Producing Early Warnings through Online News With R

Palabras clave: shiny, machine learning, vetiver, prediction, MLOps

Introduction

The availability of vast amounts of online news data has opened new avenues for early warning systems, which play a crucial role in identifying and mitigating potential risks and threats. This work explores the use of the R programming language to extract meaningful insights from online news sources and harness them for early warning purposes. To capture soaring food prices and help designing mitigating measures, we developed two complementary products (shiny apps): **Topics Explore**r, which monitors the sentiment and popularity of pre-defined topics that impact Sustainable Development Goals and **Data Lab Trends**, stepping away from pre-defined topics, the tool analyses the popularity of any search query over the articles gathered from media outlets across various countries and languages. Both products were designed considering that the timeliness of data and the capacity to quickly and automatically draw insights from data for policymaking is crucial in emergency times.

Data

The news article database utilized for this analysis is a vast and dynamic resource, consisting of more than 25 million tweets and associated articles from 650 news accounts. These articles, of varying lengths, are conscientiously scraped, processed, and stored on a daily basis. Data collection began on the 1st of January 2020 and has been ongoing ever since, spanning a total of 192 countries. On average, three accounts are considered for each country, providing articles in multiple languages, including Arabic, English, French, Italian, Portuguese, Russian, and Spanish. The database is continuously updated with approximately 22,000 new articles daily, and the Apache Solr database is used to manage all the unstructured data effectively. The **solrium** R package developed by rOpenSci provides several useful functions to query and parse data from the Solr database.

Pipeline for Topics Classification

This section provides a concise overview of the methodology used to classify the articles into topics. The literature presents various machine learning techniques for text classification, such as the Latent Dirichlet Allocation (LDA) family that discovers multiple topics in a large corpus. However, this approach is unsuitable for our case since we are not attempting to discover topics in the corpus overall. Instead, we know which topics we seek and aim to identify them in the texts. The multi-label classification method, which assigns multiple nonexclusive labels to each instance, is impractical for our problem as new topics may emerge over time, requiring model retraining and document reclassification. In contrast, binary classification, the simplest family of classifiers, is the most fitting for our problem. Each topic is treated independently, so new topics are trained separately without impacting the other models or previous classifications.

In classification methods, a training dataset is a set of labelled data used to train a machine learning algorithm to recognise patterns and make predictions or classifications based on the given inputs. Labelled data is hard to build because, in most cases, human work is needed to classify each instance manually, requiring a lot of time if you have several topics and different languages. To overcome this

issue, we propose using artificial labelled data generation, which can be beneficial when acquiring labelled data is expensive, time-consuming, or limited. However, it is important to ensure that the artificially labelled data accurately represents the real-world data and does not introduce bias or other issues that could affect the performance of the machine learning algorithm(Hastie, Tibshirani, and Friedman 2009; Bishop and Nasrabadi 2006; Efron and Hastie 2021). To build the training data, we used the package **solrium** combined with the others from the **tidyverse** family such as **dplyr** and **tidyr**.

In typical machine learning training, we follow a series of steps to ensure that we build a robust and accurate model. First, we preprocess the data and perform feature engineering to prepare it for the modelling stage. Then, we train three classifiers: XGBoost, LightGBM, and LASSO, to explore different model architectures and assess their performance. Each classifier's performance is then measured using the F-measure (see Buckland and Gey 1994; Powers 2011), which is a weighted harmonic mean of precision and recall. Based on the F-measure, we select the best model among the three. Finally, to set the optimal decision threshold, we use Youden's J statistic (see Youden 1950; also Powers 2011, 37–63), which identifies the threshold that maximises the difference between the true positive rate and the false positive rate. At this pipeline stage, the packages we used are **tidymodels**, **tidyrecipes, and themis.** The last one generates synthetic data samples by enlarging the feature space of minority and majority classes.

The process of setting tuning parameters for many machine learning models can be challenging as they cannot be directly estimated from the data. Resampling methods such as cross-validation or bootstrap are commonly used to evaluate a set of candidate values and select the best based on a predefined criterion. However, this process can be time-consuming. A more efficient approach involves adaptively resampling candidate values to eliminate sub-optimal settings. The approach involves computing performance metrics such as accuracy or RMSE for a pre-defined set of tuning parameters for a model or recipe across one or more resamples of the data. Tuning parameter combinations that are unlikely to yield the best results are then eliminated using a repeated measure ANOVA model after evaluating an initial number of resamples (see Kuhn 2014). At this stage of the pipeline, the packages we used are **tidymodels** library which provides functions for resampling and finetune package which enables the ANOVA strategy.

Finally, we use the **vetiver** and **pin** packages to version, deploy, and monitor the trained models. These packages allow us to save metadata associated with the model, making the readability and reproducibility of the whole process easy.

The total of models trained and monitored is 13 (topics) x 7 (languages), resulting in 77 models in total. Those models are applied daily to the new articles collected, predicting the topics covered by the news.

Products

Topics Explorer monitors topics covered by the global press that are related to — and have an impact on — the Sustainable Development Goals (SDGs), such as food security, climate change, and food losses and waste. It provides indicators of sentiment and popularity of the topics in many countries of the world, allowing automated analysis: (https://www.fao.org/datalab/early-warnings/topics-explorer/en)

Data Lab Trends analyses the popularity of search queries related to food security and nutrition and agrifood system transformation, over the articles gathered from media outlets across various countries and languages. With this tool, the user can search for any topic to visualize its trend over time and its geographic distribution. (https://www.fao.org/datalab/early-warnings/data-lab-trends/en)

References

Bishop, Christopher M, and Nasser M Nasrabadi. 2006. *Pattern Recognition and Machine Learning*. Vol. 4. 4. Springer.

Buckland, Michael, and Fredric Gey. 1994. "The Relationship Between Recall and Precision." *Journal of the American Society for Information Science* 45 (1): 12–19.

Efron, Bradley, and Trevor Hastie. 2021. *Computer Age Statistical Inference, Student Edition: Algorithms, Evidence, and Data Science*. Vol. 6. Cambridge University Press.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. "An Introduction to Statistical Learning."

Kuhn, Max. 2014. "Futility Analysis in the Cross-Validation of Machine Learning Models." https://arxiv.org/abs/1405.6974.

Powers, David MW. 2011. "Evaluation: From Precision, Recall and f-Measure to ROC, Informedness, Markedness and Correlation." *Journal of Machine Learning Technologies*, 37–63.

Youden, William J. 1950. "Index for Rating Diagnostic Tests." *Cancer* 3 (1): 32–35.