Predicting civil war onset with machine learning algorithms

authored by
Clément Chaffard, Quentin Hellier, Gianluca Monticone

***Abstract*—Literature on civil war onset shows that most of the common predictors of civil war onset are not robust throughout the models and the studies. Civil war data, due to its extreme unbalance isn't suited to be analysed by linear methods such as logistic regressions. In this article we use 3 different machine learning algorithms (Random Forest, Support Vector Machine and Neural Networks) to predict civil war onset and compare their accuracy using Roc curves. We show that Random Forest seems to predict civil war more accurately than its counterparts. We also analyse the important features in each model to bring to light the best predictors of civil war onset.**

I. Introduction

Extending the analysis on the different methods to predict civil war onsets form class-imbalanced data (Muchlinski, Siroky, He, and Kocher 2016), we compare Random Forest with Neural Network and Support Vector Machine. While the previous analysis demonstrated the superiority of Random Forest algorithms for predicting these rare events on the most commonly used statistical models, namely logistic regressions, even in their rare event and regularised forms, our study compares the performance of Random Forest with two other machine-learning algorithms:

- Neural Networks, which is an algorithm composed by a set of decision nodes, modelled after the human brain and designed to recognise patterns;
- Support Vector Machine (SVM), which is a supervised learning model that generates a non-probabilistic binary linear classifier based on the spatial division in two categories made on the training set of data.

Predicting civil war onsets is critical in political science as these events are incredibly dangerous and destructive. Accurate predictions on their onset could be decisive to anticipate these conflicts and find ways to prevent or contain them. In consequence, our analysis focuses on finding the most efficient available machine learning algorithm to perform the predictions. Furthermore, the analysis compares the importance that the different algorithms give to the different features and highlights if common features share a major role between all three algorithms.

II. Dataset description

Our dataset is a matrix that contains the Civil War Data (CWD) and 88 different features such as the agricultural raw materials exports as percentage of merchandise exports, the average SIP score of neighbours... It was taken from Hegre and Sambanis 2006 and used in the article used as a basis for our work (Muchlinski, Siroky, He, and Kocher 2016). The CWD are extremely unbalanced as the ratio of conflict years to peace years in the data is roughly 1:100 as shown in Fig.1.

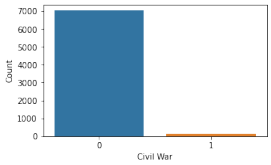


Fig. 1: Civil War Data

To further highlight the unbalanced aspect of our data, we have calculated some basic descriptive statistics shown in the table below.

Basic descriptive statistics of the CWD	
count	7140.000000
mean	0.016246
std	0.126431
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

III. Data collection

A. Dealing with unbalanced dataset over- and undersampling

Applying inappropriate evaluation metrics for model generated using imbalanced data can be dangerous. If accuracy is used to measure the goodness of a model, a model which classifies all testing samples into "0" will have an excellent accuracy (99.8%), but obviously, this model won't provide any valuable information for us. We will measure AUC: relation between true-positive rate and false positive rate.

Random oversampling involves randomly duplicating examples from the minority class and adding them to the training

dataset. In some cases, seeking a balanced distribution for a severely imbalanced dataset can cause affected algorithms to overfit the minority class

Random undersampling involves randomly selecting examples from the majority class to delete from the training dataset. This approach may be more suitable for datasets where there is a class imbalance although enough examples in the minority class. A limitation of undersampling is that examples from the majority class are deleted that may be useful, important, or perhaps critical to fitting a robust decision boundary.

IV. Methods and description of the algorithms

A. Random Forest

Random forests constructs many individual decision trees during training. Predictions from all trees are pooled to make the final prediction.

1) Implementation in Scikit-learn: For each decision tree, Scikit-learn calculates a nodes importance using Gini Importance, assuming only two child nodes (binary tree):

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

with ni_j the importance of node j, w_j the weighted number of samples reaching node j, C_j the impurity value of node j, $left(j)$ the child node from left split on node j, $right(j)$ the child node from right split on node j

The importance for each feature on a decision tree is then calculated as:

$$fi_i = \frac{\sum_{j:node j splits on feature i} ni_j}{\sum_{k \in all nodes} ni_k}$$

with fi_i the importance of feature i and ni_j the importance of node j. These can then be normalized to a value between 0 and 1 by dividing by the sum of all feature importance values.

The final feature importance, at the Random Forest level, is it's average over all the trees. The sum of the feature's importance value on each trees is calculated and divided by the total number of trees:

$$RFfi_i = \frac{\sum_{j \in all trees} normfi_{ij}}{T}$$

- $RFfi_i$ = the importance of feature i calculated from all trees in the Random Forest model
- $normfi_{ij}$ = the normalized feature importance for i in tree j
- T = total number of trees

B. Support Vector Machine (SVM)

The SVM model is a model that will try to find a hyperplane in an N-dimensional space (with N the number of features) that distinctly classifies the data points. Some data points will be identified during the training phase as points that matter for defining the boundary between two classes, they are the points that lie at the border between the classes. These points are called support vectors. The distance between the hyperplane and each support vector is calculated and the algorithm will try to maximise the distance in order to have the clearest possible boundary. The distance between data points is measured by the Gaussian kernel: $k_{rbf}(x_1, x_2) = e^{\gamma \|x_1 - x_2\|^2}$ where x_1 and x_2 are data points and $\|x_1 - x_2\|$ the Euclidean distance.

We will use cross-validation to find the best hyperplane according to the auc score of the roc curve. We will test different C and γ. The gamma parameter is the one shown in the formula above, which controls the width of the Gaussian kernel. It determines the scale of what it means for points to be close together. The C parameter is a regularisation parameter, similar to that used in the linear models. It limits the importance of each point.

C. Neural Networks

Neural Networks are organised in layers made up of interconnected nodes which contain an activation function that computes the output of the network.

Single neuron:

Step 1: For each input, multiply the input value x_i with weights w_i and sum all the multiplied values. Weights represent the strength of the connection between neurons and decides how much influence the given input will have on the neuron's output.

$$\sum = x.w$$

Step 2: Add bias b to the summation of multiplied values and introduce non-linearity into the output of the neurons

$$z = x.w + b \quad \text{and} \quad \hat{y} = \sigma(z) = \frac{1}{1 + \exp(-z)}$$

Backpropagation: computing the gradient of the loss function with respect to the weights Step 1: Loss function is calculated for the entire training dataset and their average is called the Cost function C using mean squared error:

$$C = MSE = \frac{1}{n} \sum (y_i - \hat{y}_i)^2$$

Step 2: In order to find the best weights and bias for our Perceptron, we need to know how the cost function changes in relation to weights and bias. Using partial derivation, we obtain:

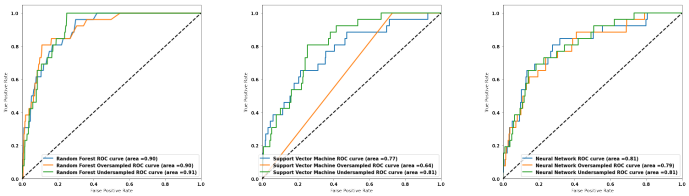
$$\frac{\partial C}{\partial w_i} = \frac{2}{n} \times \sum (y - \hat{y}) \times \sigma(z) \times (1 - \sigma(z)) \times x_i$$

$$\frac{\partial C}{\partial b} = \frac{2}{n} \times \sum (y - \hat{y}) \times \sigma(z) \times (1 - \sigma(z))$$

Optimization: The weights and bias are updated as follows and the backpropagation and gradient descent is repeated until convergence : $w_i = w_i - (\alpha \times \frac{\partial C}{\partial w_i})$ and $b = b - (\alpha \times \frac{\partial C}{\partial b})$

V. Results and findings

A. Roc Curves



(a) (b) (c)

Fig. 2: Roc curves with different training sets for the 3 algorithms (a) Roc curves for Random Forest using over- and undersampled data (b) Roc curves for SVM using over- and undersampled data (c) Roc curves for NN using over- and undersampled data.

We can observe that oversampling slightly improved Random Forest algorithm from an AUC value of 0.90 (without oversampling) to 0.92. The presence of duplicated minority class 1(war) helped for the model predictions. On the other hand, undersampling improved both Neural Network and SVM algorithms respectively from 0.81 to 0.83 and from 0.77 to 0.82. Neural Network and SVM were able to make more accurate prediction reducing considerably the size of the dataset from over 7000 samples to 500 samples.

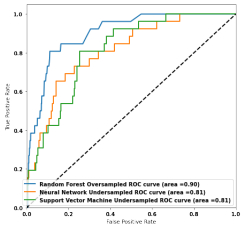


Fig. 3: Roc curves for the 3 algorithms with the best training set

The analysis presented shows that Random Forest offer best prediction model for civil war onset. Its ability of compensating a high degree of variance between each tree using multiple decision trees to predict for a low biased output variable showed being the most fit to describe the non-linear and unbalanced dataset of civil war onset.

B. Feature importance

In this section, we evaluate the importance of the features in each model to find the best predictors of civil war onset. To evaluate the feature importance in SVM and Neural Networks, we computed feature importance for any black-box estimator by measuring how score decreases when a feature is not available. Score can represent accuracy, TPR, R^2 etc. To avoid retraining the model for each feature, we replace the feature with a random noise in the test part and evaluate the score.

The feature importance for Random Forest is calculated using the Gini Score mentioned in Section IV.

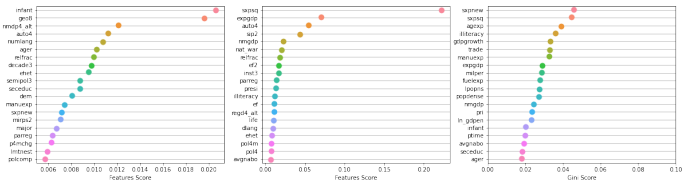


Fig. 4: Features importance in the 3 models. From left to right : Neural Network, SVM and Random Forest

First of all, we can see in Fig.4 that some features do have a greater importance in civil war onset prediction. Surprisingly, these features differ depending on the algorithm used. In order to see more clearly the variables that allow the best description, we looked at the different variables found by each of the algorithms and noted the ones that were in common. No feature was identified as one of the most important by the 3 algorithms. Among the features that were estimated important by 2 of the algorithms were Political instability, Religious fractionalisation, Infant mortality, Annual change in GDP, Illiteracy and Ethnic heterogeneity index. Those features are all linked to the political, social and economical stability of the country. It doesn't seem surprising that the political regime would influence the possibility of a civil war onset, and it is well-known and established that one of the best predictor of civil war onset is national poverty reflected by the imports/exports and the annual change in GDP. It is interesting but not counter intuitive that religious fractionalisation and ethnic heterogeneity could influence civil war onset.

VI. Conclusions

In domains where logistic regression fails to predict the different outcomes, because linearity and collinearity between parameters are not respected and with an extremely unbalanced dataset, such as political science, we compared random forest, neural network and SVM algorithms expecting more accurate prediction. Random Forest proved to have the best predictive power of the three models. Furthermore, Random Forest, Neural, Network and Support Vector Machine allowed us to enhance our understanding of rare and complex events like civil war onset, underlining well known political ,social and economical instability in countries plagued by civil wars.

References

[1] D. Muchlinski, D. Siroky, J. He, and M. Kocher, "Comparing random forest with logistic regression for predicting class-imbalanced civil war onset data," *Political Analysis*, vol. 24, no. 1, pp. 87–103, 2016. DOI: 10.1093/pan/mpv024.

[2] H. Hegre and N. Sambanis, "Sensitivity analysis of empirical results on civil war onset," *Journal of Conflict Resolution*, vol. 50, no. 4, pp. 508–535, Aug. 2006. DOI: 10.1177/0022002706289303.

[3] [Online]. Available: https://medium.com/analytics-steps/understanding-the-gini-index-and-information-gain-in-decision-trees-ab4720518ba8.

[4] [Online]. Available: https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3.

[5] [Online]. Available: https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf.

[6] [Online]. Available: https://www.kdnuggets.com/2020/01/5-most-useful-techniques-handle-imbalanced-datasets.html#:~:text=1.,Random%20Undersampling%20and%20Oversampling&text=A%20widely%20adopted%20and%20perhaps,class%20(over-sampling).

[7] S. Guido, *Introduction to Machine Learning with Python*. O'Reilly UK Ltd., Jul. 31, 2016, 400 pp., ISBN: 1449369413. [Online]. Available: https://www.ebook.de/de/product/23308778/sarah_guido_introduction_to_machine_learning_with_python.html.