Extreme Weather Events Identification using Machine Learning

Philip Bui
Department of Mathematics and Statistics
Université de Montréal
20129010
philip.bui@umontreal.ca

Abstract

This report explores the application of machine learning to classify extreme weather events, addressing the need for better prediction and preparedness in the face of climate challenges. Focusing on two rare weather events, namely tropical cyclones and atomspheric rivers, I employ data preprocessing, feature engineering, and various machine learning algorithms to achieve promising accuracy, highlighting the relevance of machine learning in environmental problem-solving.

1 Introduction

Climate-related events have become a focal point of societal concern in recent years due to their increasing frequency and severity in a warming world. To address these challenges, machine learning has emerged as a powerful tool for predictive and classification tasks. This report recognizes the critical importance of accurately classifying extreme weather events, particularly tropical cyclones (TC's) and atmospheric rivers (AR's), which have far-reaching impacts on communities, economies, and ecosystems. Accurate classification is essential for early warning systems, efficient ressource allocation, and advancing climate research. In light of these considerations, this study seeks to develop predictive models that effectively classify these rare and impactful weather phenomena. I will employ data preprocessing and feature engineering techniques to enhance model performance and address class imbalance issues inherent in the dataset.

A few notable challenges were encountered during this study. First, the ClimateNet dataset presents relatively high class imbalance due to the nature of TC's and AR's. The rarity of these events in real life results in an inherently imbalanced dataset. Secondly, while TC's have notably different characteristics than background samples, AR's present a lot of similar features to background information, making the classification between background and AR more difficult.

In response to these challenges, I conducted a series of experimental modelling, focusing mostly on tree-based models, exploring various model architectures including data augmentation techniques, regular and meta feature engineering, and model ensembling. My results demonstrated an increase in accuracy compared to the baseline, highlighting the critical role of addressing class imbalance in the success of machine learning models applied to rare event classification, showcasing the practical relevance of this approach in the context of climate science and environmental problem-solving.

IFT6390 - Fundamentals of Machine Learning, Fall 2023, Université de Montréal (LateX template borrowed from NIPS 2017.)

2 Dataset / Feature Design

2.1 ClimateNet dataset

Supervised machine learning methods often require plentiful and reliable data, which has not been readily available in climate science. The ClimateNet dataset is a proposed solution to this problem by Prabhat et al. (2021). At its core, it is an open, community-sourced human-expert-labeled curated dataset that captures tropical cyclones (TC's) and atmospheric rivers (ARs) in high-resolution climate model output from a simulation of a recent historical period. The complete data set is nearly 30 GB and contains climate variables and images from nearly 900,000 locations from 1996 to 2013. However, while such a high-resolution dataset and image information can potentially provide great predictive power, it requires great computational power and storage. It is also interesting to consider what can be done using less data with traditional machine learning methods. Thus, the dataset used for this report is a reduced version of only 120 locations while keeping all time points and only contains the tabular information (not the images) available from the ClimateNet dataset.

Each data point contains 16 atmospheric features (18 including longitude and latitude) and the result of one human-labeled segmentation map (Table 1). The labels are `'0'` for background, `'1'` for tropical cyclones, and `'2'` for atmospheric rivers. For the rest of this report, these will be referred to as vanilla features. Two aspects of this labelling are to note. Firstly, these labels were hand-drawn by climate scientists following the methodology described in Prabhat et al. (2021). Secondly, these labels present some class imbalance. As expected, TC's and ARs are relatively infrequent weather events. This is reflected in the data set as TC's and ARs represent only 4.08%, 17.33% of observations in the full training set respectively. This class imbalance is explored further in the methodology.

2.2 Feature engineering

The ClimateNet dataset, even in its reduced format, offers a wealth of information comprising fundamental atmospheric features. In my pursuit to unlock the full predictive potential of machine learning, I employed a systematic approach to augment and refine the dataset's features. The methodology employed for feature engineering involved the addition of potential features to the training set, followed by an assessment of their performance on the validation set using a meticulously tuned Light Gradient Boosting Model (LGBM). The selection of the LGBM model was based on its expedited training capabilities, allowing us to efficiently evaluate numerous feature candidates. My guiding principle was that if a feature failed to enhance the performance of a baseline LGBM model, it was considered less likely to contribute to the final model's predictive capabilities. Human judgement and exploratory data analysis was also accounted for in this exploration (*i.e.*, if the feature made sense logically).

The simple first explored feature was aggregating locations after noticing that the 120 unique locations actually comprised of 7 distinct locations on the globe. Locations were further clustered after noticing that some of them were geographically close and presented similar features and label distributions. The final 4 clusters are: Eastern Mexico, Atlantic Ocean, Pacific Ocean and Gulf of Mexico (Figure 1). Additional feature were explored from these clusters like location means, medians, max, min, etc. and secondary features such as difference from location mean.

The next simple features to explore were time-based features. To leverage the temporal aspect of the dataset, I extract year and month information from the '*time*' column. Additionally, I create a '*month_sin*' (equation 1) feature to capture the cyclic nature of months using trigonometric encoding. The year column was removed since the test set only included data from 2010 to 2013, making the years from 1996 to 2009 irrelevant as features. Next, to capture the historical context of the data, I create features based on recent data points. One of them is a feature that represents the average Total Moisture (TMQ) in the location over the last two weeks. This rolling mean is computed based on the 'location_cluster' to provide a sense of the evolving recent climate conditions. This feature had a good effect on predictive power.

Furthermore, as suggested by Lacombe et al. (2023), TC's are characterized by high wind speeds and rotation, I thus create a *wind velocity* feature based on the L_2 norm of zonal and meridional components of the wind vector field (equation 2). This feature ended up having a large predictive power.

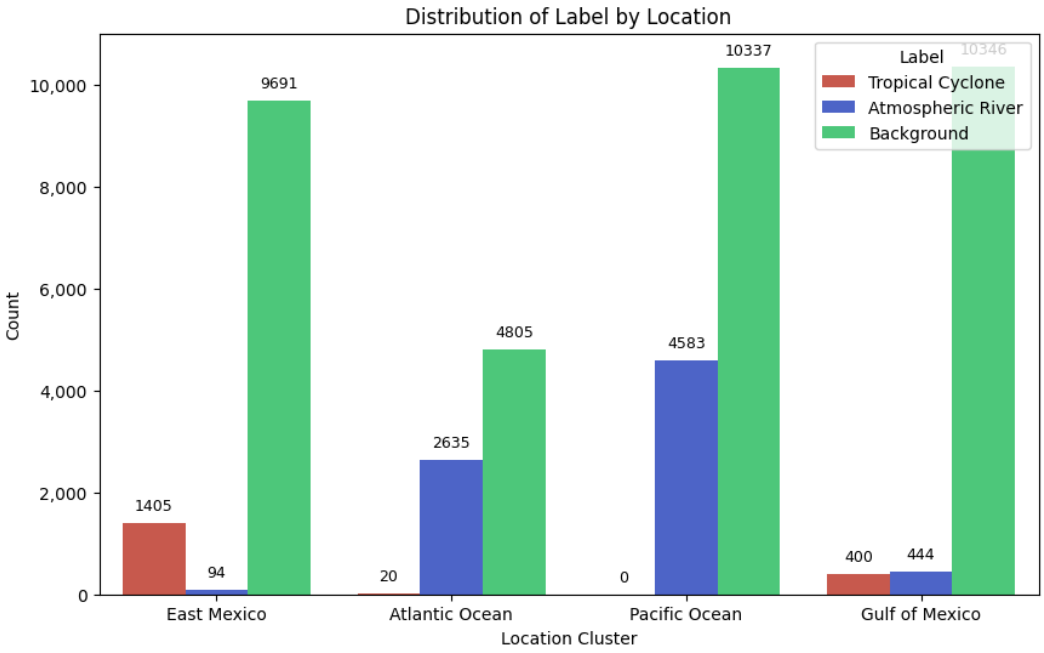

Figure 1: Distribution of labels based on the final clusters

Several features from the original dataset ended up being dropped due to their limited predictive power, irrelevance to the classification or redundancy with other features. This helped streamline the dataset and improve model training efficiency and speed. These include '*lat*', '*lon*', '*SNo*', '*time*', and other atmospheric features. The full metrics of feature engineering is presented in the results section.

3 Algorithms used

The classification task involved the exploration and application of various machine learning algorithms to achieve accurate predictions for extreme weather events. The algorithms employed in this study encompassed a diverse set of methods, each offering unique strengths and capabilities. The algorithms explored include:

Logistic Regression: Mutliclass logistic regression (implemented by hand as required in the report) served as the foundational algorithm in my analysis. It provided a simple yet effective method for multiclass classification. In further discussions, it is used as the baseline model with vanilla features of the ClimateNet dataset.

Neighborhood-Based Methods: Neighborhood-based methods involve making predictions based on the proximity of data points in the feature space. These methods were incorporated to assess their suitability for classifying extreme weather events.

Tree-Based Models: Tree-based models, including LGBM, XGBoost, and Random Forest, are a class of machine learning algorithms that are particularly well-suited for classification tasks. These models employ decision trees as their fundamental building blocks. Decision trees partition the feature space into regions and assign a class label based on the majority class within each region. Tree-based models, however, go beyond individual decision trees and operate by creating ensembles of trees.

1. Light Gradient Boosting (LGB): LGB, known for its exceptional speed and scalability, played a pivotal role in my experiments. Its ability to handle large datasets and manage class imbalances made it a valuable asset in my model selection.
2. Extreme Gradient Boosting (XGBoost): XGBoost, another ensemble method, was explored for its robustness and efficiency. Similar to LGBM, XGBoost uses decision trees as base

learners but incorporates regularization techniques to prevent overfitting and enhance model generalization.

3. Random Forest (RF): Random Forest, a popular ensemble method, was included in my model selection due to its strong performance and interpretability.

Ensembling, the practice of combining multiple models to enhance predictive performance, was also strategic choice in my approach. Ensembling leverages the diversity of different algorithms, harnessing their individual strengths, and mitigating their weaknesses. By combining the predictions of several models, I aim to create a more robust and accurate final model. The use of ensembling can help us capture intricate patterns within the data, improve generalization, and enhance the model's overall predictive performance. My ensembling method was simple and comprised of 1 LGBM model, 3 XGBoost models, 1 RF model, and the results of my baseline LR regression. The inclusion/exclusion of models in the ensembling was decided through performance on the validation set and variety of performance (*e.g.*, if a model performed particularily well at predicting a certain class). The ensembling method was simply taking the mode of the predictions. In the subsequent sections of this report, I will delve into the methodology, hyperparameter tuning, and experimental results for each of these algorithms, shedding light on their respective contributions to the predictive accuracy of my models.

4 Methodology

The reduced ClimateNet dataset is split into a **training set** spanning from 1996 to 2006 and a **validation set** from 2007 to 2009. I decided to use these splits instead of random splits to respect temporal consistency. By splitting the dataset based on years, I ensure that the model is trained on historical data and validated on more recent data. This mimics real-world scenarios as future data should not be seen during training. Note that this is different from the **test set** used for the Kaggle competition, which spans from 2010 and 2013. A random cross-validation architechture was deemed disfunctional as different folds of the data would end up with vastly varying label distribution, making for some excellent accuracy is some folds and very bad in some others. A time-based cross-validation strategy was also explored but ran into similar problems.

Instead, I pursued a more general approach. my methodology involved training models on the designated training set and evaluating their performance on the independent validation set. Notably, both hyperparameter tuning and feature engineering were carried out with the sole reliance on the validation set's performance. This approach assures that I consistently assess the models' **out-of-sample** performance, mitigating the risk of overfitting and providing a reliable indication of how well my models generalize to unseen data. In the context of the Kaggle competition, I must also be wary of **adaptive overfitting** where my model would work well only on the public test leaderboard, but badly generalize to other sets.

4.1 Hyperparamter tuning

Hyperparameter tuning is a critical step in my approach, as it fine-tunes the models to achieve their highest potential performance. Hyperparameter tuning for my tree-based models was executed through a systematic grid search. This process involved the exploration of various hyperparameter combinations to identify the optimal configuration that maximizes model performance. The specific hyperparameters tuned included learning rate, maximum tree depth, the number of estimators, and feature subsampling rates, among others. For each model—LGBM, XGBoost, and Random Forest—I performed a separate grid search. Grid search works by trying out every single combination of the parameters in my given grid and outputting the parameters that gave the best performance on the validation set. The objective was to find the hyperparameter values that resulted in the best validation set performance, as measured by the relevant metric in my competition which is accuracy. Other metrics like class-specific specificty and sensitivity were also considered to try and improve the classification of the less common classes (TC's and AR's). Models were then further fine-tuned by taking the given best parameters and trying out parameters around the given values. The outcomes of this tuning process are presented in the Results section of this report, along with a detailed analysis of the models' effectiveness in classifying extreme weather events.

Having far less parameters and thus smaller search space, Logistic Regression and K-NN model tuning was done by hand in an iterative fashion. Without the use a of a grid, I simply trained the model on the training set using various parameters and evaluated their performance on the validation set.

4.2 Oversampling

Oversampling was used to address class imbalance in the dataset. As previously mentionned, the dataset had a notable class imbalance. The trainig set comprised of 78.6% background (`'0'`), 4.1% TR's (`'1'`), and 17.3% AR's (`'2'`). This means that a model classifying everything as background would achieve a 78.6% accuracy. Given the rarity of extreme weather events, particularly tropical cyclones and atmospheric rivers, the class distribution was skewed and had different distributions for every location (Figure 1). To address this, I employed Synthetic Minority Over-sampling Technique (**SMOTE**). Lacombe et al. (2023) mentionned the weakness of oversampling due to the loss in geographical specific features. What set my approach apart, given the reduced locations dataset, was the targeted oversampling of specific location clusters, thus preserving the geographic characteristics of extreme events (e.g., the Pacific Ocean cluster has different behavior than the Atlantic Ocean cluster). This approach acknowledged the geographic clustering often observed in extreme weather occurrences, allowing us to amplify their representation while maintaining spatial integrity in the dataset.

The strategic oversampling process, as outlined, aimed to bolster the models' capacity to identify and classify extreme weather events within geographically concentrated regions. It's important to emphasize that oversampling was applied exclusively to the training set and not to the validation set. This distinction was made to prevent data leakage and ensure that the models Ire evaluated on their ability to generalize to unseen data. The impact of this technique on model performance and its role in mitigating class imbalance is discussed in subsequent the results section, shedding light on its contribution to the overall effectiveness of my classification models. Different **SMOTE** parameters were explored such as class weights, sampling method and tuned in a manner similar to the hyperparameter tuning.

5 Results

In this section, I present the results of my experiments, including a comparison of the performance of different machine learning models with varying hyperparameters. I also provide visual representations of the results for a more comprehensive analysis.

5.1 Model Analysis

I explored multiple machine learning models, namely LightGBM, XGBoost, RF, k-NN, and Logistic Regression, with various hyperparameter settings to evaluate their performance on my dataset. The summary of these models is presented in Table 1. I do not report class 0 (background) metrics as the class of interests in this report are TC's and AR's and were fairly similar for all models.

To provide a benchmark, the baseline logistic regression model demonstrated relatively strong performance even with the vanilla features. This suggests that the logistic regression model, even in its basic form, is capable of making meaningful predictions on the data. This observation hints at the potential for machine learning models to learn from it effectively.

Among the single models evaluated, the tuned LightGBM (LGBM) model achieved the highest accuracy at 82.39%. However, it's important to interpret this accuracy with caution due to the presence of class imbalance in the dataset. The high accuracy might be influenced by the dominant class, making it essential to consider other evaluation metrics.

Sensitivity measures a model's ability to correctly identify a specific event. For TC's, the tuned XGBoost model exhibited the highest TC sensitivity at 62.62%, indicating its effectiveness in identifying TC events. Similarly, for ARs, the tuned LGBM model excelled with a sensitivity of 68.02%, highlighting its capability in detecting AR events. These findings suggest that the possibility of creating an ensemble model, which combines the strengths of different models, could be promising for the final model.

			Feature Engineering & Oversampling					
Metric	**LR**	**LGBM**	**k-NN**	**LGBM**	**LGBM***	**XGBoost***	**RF***	**Ensemble**
Accuracy	0.7890	0.7890	0.7132	0.8058	0.8239	0.8234	0.8100	**0.8265**
TC Sensitivity	0.3615	0.6203	0.2957	0.5079	0.5845	**0.6262**	0.6865	0.6410
TC Specificity	0.9442	0.9602	0.9477	0.9568	0.9602	**0.9607**	0.9485	0.9592
AR Sensitivity	0.5716	0.5245	0.3427	0.6365	0.6802	0.6671	0.6823	**0.6836**
AR Specificity	**0.9092**	0.8882	0.8715	0.8960	0.8907	0.8906	0.8806	0.8971

* Represents hyperparameter tuning

Table 1: Model Performance Comparison

Specificity gauges a model's ability to correctly identify instances that do not belong to a specific event. Regarding TC's, given their rarity and distinct characteristics compared to background and ARs, it is understandable that all models achieved high TC specificity. In the case of ARs, their identification is more challenging, as many of their characteristics overlap with background samples. Consequently, models exhibited lower AR specificity, reflecting the difficulty of distinguishing ARs from background events.

The ensemble model, which combines the strengths of multiple models, deserves special attention and is the model used for the final Kaggle submission. It achieves the best accuracy of 82.65% and the best AR Sensitivity, acknowledging the strenght of ensembling multiple different models. It should be noted that ensembling offers the potential to improve overall performance by leveraging the complementary strengths of individual models while buffering their weaknesses. Further exploration and fine-tuning of the ensemble approach (ensembling methods, more or less models, etc.) could yield even better results.

5.2 Oversampling

Location-based oversampling had mitigated effects on performance. For the identification of Tropical Cyclones (TC's), oversampling was notably beneficial. TC's, being rare events, were easier to identify when the TC class was oversampled, resulting in improved sensitivity. However, for Atmospheric Rivers (ARs), which share similar characteristics with the background class, oversampling led to mixed results. While it improved sensitivity, it also increased the risk of misclassification, reducing specificity. These findings emphasize the need for a nuanced approach to dealing with class imbalance, tailoring strategies to the unique characteristics of different classes. Future work could focus on fine-tuning oversampling techniques to strike a better balance between sensitivity and specificity, particularly when working with classes exhibiting varying degrees of similarity to the majority class.

6 Discussion / Future Work

In the analysis, we explored various machine learning models, including LightGBM, XGBoost, RF, k-NN, and Logistic Regression, to evaluate their performance on our dataset. We focused on Tropical Cyclones (TC's) and Atmospheric Rivers (ARs), aligning with the project's objectives. One strength of our approach was the comprehensive assessment of multiple models. We also included a baseline logistic regression model, providing a useful reference point. Additionally, we considered hyperparameter tuning to optimize model performance.

However, there are areas for improvement. Addressing class imbalance in other ways, model interpretability, and delving deeper into ensemble techniques are potential next steps. Additionally, a weakness of our approach is strenght of our engineered features. Though an extensive amount of time was spent on feature engineering and the ones created had a satisfactory effect on performance, some other features could be explored such as other geographical features, time-based features, and other meta features to improve performance even further.

In future work, a key focus should be on addressing the mixed effects of oversampling on class-specific performance. A nuanced approach to oversampling, accounting for the distinct characteristics of different classes, can lead to improved strategies for tackling class imbalance. Additionally, enhancing model interpretability through techniques like feature importance analysis is crucial for making models more transparent and practical for real-world applications.

7 Statement of Contributions

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

References

Lacombe, R., Grossman, H., Hendren, L., and Lüdeke, D. (2023). Improving extreme weather events detection with light-weight neural networks. Accessed on arXiv 11/3/2023.

Prabhat, Kashinath, K., Mudigonda, M., Kim, S., Kapp-Schwoerer, L., Graubner, A., Karaismailoglu, E., von Kleist, L., Kurth, T., Greiner, A., Mahesh, A., Yang, K., Lewis, C., Chen, J., Lou, A., Chandran, S., Toms, B., Chapman, W., Dagon, K., Shields, C. A., O'Brien, T., Wehner, M., and Collins, W. (2021). Climatenet: an expert-labeled open dataset and deep learning architecture for enabling high-precision analyses of extreme weather. *Geoscientific Model Development*, 14(1):107–124.

Appendix

Feature	Description	Units
lat	Latitude	°
lon	Longitude	°
TMQ	Total (vertically integrated) precipitable water	kg/m^2
U850	Zonal wing at 850 mbar pressure surface	m/s
V850	Meridional wind at 850 mbar pressure surface	m/s
UBOT	LoIst level zonal wind	m/s
VBOT	LoIst model level meridional wind	m/s
QREFHT	Reference height humidity	kg/kg
PS	Surface pressure	Pa
PSL	Sea level pressure	Pa
T200	Temperature at 200 mbar pressure surface	K
T500	Temperature at 500 mbar pressure surface	K
PRECT	Total (convective and large-scale) precipitation rate (liq + ice)	m/s
TS	Surface temperature (radiative)	K
TREFHT	Reference height temperature	K
Z1000	Geopotential Z at 1000 mbar pressure surface	m
Z200	Geopotential Z at 200 mbar pressure surface	m
ZBOT	Lowest modal level height	m
LABEL	0: Background, 1: Tropical Cyclone, 2: Atmospheric River	-

Table 2: ClimateNet dataset features and labels

Feature	Description	Units
wind_velocity_850	Wind speed (equation 1) at 850 mbar pressure surface	m/s
avg_TMQ_location	Average TMQ at a given location	kg/m^2
avg_TMQ_location_period	Average TMQ at a given location in a given period	kg/m^2
diff_from_avg_TMQ	Difference from avg_TMQ_location	kg/m^2
month_sin	Cyclical labelling of current month (equation 2)	-
season	Season (0-3)	-

Table 3: Engineered features and labels

This equation represents the magnitude of the wind velocity, combining the zonal and meridional components. It's a fundamental equation in meteorology for calculating wind speed. The wind velocity (V) can be calculated using the zonal wind (u) and meridional wind (v) as follows:

$$V = \sqrt{u^2 + v^2} \tag{1}$$

The month_sin (MS) feature is calculated to encode the cyclic nature of months in a year using the sine function. It is commonly used in time series data to capture seasonal patterns. The formula for calculating month_sin is as follows:

$$MS = sin(\frac{2 * \pi * \text{month}}{12}) \tag{2}$$