme weather events

Michell Mercedes Payano Pérez, Matricule:20265175, MichellP

[a] *University of Montreal*

Abstract

In this report we will be able to see step by step the process to design a supervised machine learning model capable of accurately classify weather events as part of the Kaggle competition about detection of extreme weather events from atmospherical data.

Keywords: Atmospherical data, Logistic regression, Ensemble models, classification

1. Introduction

The objective of this report is to present the whole process and strategies for model selection followed in order to obtain the best model that could accurately predict extreme weather events from atmospherical data. The three classes given in the data set are standard background conditions, tropical cyclones and atmospheric river from 120 locations. At first 4 models were fitted, including a Logistic Regression model built from scratch by the author of this report, and other 3 ensemble models, on 6 versions of the original data. Then, the second stage of this experiment was to perform hyperparameter tuning using exhaustive grid search in order to evaluate the best combination of hyperparameters. As a result, the best model was the Histogram-based Gradient Boosting, that returned an accuracy in the test set of 0.786.

2. Feature Design

In this data set, we have 19 predictors (18 numerical and 1 date time columns) after excluding the SNo column, which only represents an id or index of the observations. In this experiment, 4 models were applied, including the logistics regression, to different variations of the data set which are detailed as follows:

1. First Data set: After examination of the longitude and latitude columns, it is possible to see in Figure 1, as it is presented in for both the train and test data sets, that the locations are clustered. In this case, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), was used to make the clusters of this data just as Boeing (2018) did when working with spatial data set of GPS latitude-longitude. As a result, in this case the features used for prediction contained the new created variable (season), and the rest of the numerical values from the original data set, excluding longitude and latitude for this first trial, since the new variable was created using both of them.
2. Second Data set: For a exploratory analysis, the features of longitude and latitude where included, just to evaluate if there could be a change in the prediction before and after excluding them.

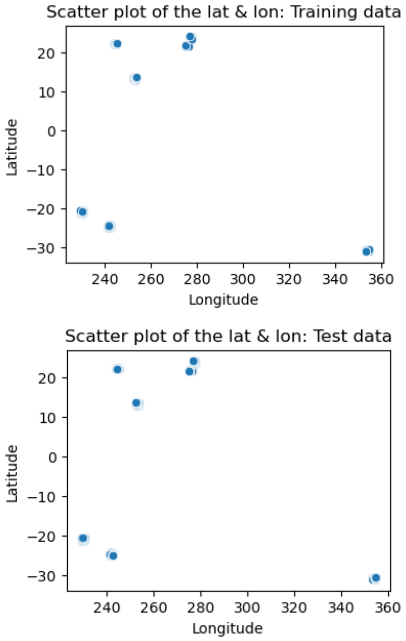

Figure 1: Simple scatter plot that shows how the 120 locations are distributed.

3. Third Data set: Since weather events occur mostly in a given time frame during the year such as tropical cyclones which, according to the National Hurricane Center and Central Pacific Hurricane Center, official season for the Atlantic basin is from June to November of every year. For this reason, a new categorical column called "season" was created, taking into account the latitude and the month extracted from the time column. This feature contains 4 categories: winter, spring, summer and fall, and was created taking into account if the location is in the Northern or Southern Hemisphere.
4. Fourth Data set: After performing another exploratory analysis in the original features, we can see in the correlation matrix that there is linear relationship among most of the numerical variables as it is showed in Figure 2. Even though we cannot determine non-linear relations using the

typical correlation matrix, at least we can see that there are many variables that are strongly correlated with one another, so that means that it is important to perform feature selection to remove redundant features. For this case, the method used to perform feature selection was the "Recursive feature elimination" or RFE. Such as Kuhn and Johnson (2019) defines it, RFE is a simple backwards selection procedure where the largest model is used initially and, from this model, each predictor is ranked in importance. In other words, a model is fitted with the whole data set first, then the less important features are removed and the model if refit again until it reaches a specific number of features that yield the best performance. That said, the model used to evaluate the feature importance among the variables is the Decision Tree.

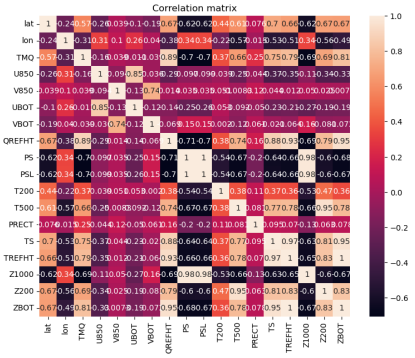

Figure 2: Correlation matrix among numerical features.

5. Fifth Data set: From the features selected in the experiment above, a non linear transformation was applied, specifically the degree 2 polynomial.
6. Sixth Data set: Since the original data in highly imbalanced, as it is depicted in Figure 3. In this case, the 0 class means standard conditions, 1 is tropical cyclone and 2 is atmospheric river. For this case, an oversampling method was used to address this issue. More specifically, the Synthetic Minority Oversampling Technique (SMOTE) was applied, which consists on increasing the amount of minority classes by creating artificial examples using the K Nearest Neighbor (KNN) algorithm.

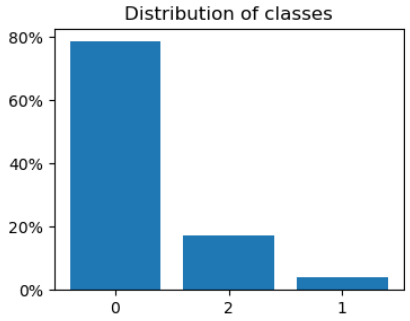

Figure 3: Imbalanced data set.

3. Algorithms

3.1. Logistic Regression (LR)

This is linear classifier that determines the probability of an output to belong a given class by applying the sigmoid function. For this project the logistic regression model for a multi class problem was built from scratch. The particular strategy applied to perform the classification of the 3 classes is one-vs-all, which consists in splitting the classification problem into many binary classification models. Therefore, as there are 3 classes (0, 1 and 2), 3 binary logistic regression models were applied.

3.2. Random Forest (RF)

This is an ensemble machine learning model, which means that combines the predictions of other models, in this specific case, of trees. The essential idea of RF is that they employ bagging to improve the performance and reduce the variance of a predictive model. This kind of model starts by creating multiple subsets of the original dataset and creates multiple trees, each one yielding to its own prediction. When making predictions, all the decision trees in the forest individually produce their own outputs (classifications or regression values). In the case of classification, the final prediction is often determined by a majority vote among the trees, and for regression it is usually the mean of the predictions.

3.3. Extreme Gradient Boosting (XGB)

This is also an ensemble machine learning model, which has been widely popular in many kaggle competitions. Such as RF, it combines the predictions of many trees, but with the difference that it works by iteratively improving the model's performance, focusing on the instances where the previous models, trees, made mistakes. In other words, each tree is built one at a time in order to correct the prediction errors of the previous one. It also incorporates the regularization component, either L1 or L2 norm, in order to control overfitting.

3.4. Histogram-based Gradient Boosting (HGB)

This model also tries to build trees iteratively in order to improve the performance of these models. However, the difference is that that instead of creating the decision trees by considering all the possible splits for each feature, the data of each feature is binned, which reduces the number of splits. As a result, the time for the training process is reduced.

4. Methodology

Even though there are many metrics to use in order to evaluate the performance of these classifiers, especially when dealing with an imbalanced data set, the accuracy was the only metric used for the purpose of this competition. As a first step, all the models were fitted to each data set using their default hyperparameter settings. After comparing them, the best performing model, with the exception of the Logistic Regression, was chosen with the given data set and then it's hyperparameters were tuned using the grid search methodology.

It is important to note that during the first stage of training and evaluation, a stratified k fold cross validation with 10 partitions was used. Furthermore, this process was repeated 10 more times, since as Brownlee (2020) states, the estimate performance of a model using a single k fold cross validation process may get noisy results, so a possible solution is to is to repeat the cross validation process multiple times and report the mean performance across all folds and all repeats. Another important thing to consider is that for each data set, the numerical features were normalized in order to have mean 0 and standard deviation of 1, and the categorical columns were one hot encoded.

Given that the main objective of the competition is to compare the performance of a logistic regression model implemented from scratch, with any other model, the results of the first experiments will be presented below, followed by the tables of their corresponding hyperparameters that were tuned. For the case of the LR, it is a simpler version of the model than those that can be found using python libraries, such as Scikit-Learn for example. Therefore, the only hyperparameters included in this version of the LR are the l2 regularizer, the number of steps or iterations and the step size or learning rate.

5. Results

After running the first stage of this project, which is fitting each model to the 6 data sets describe in the Feature Design section, using their default hyperparameters, is presented in Table 1. As we can see, the model with the best performance is the Histogram-based Gradient Boosting. Furthermore, the majority of the classifiers presented a slight increase in their performance using the data set 4, which is the one that contains only the selected features after applying the recursive feature selection method.

Table 1. Accuracy of the classifiers per experiment

Models	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Exp.6
LR	0.775	0.780	0.776	0.784	0.737	0.511
RF	0.862	0.862	0.862	0.862	0.862	0.849
XGB	0.880	0.880	0.880	0.881	0.878	0.869
HGB	0.886	0.886	0.886	0.887	0.886	0.873

Since the best performing model is the HGB, the Table 2 show the hyperparameters that were tuned using grid search. Since this method tries an exhaustive combination of all the values and this can be computational expensive, instead of applying a 10 fold cross validation repeated 10 times, a 5 fold one repeated 3 times was used. That said, only 4 of the 20 hyperparameters we tuned. As a result, the best perfomance, given the best selection of the hyperparameters, is equal to 0.892, which is not that different as the one given the default settings of the model. As a result, this yield to a 0.786 in accuracy for the test set at the Kaggle platform.

Table 2. Hyperparameters of the HGB model

Hyperparameter	Options	Selected
regularization	[0, 0.001, 0.01]	0
learning rate	[0.1, 0.01, 0.5, 0.05]	0.05
max bins	[225, 255]	225
max iter	[100, 150, 200]	100

In the case of the LR, also the fourth data set was used to tune its hyperparameters. As explained in the Methodology section, there are only 3 hyperparameters in the model built from scratch, however, in the Scikit Learn version from python, it has 15. The same repeated cross validation method was applied as in the case of the HGB. As a result, the best combination resulted in an accuracy of 0.810 and for the test set submitted at kaggle of 0.765.

Table 3. Hyperparameters of the LR model

Hyperparameter	Options	Selected
regularization	[0.01,0.05,0.1,0.5]	0.01
stepsize	[0.01,0.05,0.1,0.5]	0.5
n steps	[1000,1500]	1000

6. Discussion

We just saw that the model that performed better at classifying extreme weather events was the Histogram-based Gradient Boosting using a subset of the features given in the original data set. Even though this data set is highly imbalanced, and after applying the oversampling technique called SMOTE the models didn't show a better performance. There are many other techniques for either oversampling and also undersampling that can be tested for this project, even though due to a time constraint they weren't implemented, it is still a good a idea to so do.

Another important aspect to note is that, during this experiment hyperparameter tuning was performed on the model that yielded the best accuracy using first its default setting, besides on Logistic Regression, which as we could see in Table 1, was the HGB. However, before and after the process of model selection, we saw that there wasn't a relevant change in the accuracy metric for the HGB, even though it was for the LR. This could suggest that it is important to perform more feature engineering strategies, which could be by applying some methodological knowledge in order to create other features using the original ones. Other thing that could improve model selection is to specify a wider range of the values for each hyperparameter, which could lead to more computational and time power, but it could be worth of trying.

Statement of Contributions

I hereby state that all the work presented in this report is that of the author.

References

Boeing, G., 2018. Clustering to reduce spatial data set size. arXiv preprint arXiv:1803.08101 .

Brownlee, J., 2020. Repeated k-fold cross-validation for model evaluation in python. URL: https://machinelearningmastery.com/repeated-k-fold-cross-validation-with-python/.

Kuhn, M., Johnson, K., 2019. Feature engineering and selection: A practical approach for predictive models. Chapman and Hall/CRC.