Anales del Instituto de Actuarios Españoles, 4ª época, 28, 2022/149-168
ISSN: 0534-3232 / eISSN: 2531-2308/ D.L.: M-3160-1961 / DOI: 10.26360/2022_6

CLASSIFICATION OF CLIMATE-RELATED INSURANCE CLAIMS USING GRADIENT BOOSTING

CLASIFICACIÓN DE SINIESTROS DE SEGUROS RELACIONADOS CON EL CLIMA MEDIANTE EL USO DE GRADIENT BOOSTING

George Tzougas[1], Viet Dang[2], Asif John[3],
Stathis Kroustalis[4], Debashish Dey[5], Konstantin Kutzkov[6]

[1]*Department of Actuarial Mathematics and Statistics, Heriot-Watt University, Edinburgh, EH14 4AS, UK.* George.Tzougas@hw.ac.uk
[2 3]*Ar Genesis:* https://argenesis.com/
[4]*Department of Statistics, Athens University of Economics and Business, Athens, 76 Patission str Greece.*
[5]*WTW, 51 Lime Street, London, EC3M 7DQ, UK*
[6]*Teva Pharmaceuticals:* https://www.tevapharm.com/

Fecha de recepción: 18/11/2022
Fecha de aceptación: 07/12/2022

Abstract

The aim of this paper is to implement, one of the most representative supervised learning approaches, the decision tree based ensemble method called gradient boosting for classifying the number of claims caused by storms in Greece using data from a major insurance company operating in Greece. Finally, a machine learning algorithm is used to for categorising the number of claims which have been occurred by a "storm event" into 3 categories: "no claims", "1 claim", "2 or more claims".

Keywords: cimate-related insurance claims, ensemble learning, decision trees, boosting.

Resumen

El objetivo de este trabajo es aplicar uno de los enfoques de aprendizaje supervisado más representativos, el método de conjunto basado en árboles de

decisión denominado gradient boosting, para clasificar el número de siniestros causados por tormentas en Grecia utilizando datos de una importante compañía de seguros que opera en este país. Por último, se utiliza un algoritmo de aprendizaje automático para clasificar el número de siniestros que se han producido como consecuencia de un "evento de tormenta" en 3 categorías: "ningún siniestro", "1 siniestro", "2 o más siniestros".

Palabras clave: siniestros de seguros relacionados con las tormentas, aprendizaje conjunto, árboles de decisión, boosting.

1. Introduction

Weather-related phenomena such as floods, windstorms, hailstorms and wildfires can cause extensive financial losses. Insurance companies can find it challenging to distinguish between the impacts of the underlying meteorological perils for each given event and claim due to various factors including complexity and the business cost involved. The individual impacts of flood, strong winds and hail are often combined and classified into a single cause for an insurance claim such as a “storm event”. To better price and reserve for future perils, it could be beneficial for insurers to distinguish between these underlying components even when within the same insurance claim. Providing a business solution to tackling this problem is important because by having a better model and understanding of the underlying perils for a claim, an insurer can achieve better pricing and reserving whilst policyholders can pay a fairer price for the policy that they buy. Insurers are expanding traditional techniques of analysis, for example by incorporating data science techniques, to hopefully improve models, leverage larger sets of data, “unlock” hidden data patterns and free up resources within the team, where currently a lot of the data, data cleansing and experience analysis is driven by people within the team. Given this and the potential large amounts of data involved, using machine learning approaches seems optimal to analyse such large sets of insurance, geographical and meteorological data.

In recent years, there have been various non-life insurance studies related to climate risks. Below, we share an example spread of climate-risk work - though please do note that this is not meant to be a definitive and exhaustive list of such works. Starting from Yang *et al.* (2022), the focus lies on the increasing frequency and severity of flood risk due to climate change which could have the potential to increase homeowner insurance claims. With this in mind, they build a “random forest” model with two steps: i. damage level classification and claim number classification; and ii. sub-sampling strategies, using a combination of

spatial, topography, land and meteorological data. In Lyubchich, Newlands, Ghahari & Gel (2019) one can find a thorough review of recent advancements on weather-related risk modelling and assessment for agricultural and home insurances using various statistical and machine learning methodologies. For instance, Ahmed & Serra (2015) uses statistical copulas to identify how by introducing agricultural revenue insurance contracts in Spain, this would influence insurance premiums compared to yield insurance schemes. Moreover, Choudhury, Jones, Okine & Choudhury (2016) builds a multiple classification and autoregressive error model whilst they also perform a model-based cluster analysis to identify what signals would trigger a drought-insurance payout. Also, Tack & Ubilava (2015) is concerned with the effect of climate tele-connections, such as the El Ni˜no southern oscillation on US government-set premium rate pricing for cotton, using a moment-based maximum entropy modelling approach indicating that private insurers would have the potential to decrease claims paid by 10-15 percent on average. The following article Rohrbeck, Eastoe, Frigessi & Tawn (2018) accounts for the dependence between various weather events such as rainfall or snow-melt, and the count of water-related property insurance claims aiming for modelling large claims counts within a mixture and extremal mixture modelling statistical framework being based on a discretised generalised Pareto distribution. Moreover, they develop a temporal clustering algorithm for the derivation of new explanatory variables in an effort to better comprehend how claims and weather-related events relate. The article of Mobley, Sebastian, Blessing, Highfield, Stearns & Brody (2021) highlights the importance of using machine learning methods to estimate flood hazards across large geographical territories in a computationally cost effective way, where they build a random forest model for classification in order to predict flood probability across the Texas Gulf Coast region using a large National Flood Insurance Program (NFIP) insurance claims geospatial dataset.

The article of Leblois, & Quirion (2013) gives a summary of the methods and challenges faced by weather-based insurance indices, i.e. rainfall, water stress, drought, and remotely sensed vegetation indices suggesting that focusing on past data does not allow for accurately accounting for complex non-stationarities in space and time associated with events of high temperature or rainfall. The article of Knighton, Buchanan, Guzman, Elliott, White & Rahm (2020) develops a random forest to predict parcel-level and tract-level flood insurance claims within New York State based on a US hydrologic and social demographic dataset for flood exposure mapping. In addition to weather risks affecting spatial and temporal correlation of insurance indices and their covariates, climatic change also can increase the volatility of weather variables, generating non-stationary loss distributions, which are challenging to estimate

reliably, adding uncertainty to actuarial rate-making as in the article of Odening & Shen (2014). The article of Daron & Stainforth (2014) develops Bayesian belief networks for helping insurers to assess weather index insurance viability in developing countries from a climate viewpoint. The article of Tesselaar, Wouter Botzen & Aerts (2020) examines how vulnerable EU river line flood insurance systems are to global reinsurance market conditions, i.e. hard and soft markets, and climate change by applying a proprietary flood insurance model.

As we have seen in the aforementioned literature, the use of "ensemble learning" methods using "decision trees" as "base learners" for studying climate-related claims in non-life insurance is a popular option, see for instance Yang *et al.* (2022), Mobley, Sebastian, Blessing, Highfield, Stearns & Brody (2021), and Knighton *et al.* (2020). However, the focus has been predominantly on "random forest" methodology and other powerful ensemble learning approaches such as boosting were not explored. In this article, we adopt a simplified approach and suggest a well-known decision tree based ensemble method called gradient boosting to classify the number of claims caused by storms in Greece, where the data has been taken from a local insurer in Greece, and finally use a machine learning algorithm to help categorise the number of claims which have been occurred by a "storm event" into 3 buckets: "no claims", "1 claim", "2 or more claims". In particular, the "gradient boosting" process we use in this article is described in this section involves the following steps. Firstly, we fit a "shallow decision tree" on to the data. By "shallow" we mean a "decision tree" with very few splits on the predictors with highest predictive power. These splits are determined by a heuristic approach, as finding the optimal set of splits is a computationally difficult problem Hyafil & Rivest (1976). These trees are so called weak learners and give just a lower prediction error rate than the one that would result from random guessing. This has to do with the main idea behind ensemble modelling, see for example Breiman (1996), which is to build a number of weak models and then combine them into a stronger one in order to reduce bias, and increase model accuracy. After we have fit a "shallow decision tree" to the data, we then fit the next "decision tree" to the residuals of the first one. The residuals are the differences between the prediction and the actual class labels, and thus the second decision tree addresses the shortcomings of the first one. Then, we add this second tree into the algorithm. We continue the process and fit a third tree to the residuals of the second tree and then we proceed by adding the third tree into the algorithm and so on. The process stops when adding more decision trees does not result in a more "powerful" model. Each individual model is weighted according to some criterion and the ensemble is a "stage-wise additive" model (i.e. adding "decision trees" sequentially of all individual trees. The structure of the research paper is the

following. In Section 2, we discuss some relevant machine learning techniques. In Section 3, we present the novel dataset and in Section 4 the machine learning method we use to solve our classification problem. In Section 5, we present the results and in Section 6 we provide some remarks and future research ideas.

2. Machine learning preliminaries

Interest and use of machine learning techniques are increasing each day - whether in our daily lives for example via smart technology, or in business, for example with a greater focus from insurers, and they look to incorporate such techniques to reduce internal costs and improve consumer experiences. Before we continue with our analysis, in this section, we will introduce some machine learning concepts that will hopefully be useful for the reader to understand the approaches used by us. The explanations given here are not meant to be complete nor exhaustive, but more to give the reader a better overview of the methods used and hence how and why they fit into this study.

2.1. "Supervised" and "Unsupervised" learning

"Supervised" and "unsupervised" learning are 2 fundamental approaches which we can take in machine learning. "Supervised learning" builds models where the dataset is labelled, and has input data and out-put variables. Hence, "supervised learning" can be used to calculate outcomes e.g. for future predictions of values based on historic data. "Unsupervised learning" builds models where the dataset outputs are not labelled, and hence can be used to find "hidden" patterns and relationships within the data e.g. for categorisation problems. Generally, "supervised learning" is computationally simpler, produces a more accurate output but requires external supervision to "train" (i.e. construct) the model when compared to "unsupervised learning". However, it also requires more human intervention and as such may potentially cost more to implement. For more on general background to both approaches, please see for example www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning

2.2. Algorithms and datasets

A typical process within machine learning is to use data to develop a model which hopefully mimics the outcome required, within a certain level of accuracy/materiality threshold. The original dataset itself is split into a "train"

dataset and "test" dataset, where the "train" dataset is the larger proportion of the data. The "train" dataset is used to develop an initial model, where the model is based on an underlying individual algorithm/set of algorithms, usually via an iterative process until a certain level of accuracy is reached. Once the model is "trained", we use the "test" dataset to check the accuracy of the model, and ensure that the model has not been "overtrained" and biased to the input data originally used. There is a question of which algorithm to use: we may choose a range of algorithms which we initially feel are appropriate for the problem we are analysing, and compare the accuracy of the outputs of each algorithm before decision which to choose.

2.3. Ensemble learning

Above we mentioned the use of algorithms within machine learning. Ensemble learning is a collection of models which tend to produce a more accurate output i.e. "the whole is greater than the sum of the parts". For more details of merging predictions in such a manner please see for example Breiman (1996), Clemen (1989), Perrone (1993) and Wolpert (1992). The building blocks within this framework are called "base learners". Ensemble learning takes a "blend" or "average" of these "base learners" to hopefully remove any inherent bias produced by an individual "base learner" and to produce an overall better output. One popular base learner is a decision tree and it is discussed further below. For more background on "ensemble learning", including some background to mathematical detail, please see Dietterich *et al.* (2002).

2.4. Decision trees

A "decision tree" is a "supervised learning" approach where the output is decided based on a series of decisions at each "node" - similar to a branches on a tree where the decisions can branch out. For more background, include graphical explanations of "decision trees" please see Charbuty & Abdulazeez (2021). We present a mathematical formulation of a classification decision tree. Suppose that we have a collection of n observations, each of which consists of a response variable and k predictors. For the ith observation, we denote it with the notation (yi, xi1, xi2, ..., xip). To ease notation, we will also denote the p-dimensional vector of predictor (xi1, ..., xip) by xi. For the purpose of this paper, we assume that yi is categorical and can take up to K distinct values, which we shall denote by $\{0, 1, \cdot \cdot \cdot, K - 1\}$. Any of the predictors can be either continuous and numeric, or categorical. For ease of

exposition, let us assume that they are all continuous and generalise to the categorical case later.

The goal of decision tree is to split the feature space into m non-overlapping regions, denoted R1, ...Rm, such that all observations within the same region will have the same prediction. However - given the enormous number of ways one can partition the feature space, one needs to restrict the shape that these regions can take. One simple way to do that is to restrict the regions Ri to be a *cylinder set*, i.e. sets of the form:

$$\{\mathbf{x} \in \mathbb{R} | L_1 \leq x_1 \leq U_1, L_2 \leq x_2 \leq U_2, ..., L_p \leq x_p \leq U_p\}, \quad (1)$$

where all Li, Ui belongs to [−∞, ∞]. We also allow any of the inequalities to be strict or non-strict. As it turns out, even under this restriction, to find in some sense an optimal collection of cylinder sets would present an impossible computational challenge. Therefore, we will need to restrict also the procedures by which these cylinder sets might arise. We will describe the procedure here in a somewhat informal way. We start with the entire feature space, i.e, the set of all possible values for the predictors, which in this case is simply Rp. We will now find an index-threshold pair (l, t), where t ∈ {1, ..., p} and t ∈ R and split Rp into two sets:

$$R_1\{x \in \mathbb{R}^p | x_l \leq t_l\}, \quad R_2\{\mathbb{R}^p - R_1\}. \quad (2)$$

All observations in R1 (respectively, R2) should have the same prediction. In the case where the response is categorical, we take the estimate to be the most commonly occurring value. Formally, we define:

$$\hat{\mu}_1 = \text{Mode}\{y_i | \mathbf{x}_i \in R_i\} \quad (3)$$

μ2 is defined similarly.

In order to optimise the choice of (l, t), we need some notion of optimality, a mechanism that punishes 'bad' estimates. Intuitively, a leaf node is good when it correctly classifies a large proportion of the responses, i.e. they coincide with the *modal class*. We say that this region is *pure* and seek to minimise the *impurity* of this region. Given a region R and an estimate ˆμ for the modal class, by an *impurity* function we mean a mapping Q(R, ˆμ) that, informally speaking, is increasing with the *impurity* of this region. We shall

minimise the following weighted sum (we abuse notations where and write, for example, |R1| for the number of observations with predictors in R1):

$$L(l, l_t, R)\frac{|R_1|}{|R|}Q(R_1, \hat{\mu}_1) + \frac{|R_2|}{|R|}Q(R_2, \hat{\mu}_1) \tag{4}$$

Thus, the optimal (l∗, t∗) is defined by:

$$(l^*, t^*)argmin_{(l,t)}L(l, l_t)$$

There are a number of candidates loss functions that can play this role. Given a region R and a class k, we define:

$$\hat{p}(R, k)\frac{1}{|R|}\sum_{y \subset R}\mathbb{I}_{\{y=k\}}, \tag{5}$$

the proportion of class k observations in class m. One natural and easily interpretable candidate is the misclassification error rate.

$$E(R, k) = 1 - \max_k \hat{p}(R, k). \tag{6}$$

Two other options are the Gini Index and the Cross Entropy (denoted G and D below, respectively). It can be shown that these functions will tend to zero as ˆp(R, k) is either really close to 0 or 1, implying a high level of class purity.

$$G(R, k) = \sum_{k=1}^{K}\hat{p}(R, k)(1 - \hat{p}(R, k)) \tag{7}$$

$$D(R, k) = -\sum_{k=1}^{K}\hat{p}(R, k)\log(\hat{p}(R, k)) \tag{8}$$

Let us now turn our attention to the leaf node R1. At this stage, we will go ahead and split each of the leaf nodes into further regions. For instance, at R1, we will search for another pair (h, th) that splits the R1 into two regions:

$$R_{1,1}\{x \in R_1 | x_h \leq l_h\}, \quad R_{1,2}\{\mathbb{R}_1 - R_{1,1}\}. \tag{9}$$

With ˆμ1,1 and ˆμ2,1 defined as above, we split R1 by minimising L(l, t, R1). After this step, R1 will no- longer be the leaf node. Rather R1,1 and R1,2 will

become the leaf node. R2 will be split similarly. At each iteration, we can split a leaf node into two leaf nodes, and each of them into two further leaf nodes, hence the name 'recursive binary splitting'. At some point, we will want to terminate this splitting process. Clearly, it is impossible to split a region that has only one observation. However, such a small region would imply a very finely divided and overfitted tree, and we want to stop the process some time before that. Thus, we implement a *terminating rule* that inhibits a leaf node from splitting. Two simple terminating rules are: 1. *minimal leaf size*, which disallows splitting once a leaf node falls below a certain size and 2. *maxmium tree depth*, which terminates the algorithm once the tree has reached a certain number of layeres.

Generalisation to Categorical Predictors. Now, let us assume that some of the predictors are categorical. For the sake of concreteness, suppose that x1 is categorical with L different levels. The feature space is now no longer Rp. Let us denote it by Θ. Then given any region R, if we attempt to split it by x1, we will search for a pair of index-subset (l, T), where l ∈ {1, .., p} and T is a subset of {1, ..., L} and divide R into two sub-regions:

$$R_1 \{\mathbf{x} \in \Theta | x_1 \in T\}, \quad R_2 R - R_1.$$

The rest of the algorithm proceeds exactly as before.

2.5. Boosting

The general idea behind "boosting" is that the "base learners" within an "ensemble" are developed sequentially in a way that each of them focuses on the data which produced worse predictions to now make more accurate predictions, such that the accuracy of the overall model increases. Hence, this should reduce the bias and variance of the outputs. The "base learners" are normally "decision trees" which are constructed through an iterative process namely "recursive partitioning", see Breiman, Friedman, Olshen & Stone (1984), which splits the data into an initial partition, and then splits this partition into smaller groups with the process continuing up until a stopping criterion is reached. There are many variations of boosting and gradient boosting, see Friedman, Hastie & Tibshirani (2001) and Friedman (2001) across a plethora of applications. The underlying philosophy is that we can minimise the overall prediction error by combining the best possible next base learner to the previous ones, see for example Freund, Schapire & Abe (1999).

The additive mechanism of boosting implies that it is suited for prediction of continuous values. Thus, for classification trees, we must find a connection between the categorical outcome, and the continuous values returned by the additive ensemble method. The technique here is borrowed from GLM: instead of predicting the categorical outcome, we will predict the probabilities of the outcomes. Moreover, at each step, the deviance residual is utilised.

To simplify the exposition, let us consider the case where the response can take only two values: 0 and 1. In this case, the response can be interpreted as coming from a Bernoulli distribution, whose parameters depending on the values of the predictors (or rather pedantically, the region in the feature space it belongs to). We digress briefly to recall some facts about the logistic regression model, the classic binary outcome GLM. In this model, the connection between the Bernoulli success probability and the predictors is established canonically by considering the log-odd, i.e.:

$$\log\left(\frac{\pi}{1-\pi}\right) = \eta,$$

where η is the linear predictor. Moreover, π can be retrieved by inverting the relationship above to π = σ(η), where

$$\sigma : \eta \to \frac{e^{\eta}}{e^{\eta}+1}$$

is the *sigmoid function*.

Firstly, we consider a tree stump where all observations are predicted the same log-odds. That is:

$$\hat{\pi}_{\text{ini}} \frac{\sum_i y_i}{n}, \quad F_0(x_i) = \log\left(\frac{\hat{\pi}_{\text{ini}}}{1-\hat{\pi}_{\text{ini}}}\right).$$

Before the incrementing step, we need to borrow another concept from GLM. For a logistic regression mode, given a prediction ˆπi for each observation, the total deviance residuals is given by:

$$D(y_i, \hat{\pi}_i) = \left\{ y_i \log(\hat{\pi}_i) + (1-y_i)\log(1-\hat{\pi}_i) \right\}.$$

At each step, F0(xi) is incremented in such a way that minimises the deviance. Mathematically, for m = 1, · · · , M , we fit a tree hm such that:

$$h_m = \text{argmin}_h \sum_{i-1}^{n} D\left(y_{i,}, \sigma\left[F_{m-1}(x_i) + h_m(x_i)\right]\right),$$

and define Fm = Fm−1 + hm. The probability of yi belonging to class 1 is σ(FM (xi)).

Multiclass Classification Case. Let us now turn into the case where the response can take an arbitrary number of values. For concreteness, suppose there are K possible levels, denoted {0, 1, ..., K−1}. We observe that, in the binary classification case, the log-odds can be interpreted as follows:

$$\log\left(\frac{\pi}{1-\pi}\right) = \log\left(\frac{\mathbb{P}(Y=1)}{\mathbb{P}(Y=0)}\right).$$

The broad idea for this case is to treat these K − 1 log-odds independently and build K − 1 corresponding boosted ensemble, before combining them. Since the general framework has been described in details in the binary case, we take the liberty to describe this case rather briefly.

First, define for j = 1, · · · , K – 1

$$\hat{\pi}_{\text{ini}}^{(j)} = \frac{\sum_i \mathbb{I}_{\{y_i=j\}}}{n}, \quad F_0^{(j)}(x_i) = \log\left(\frac{\pi_{\text{ini}}^{(j)}}{1 - \sum_{l-1}^{K-1} \pi_{\text{ini}}^{(l)}}\right)$$

For m = 1, · · ·, M , define

$$h_m^{(j)} = \text{argmin}_h \sum_{i-1}^{n} D\left(y_i, \sigma\left[F_{m}^{(j)}{}_{-1}(x_i) + h(x_i)\right]\right)$$

$$F_m^{(j)}(x_i) = F_{m-1}^{(j)}(x_i) + h_m^{(j)}(x_i)$$

At the end, we obtain the following system of K − 1 equations:

$$P(Y_i = 1|x_i) = P(Y_i = 0|x_i)e^{F_M^{(1)}(x_i)}$$
$$\vdots$$
$$P(Y_i = K-1|x_i) = P(Y_i = 0|x_i)e^{F_M^{(K-1)}(x_i)},$$

which can be simplified thanks to the relation

$$\sum_{k=0}^{K-1} P(Y_i = k|x_i) = 1.$$

this reduces to:

$$P(Y_i = 0|x_i) = \frac{1}{1 + \sum_{k-1}^{K-1} e^{F(k)(x_i)}}$$
$$P(Y_i = 0|x_j) = \frac{e^{F(j)(x_i)}}{1 + \sum_{k-1}^{K-1} e^{F(k)(x_i)}}, \quad j = 1, \cdots, K-1.$$

One form of boosting is Gradient Boosting. The ensemble here is made up of "base learners" where each "base learner" is a "decision tree". The "gradient boosting" element of this process refers to the underlying algorithm (a "gradient descent" algorithm) which is used reduce the loss with each incremental additional to the overall "ensemble" model. It is a common choice for classification problems to combine "decision trees" with "gradient boosting" because of the scalability of the approach and optimal results it yields on tabular data, and currently, this seems popular in industry to use "gradient boosting". For more details on "gradient boosting", and detail of the open-source software used, including more mathematical background, please see for example here Ke *et al.* (2017).

2.6. Model

Starting from a Greek P&C insurance dataset $D = \{x_i, y_i\}_{i=1}^{N}$, we use gradient boosting to find an approximation, $\hat{F}$(x), of a function F^*(x), which is the mapping from instances x to their output values y, via minimizing the expected value of a given loss function, L(y, F(x)). In our case, we solve a multiclass classification problem with the three classes presented in Figure 2. The loss function is categorical cross–entropy defined as

$$\sum_{i-1}^{N} -y_i \log \tilde{y}_i$$

where ˜yi = Pr[F (xi) == yi] is the probability that example xi belongs to class yi. The probabilities are computed as described in the previous section. Note that the cross–entropy function is convex and the gradient with respect to each tree can be efficiently computed. By using gradient boosting, we develop an additive approximation of F ∗(x) as a weighted sum of functions,

$$F_k(x) = F_{k-1}(x) + \rho_k g_k(x) \tag{10}$$

where ρ_k reflects the weight of the kth function, g_k(x). It is worth noting that these functions are base learners within a model ensemble and these base learners, in our case, are decision trees, see Breiman, Friedman, Olshen & Stone (1984). The construction of this additive approximation of F(x) happens iteratively. In particular, firstly, we obtain a constant approximation of F^*(x) as

$$F_0(x) =_{\alpha} \sum_{i=1}^{N} L\,(y_i, \alpha) \tag{11}$$

and then the succeeding models need to minimize

$$(\rho_k, g_k(x) = \ _{\rho,g} \ \sum_{i=1}^{N} L\,(y_i, F_{k-1}(x_i) + \rho g(x_i)). \tag{12}$$

The value α is constant in the sense that it is independent of the input features xi. In fact, the optimal constant α optimizes the loss function for the given label distribution. In the case of categorical cross-entropy, α is the proportional distribution of the labels yi, i.e.,

$$\alpha_c = \sum_{i-1}^{N} (y_i == c)/N$$

for each class c.

Nevertheless, the aforementioned optimization problem is not solved directly. In particular, we train each model g_k on a new dataset $D' = \{x_i, r_{ki}\}_{i=1}^{N}$, where the residuals, r_{ki}, are computed by

$$r_{k,i} = \left[\frac{\partial L(y_i, F(x))}{\partial F(x)}\right]_{F(x)=F_{k-1}(x)} \tag{13}$$

and then the value of ρ_k is subsequently calculated using a line search optimization strategy. Of course, as with other machine learning algorithms the prediction performance of gradient boosting ensemble is assessed using a separate test set with unseen data points or using cross validation techniques. An important advantage of gradient boosting compared to other methods such as logistic classification or a neural network model is that we don't need careful feature scaling which is appealing given our dataset. A scheme of the gradient boosting method for our classification problem is presented in Figure 2. The gradient boosting modelling results for the Greek P&C dataset follow in the next section.

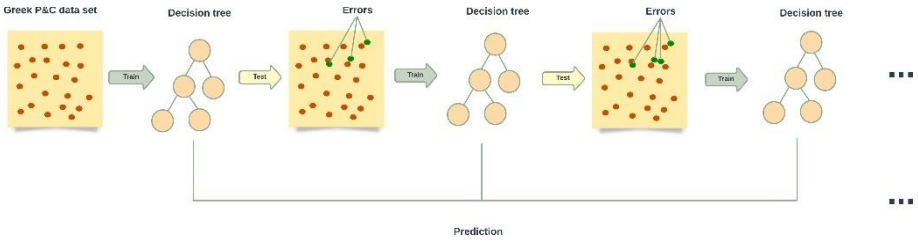

Figure 1. Basic diagram to illustrate the process of "gradient boosting" used in this article.

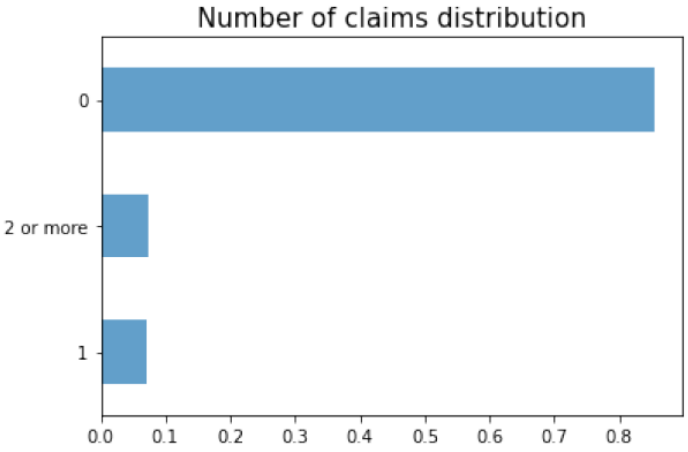

Figure 2. Number of claims distribution.

3. Data

We have taken annual data from a local insurer in Greece, from different insurer branch locations across Greece. The data consists of over 60 "columns" of data. There was no material manipulation of the data before training - this is an advantage of using "decision trees" and "gradient boosting". There are around 30 columns of data used for classification. Each branch is described by a number of features including meteorological conditions, the number of insurances for different types, and in particular the number of events with wind or rain during the year. Also, we have the number of claims as a feature. Again, these features are for each branch, aggregated over a full year. There are no timestamps in the data.

Specifically, the dataset represents the filed insurance claims as a result of weather anomalies in different geographic areas in Greece, represented by their postcodes, between the years 2012 and 2019 for a specific local insurer. There are 5,216 unique postcodes, and in each postcode one or more insurer branches can be present. In total, there are six branch categories that handle different accidents such as properties, cargo, motors, etc. In total, there are 36,939 examples. Each such example represents the number of claims in a given branch with existing insurance policies, at a given postcode, for a given year. For example, in 2017 at postcode 41335 there are two insurance claims for damaged properties, one for a motor vehicle and no filed claims for yachts or cargo. Figure 2 represents the number of claims from such the dataset. As it can be seen, the dataset is highly imbalanced, with 85.5% of the values being 0 (no claims), 7% with a single claim and 7.5% with two or more claims, see Figure 2. In total, the data has 88 such features. (Note that many features are one-hot encoded categorical features. For example, the CORINE land cover classification introduces 44 unique features.) The postcodes are described by a number of features that can be divided into several categories:

- Geographic features. These features describe the postcode area and contain information such as mean, minimum and maximum altitude, the length of coast, surface roughness and slope, etc. Also, the CORINE land cover codes of the area, see European Environment Agency (2022).
- Meteorological features. Information about the weather conditions such as wind and rain intensity, flood vulnerability index, etc.
- Insurance features. The number and different kinds of insurance policies.

4. Results

We use the Python framework LightGBM (Ke *et al.*, 2017), which is free and open source. It was developed inpart by Microsoft. LightGBM is a gradient boosting framework that uses tree based learning algorithms, according to their main Github page here: https://github.com/microsoft/LightGBM. For more commentary on LightGBM see for example Ke *et al.* (2017). LightGBM contains highly efficient implementations of various gradient boosting algorithms for classification and classification tasks.

We divide the data into three subsets using the train, validation, test paradigm as follows:

- Train data. Examples between the years 2012 and 2016, consisting of 20,143 examples. The data is used to learn a sequence of decision trees.
- Validation data from 2017 and 2018, in total 11,150 examples. This dataset is used to validate the performance of the model and avoid overfitting.
- Test data from 2019, in total 5,646 examples. The data is used to report the performance of the model on unseen data and simulate the behaviour of the model on new data.

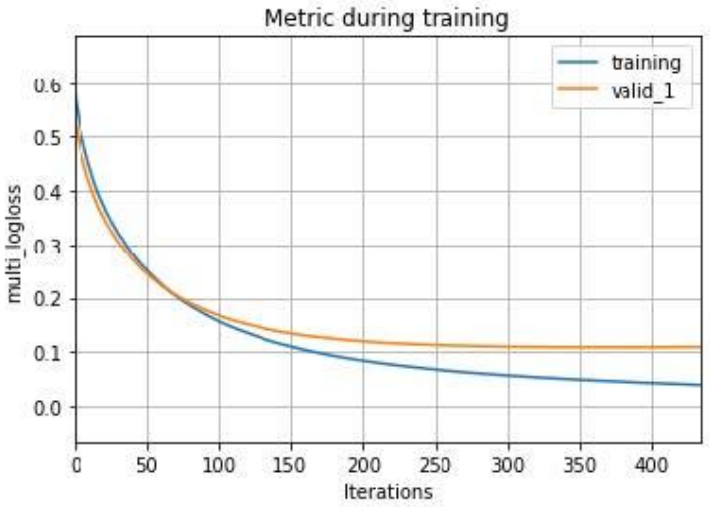

Figure 3. The train and validation error evolution. Each iteration corresponds to an additional decision tree.

The model itself outputs predicted probabilities allocating to one of the three categories ("0 claims", "1 claim", "2+ claims". An example output could be [0.2, 0.7, 0.1] - where the model is inferring this the second category in our list "1 claim").

We train the model using early stopping on the validation dataset for regularization. More precisely, if the loss function (in this case a log loss function, called "categorical cross entropy" - see below) has not improved on the validation dataset for 50 consecutive iterations, i.e., 50 new decision trees, we stop training. The plot depicting the evolution of training and validation error is shown in Figure 3. As usual, the loss improves for both datasets for some time and then the model starts to overfit the training data. The overall accuracy of the model is 0.983. The accuracy only for class 1 and 2, i.e., one claim vs. 2 or more claims, is 0.893. In Figure 4 we show the ROC curves and the corresponding AUC scores for the three classes. We see that the model is highly accurate. In Figure 5 we plot the most important features. As expected, the meteorological conditions have the highest impact on the number of insurance claims, followed by some policy features, and the type of area.

5. Concluding Remarks

Insurers may categorise the impact of climate-related claims due to a single issue: a "storm event".

Trying to understand the influences of each underlying climate-related peril can be challenging, in part due to due to complexity to model such a claim as accurately, the amount of data it would require to better understand such underlying influences, and the cost involved from a business perspective to produce such a model. If we could, however, produce a realistic approach for insurers to categorise the underlying perils and the influence of each peril, this could lead to better pricing and reserving models for an insurer, as well as more appropriate policy fees being paid by an insured. Given the complexities and potential volumes of data involved, we discussed in this paper an example of how machine learning techniques could be used towards this direction. Given the complexity of this problem, as a first step, we analysed insurer data, using "supervised learning" techniques to categorise if a "storm event" would result in a claim, and if so, how many claims. We used a machine learning algorithm called "gradient boosting" to categorise the claim as "0 claims", "1 claim" and "2+ claims" based on local insurer data from Greece. For future research, we can build on the approach and model used in various ways. Firstly, we can expand the dataset to incorporate other insurers within Greece

and potentially different countries, so as to work towards a more commercial solution. Secondly, we could incorporate external weather data at more regular intervals so that the model can infer relations between the weather and potential for a claim to incur. Thirdly, we could try out other algorithms/approaches and use "rank to learning" methods to rank the importance of each peril by claim to hence better understand the drivers for each claim and hence quantify the financial impact caused by each peril. Fourthly, we could expand the model to incorporate larger datasets and move to an "unsupervised" solution as per our original goal and move away from the open source software of LightGBM to more bespoke solution, which is coded from the ground up.

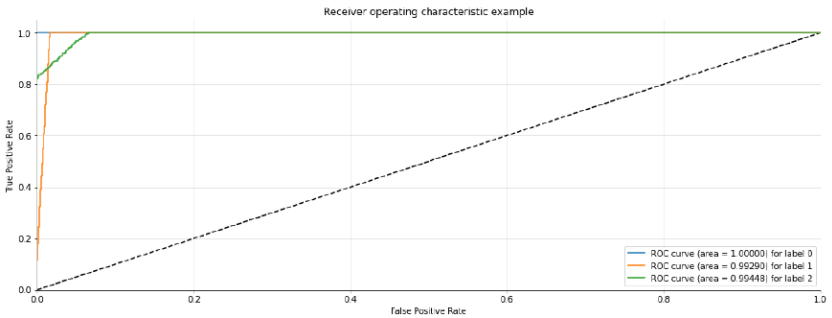

Figure 4. The AUC plots for the three classes.

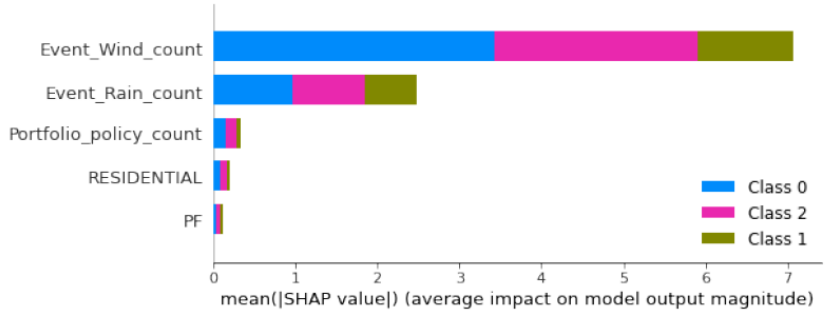

Figure 5. The most important features and their contribution to the classifying each example to a given class. (PF stands for policy fee).

6. References

Ahmed, O., & Serra, T. (2015). Economic analysis of the introduction of agricultural revenue insurance contracts in Spain using statistical copulas. Agricultural Economics, *46*(1), 69-79.

Breiman, L. (1996). Stacked regressions. Machine learning, *24*(1), 49-64.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Classification and regression trees. Wadsworth & Brooks/Cole Advanced Books & Software.

Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20-28.

Choudhury, A., Jones, J., Okine, A., & Choudhury, R. (2016). Drought-triggered index insurance using cluster analysis of rainfall affected by climate change. *Journal of Insurance Issues*, 169-186.

Clemen, R. (1989). Combining forecasts: A review and annotated bibliography. *International journal of forecasting*, *5*(4),559-583.

Daron, J. D., & Stainforth, D. A. (2014). Assessing pricing assumptions for weather index insurance in a changing climate. *Climate Risk Management*, 1, 76-91.

Dietterich, T., *et al.* (2002). Ensemble learning. The handbook of brain theory and neural networks, *2*(1), 110-125.

European Environment Agency (2022). What is CORINE land cover? https://www.eea.europa.eu/help/faq/what-is-corine-land-cover.

Freund, Y., Schapire, R., & Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society for Artificial Intelligence*, *14*(771-780), 1612.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 1189-1232.

Friedman, J. H., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning. Springer series in statistics.

Hyafil, L., & Rivest, R. (1976). Constructing optimal binary decision trees is NP-complete. *Inf. Process. Lett.*, *5*(1), 15-17.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, 3146-3154.

Knighton, J., Buchanan, B., Guzman, C., Elliott, R., White, E., & Rahm, B. (2020). Predicting flood insurance claims with hydrologic and socioeconomic demographics via machine learning: Exploring the roles of topography, minority populations, and political dissimilarity. *Journal of Environmental Management*, 272, 111051.

Leblois, A., & Quirion, P. (2013). Agricultural insurances based on meteorological indices: realizations, methods and research challenges. *Meteorological Applications*, *20*(1), 1-9.

Lyubchich, V., Newlands, N., Ghahari, A., Mahdi, T., & Gel, Y. (2019). Insurance risk assessment in the face of climate change: Integrating data science and statistics. *Wiley Interdisciplinary Reviews: Computational Statistics*, *11*(4), e1462.

Mobley, W., Sebastian, A., Blessing, R., Highfield, W., Stearns, L., & Brody, S. (2021). Quantification of continuous flood hazard using random forest classification and flood insurance claims at large spatial scales: a pilot study in Southeast Texas. *Natural Hazards and Earth System Sciences*, *21*(2), 807-822.

Odening M., & Shen, Z. (2014). Challenges of insuring weather risk in agriculture. *Agricultural Finance Review*, *74*(2), 188-199.

Perrone, M. (1993). Improving regression estimation: Averaging methods for variance reduction with extensions to general convex measure optimization [PhD thesis]. Physics Department, Brown University, Providence, RI.

Rohrbeck, C., Eastoe, E., Frigessi, A., & Tawn, J. (2018). Extreme value modelling of water-related insurance claims. *The Annals of Applied Statistics*, *12*(1), 246-282.

Tack, J., & Ubilava, D. (2015). Climate and agricultural risk: measuring the effect of enso on uscrop insurance. *Agricultural Economics*, *46*(2), 245-257.

Tesselaar, M., Wouter Botzen, W.J., & Aerts, J. (2020). Impacts of climate change and remote natural catastrophes on EU flood insurance markets: An analysis of soft and hard reinsurance markets for flood coverage. *Atmosphere*, *11*(2), 146.

Wolpert, D. (1992). Stacked generalization. *Neural networks*, *5*(2), 241-259.

Yang, Q., Shen X., Yang, F., Anagnostou, E., He, K., Mo, C., Seyyedi,H., Kettner, A. & Zhang, Q. (2022). Predicting flood property insurance claims over conus, fusing big earth observation data. *Bulletin of the American Meteorological Society*, *103*(3), E791-E809.