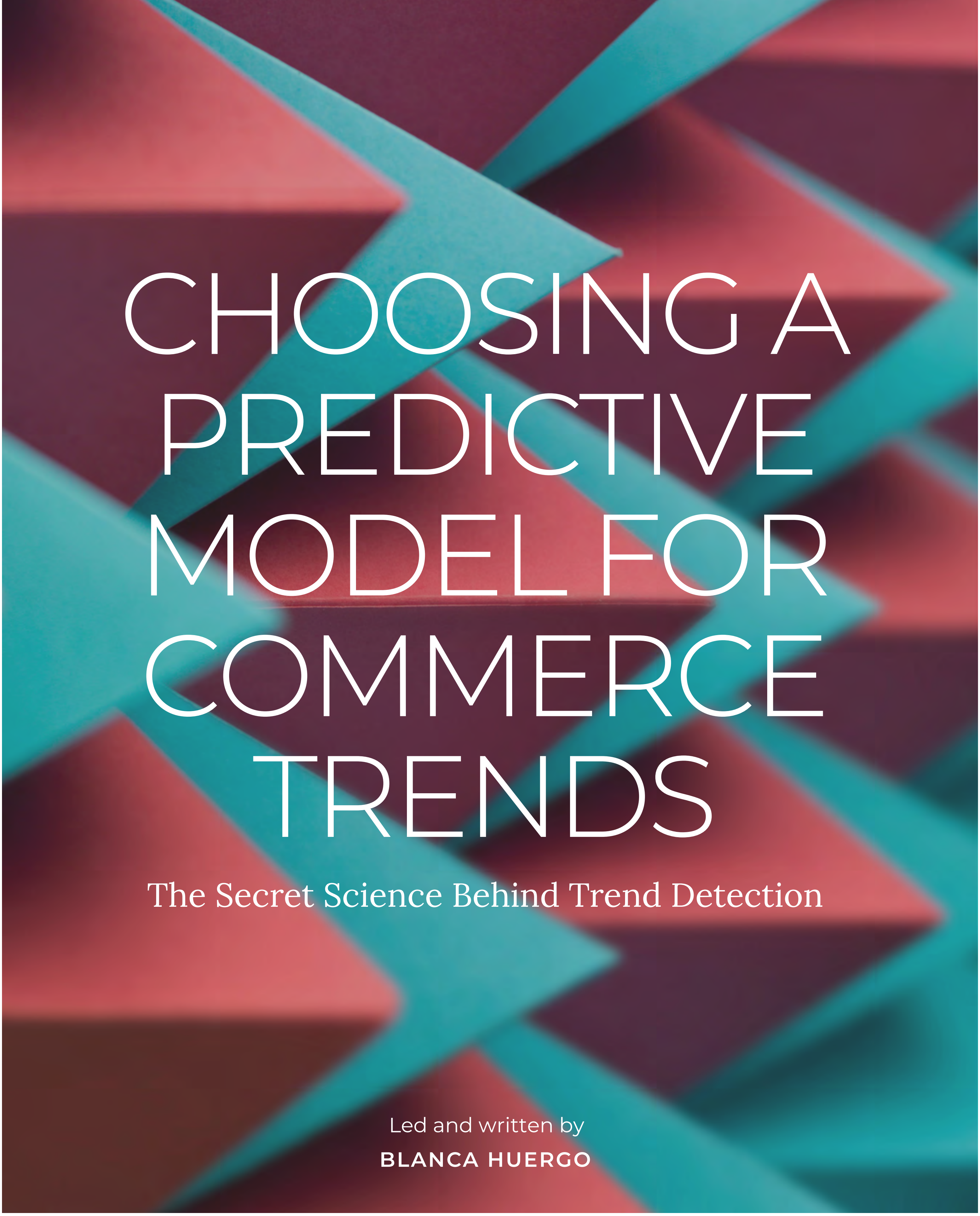


CHOOSING A PREDICTIVE MODEL FOR COMMERCE TRENDS

The Secret Science Behind Trend Detection

Led and written by

BLANCA HUERGO

EMPATHY.CO

CONTENTS

CONTENTS

01 INTRODUCING THE PREDICTIVE MODEL

When analysts peruse e-commerce data, they usually handle customers as a homogenous crowd in which individuals whose tastes differ from the majority's are just classed as outliers.

We recognize that each separate consumer desire among the large mass of queries may not be monetizable. But narrowing one's perspective as an analyst to the small selection of popular queries means giving up on a substantial number of not-so-popular queries and, consequently, leaving a significant number of customer journeys (possibly greater than the number actually taken into account!) with an unhappy ending.

Every customer journey should have a happy ending. With this motivation as a starting point, we wish to go beyond search KPIs and dive into different mindsets. For that task, Empathy.co has developed the predictive model.

Three key components define predictive model:

KPI WARNINGS

KPI warnings make the client aware that one of the main metrics used by Empathy to measure success in their e-commerce site has a value which is lower than expected. The analyst who sees this message may then gain a greater level of insight, getting to know the reasons behind this change in consumer behaviour.

TREND PREDICTION

Trend prediction predicts interests that are rapidly gaining traction. This allows Empathy's clients to be ready for changes in consumer tastes before their competitors.

PERFORMANCE ALERT

Performance alert quickly exposes the large-volume queries (shirts, shorts...) that are performing worse than the store's average so that their performance can be treated.

This paper will be focus on the first type of alert: **KPI warnings**, and in particular time signals which measure a countable variable, as opposed to percentage signals.

We will use the total queries signal as an example and we'll explain how predictive models and their output are used as a base for the KPI warnings. Total queries are often used as a measure of website traffic and, therefore, of consumer interest in the brand.

02

CHOOSING A PREDICTIVE MODEL

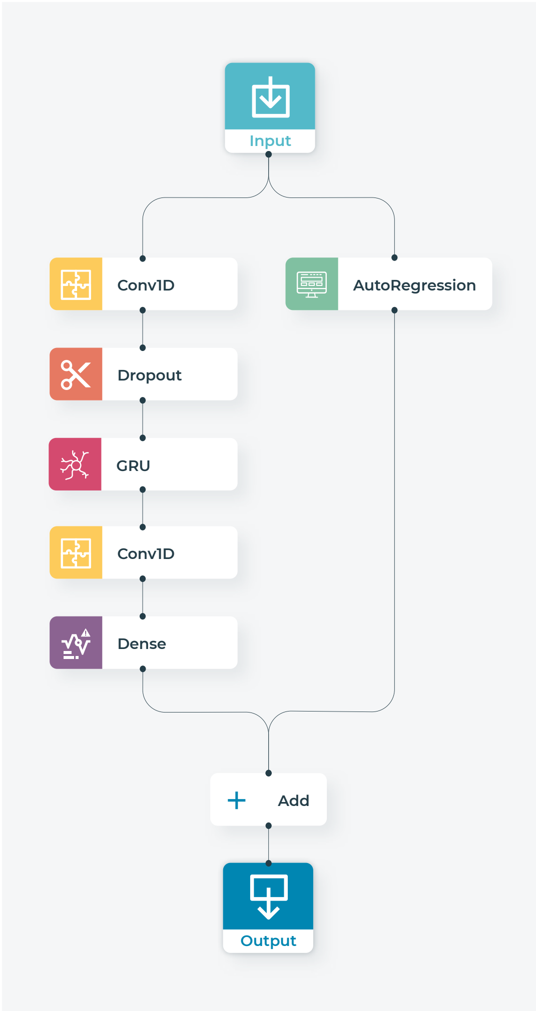
When choosing a predictive model, it's important for the model to take into account the growing trend of the signal, as well as its seasonal component and special events (Black Friday or New Year's Eve).

However, most fluctuations of the time signal are not predictable in advance, mainly because the data which are used in the univariate prediction are just the value of that same time signal for a certain window of time steps before the predicted one.

Most of the abrupt changes are either random or controlled by external variables unknown to the system making the prediction. This means that the objective is not for the model to predict these fluctuations which, given the input data and their shortcomings, can mostly be treated as random error. Instead, the intent is for the model to extract short-term and long-term relationships between the time steps, modelling the signal's general behaviour. The flag would then be raised in the case of a significant fall in the recorded signal, compared with the prediction.

Our initial approach tested how the out-of-the-box Facebook Prophet library worked on this data. This method had the advantage of taking the three desired components (trend, seasonality and special events) into account. But it worked very poorly with clients with only one or two years' worth of data (predicting a mere horizontal line, not even detecting the growth trend), and that is a sector for which the predictive model must also work. Therefore, even though it had promising results for a small number of older clients, it did not scale well to many others, and so we discarded it.

After working with Prophet and some other classic ML methods, creating a custom model seemed appropriate for this project, as none of the algorithms tested were able to capture the dependencies to the exact extent desired. Inspired by recent advances in time series analysis research, such as LSTNet, the next step was to design a two-branch combined neural network.



The right branch of the diagram is an autoregressive section. This finds the growth trend in the data, which is the weak point of recurrent neural networks, which detect periodic relationships with ease but have more trouble plotting the general trend of the signal.

The left branch of the diagram consists of two main sections: a convolutional layer and a gated recurrent unit layer. Between them, dropout layers are included to avoid overfitting. A final dense layer converts the result to a single scalar.

The convolutional section finds short-term dependencies in the data, such as how day X's value is affected by day (X-3)'s value. On the other hand, the recurrent section extracts long-term dependencies, such as how the total queries signal has two peaks, one in winter and one in summer.

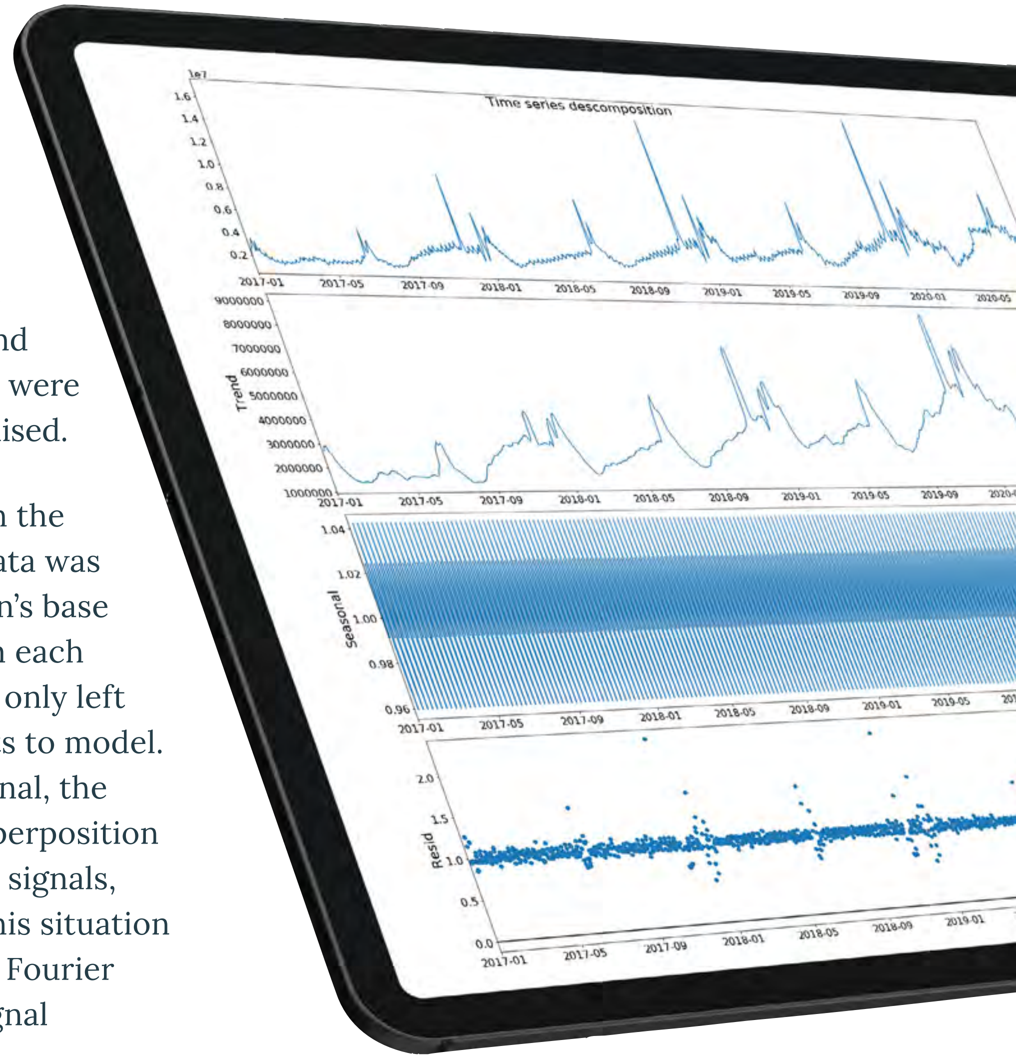
The scalar obtained from the left branch is then added to that obtained by the right branch, giving the final output prediction. This model had a clear edge over Facebook Prophet for customers with scarce amounts of data, but still faced a problem, which it shared with Prophet: the model was trying too hard to predict the unprecedented fluctuations, losing sight of the bigger picture. Prophet did not allow for much customisation. But in the neural model, it managed to reduce this effect by changing the number of epochs (gradient descent iterations) and setting the gradient descent objective function to mean absolute error, as opposed to mean squared error. Why?

MAE penalised predictions closer to the trend, and hence further from the fluctuations, by a smaller degree than MSE did. However, the model's performance was still too constrained by the objective functions and the random fluctuations; it gave too little weight to the recurrent component, overlooking crucial long-term relationships to give more importance to recent data, rendering the model short-sighted.

For the final model, the signal was processed by decomposing it into its different components. The program first extracted the growing trend from the signal, giving the model three options: linear, exponential and logarithmic. The three curves were fitted on the signal and optimised.

Of these curves, the one with the lowest loss on the training data was then chosen as the prediction's base and its value subtracted from each point of the time signal. This only left seasonality and special events to model. Looking at the detrended signal, the pattern seemed to be the superposition of several individual periodic signals, such as a biyearly pattern. This situation is optimal to carry out a Fast Fourier Transform, projecting the signal into the frequency domain and then extrapolating this knowledge into the future. Even though the accuracy of this approach increases with more years' worth of data, the function still modelled the seasonality component with surprising precision for clients with one or two years' history.

*Decomposition of the time series into its different components and the multiplicative model that relates them y(t) = Trend * Seasonality * Noise (resid).*



03

PREDICTING SPECIAL EVENTS

Finally, in order to predict the increased traffic during special events, we fed the model special events with their past and future dates. By calculating the growth trend in the traffic of these events in the past in a manner similar to how the initial trend of the signal was found, the future traffic on these special dates could be found with high accuracy.

In the case of websites with only one or two years' history, special events predictions can be extracted from the general growth trend of the signal by plotting this trend over the past event values, extrapolating into the future.

To extract the final result, the general trend is added to the Fourier forecast, except for the case of special events, where the prediction for that time step is substituted for that different value.

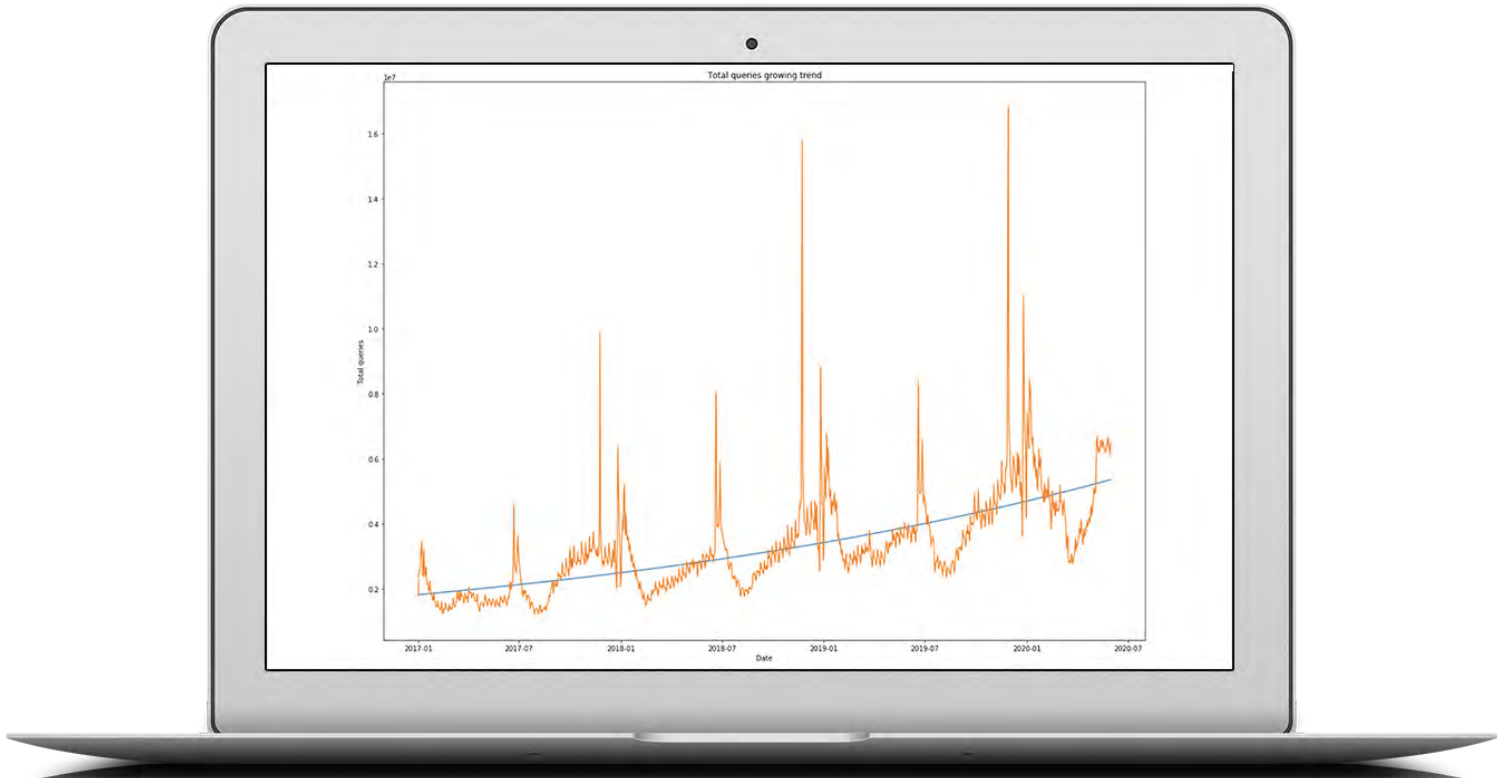
As a whole, the final model was the one that best adjusted to the initial expectations and objectives. Without focusing excessively on random peaks and troughs, it extracted the dependencies between the different time steps, allowing us to choose a threshold of relevance and hence selecting those dependencies with stronger empirical evidence and discarding those found from the random fluctuations.

Time signal projected into the frequency domain

Moreover, this model is the one with the highest degree of interpretability of the three, with the option of examining the frequency domain graphically, as well as the base trend.

Client X's total queries data will be used to showcase results in this example application. This customer is in the fashion industry.

In order to make the 2020 prediction, data was taken up to 31 January 2019 and we trained the model to forecast a signal for 2020. As can be seen in the 3rd graph, January and February, including Winter sales, perfectly adjust to the forecast, whereas the quarantine period data are significantly lower than expected, followed by a sharp rebound. This black swan was impossible to see in advance and is the type of event for which the total queries alert should have quickly popped up, as it did in my experiment, urging Empathy's client to take action, consequently encouraging



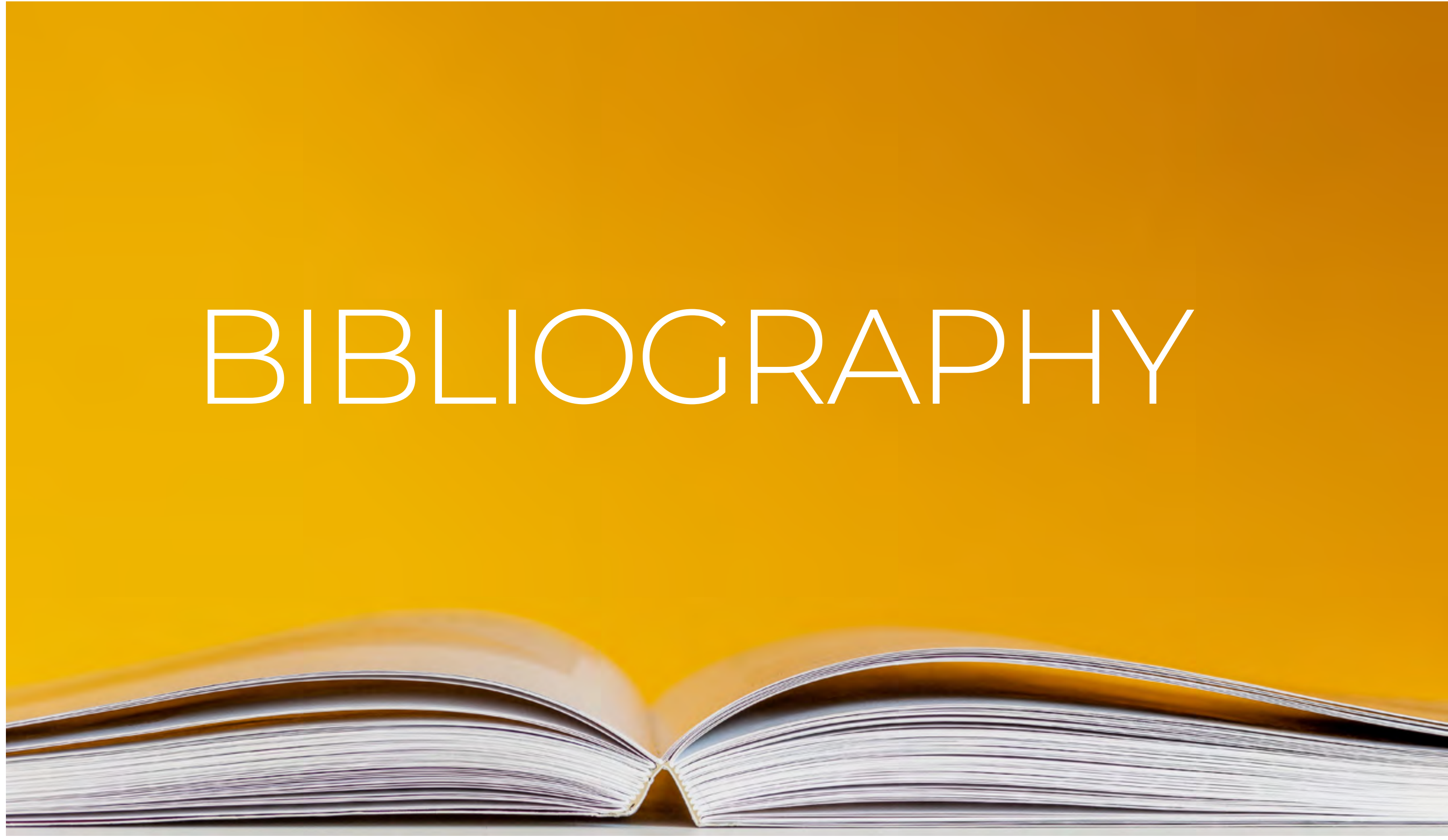
The growth trend extracted from the signal.

consumers to go back to browse their website. The rest of the months show the trend of past years, with the winter and summer peaks and including special events such as Black Friday.

Hopefully, with the alert system in place, the client would have reacted earlier to the fall in total queries and would have reverted this effect with the Trend prediction and Performance alerts' advice.

Due to its performance, this model will therefore be integrated into the predictive model's KPI warnings, which will appear in three different intensities (green, yellow and red) depending on the severity of the situation, and which will pop up when the recorded signal for one day is worse than that expected by a certain proportion. The training time is below 2 seconds for the data-richest consumers and can make spot-on forecasts with several months' anticipation, even for those with a small history, enhancing its applicability.
The next post in this series will explore KPI warnings for percentages in detail.

Both components above added together, with the inclusion of special events.



BIBLIOGRAPHY

- The Signal and the Noise, Nate Silver
- IBM- Advanced Machine Learning and Signal Processing: **https://www.coursera.org/learn/advanced-machine-learning-signal-processing**
- A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting: **https://www.sciencedirect.com/science/article/pii/S0169207019301153**
- Facebook Prophet Documentation: **https://facebook.github.io/prophet/**
- N-BEATS: Neural basis expansion analysis for interpretable time series forecasting: **https://arxiv.org/abs/1905.10437**
- Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks: **https://arxiv.org/abs/1703.07015**

EMPATHY.CO

LONDON | NEW YORK | ASTURIAS | GALICIA

info@empathy.co

www.empathy.co