Hi!ckathon 2023 : AI & Supply Chain

Matthieu Bricaire, Victor Hoffman, Changhao Li,
Marius Ortega, Hussein Rammal, Thomas Salomon
Team 5

December 1th-3rd 2023

Abstract

This article is written in the context of the Hi!ckathon Data Challenge 2023 hosted by Hi!Paris, organization born from the collaboration of HEC and Institut Polytechnique de Paris. In the following content, we present our journey through the proposed challenge, including our general scientific approach, the choices we made regarding the feature engineering procedure applied to the data, and the model we developed, called Next Leaf.

Project Background and Description

Considering the highly legitimate concerns about the environment's current state and knowing the potential impact of AI on environment-related KPIs, one of the main point of our project is to propose an impactful and sustainable solution. In addition, we believe that sustainability and performance are key metrics, that should be monitored and optimized conjointly. This is why we worked to propose an innovative and supply-chain-based solution, that we present in this paper.

Our solution, Next Leaf, is a prediction algorithm based on Microsoft's LightGBM model [1]. It was trained to predict the sales value of the last month of a given quarter for different products.

Model Explanation

Preprocessing

In order to prepare the dataset for the computation we followed multiple steps :

Categorical Variables : Our main training dataset contains many categorical variables with a major part of them being proxy variables. Consequently, we must encode these variables. Depending on the amount of unique classes per column u, we apply different encoding methods based on a threshold value n.

$$\begin{cases} u < n \Rightarrow \text{One-hot encode} \\ u >= n \Rightarrow \text{Target encode} \end{cases} \tag{1}$$

Sales Value Imputation: We have 4 columns associated with sales values (Month 1 to 4), with Month 4 being our target. We encountered two main issues with sales values. Firstly, the columns were `'str'`, thus we had to parse them. Secondly, Month 1 sales contained missing values. To solve this second issue, we inferred their value with a linear regression. To do so, we first had to split the dataset into two subsets, corresponding to the known and unknown (`NaN`) values for the Month 1 column. Then, we split the "known-values" subset into train and validation sets on which we fitted the linear regression, before inferring the missing values on the remaining "unknown-values" subset.

Geographical Data :

The initial dataset contains six columns related to geographical information linked to the product : (Region, Country, Site, Operation, Zone, Cluster). To reduce the number of features of the dataset, we chose to select only the most granular geographical information, which is contained in the Site variable. To transform it into a quantitative variable, we mapped it to its geographical coordinates in the form of a (longitude, latitude) tuple.

Data Aggregation :

In order to give the Next Leaf algorithm a more holistic vision of the data, we chose to aggregate additional datasets to the initial one. In particular we added the following :

- **World bank economic data** : the economic health of a country/region is tightly linked to its purchasing power, hence influencing sales for given products. Subsequently, we chose to include the GDP, Final Consumption Expenditure Growth, Imports, Exports, Manufacturing, Industry and Services variables (yearly) in our model. To do so, we converted them into quarter amounts, imputed the `NaN` values (the mean of the corresponding column was used), and merged the result to the main training dataset using the Country column as a key.
- **World bank inflation data**: the inflation related indexes seem relevant to help predict sales, as a high inflation may be synonym of a decrease in the consumption, and thus lower sales. Hence, we chose to include the Energy Price Index and Headline Consumer Price Index variables in our model. Since the previously mentioned indexes are released on a monthly basis, we had to retrieve only the data that belonged to the September 2020 - July 2023 temporal window. Then, we averaged the remaining data over a 4-month period, before merging the result to the main dataset on the Date column.
- **GSCPI data** : the Global Supply Chain Pressure Index is another extremely powerful KPI to get information on the market state. Since it is a monthly index, we applied a preprocessing similar to what we did on the previously described dataset.
- **LPI extend** : Finally, we included population and growth rates, as well as other LPI data related to the quality of infrastructure and logistics resources in the countries. We applied PCA dimension reduction and kept the two principal components to get a summarized representation of this demographic data.

Algorithm

To design our model, our starting point was the fact that we mainly had to deal with tabular data. Indeed, even though the outcome to predict (sales of Month 4) can be thought of as the next time step of a time series (through the sales value from Months 1, 2, and 3), the vast majority of the available features consist in tabular variables that have no temporal structure. Thus, we decided to focus on tree-based regression methods as well as penalized linear regressions, which we thought would be well suited to tackle our problem considering explanability, sustainability and efficiency factors. Namely, we tested several methods, including Boosting models [3] (XGBoost, Light GBM) and ElasticNet, which can be thought of as a linear regression that combines Lasso (L1) and Ridge (L2) penalizations.

While being computationally light and highly efficient, Light GBM is also an open source Microsoft product [1], meaning it is a well maintained, stable and trustworthy algorithm [2]. During our experiments, it proved to perform consistently well for our regression task, so we chose it as the basis for our final model.

To be more precise, Light GBM grows trees vertically (leaf-wise) while other algorithms grow trees horizontally (level-wise). It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithms can reduce the loss function more than level-wise algorithms. In addition, this architecture allows to obtain smaller trees leading to more frugal, hence more environment-friendly models.

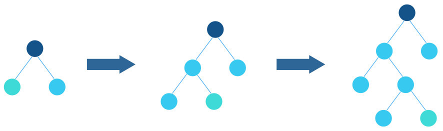

Figure 1: Light GBM leaf-wise growth

Carbon Footprint

To take into account the carbon emission in the design of our product, the utilized process includes two main features described in below.

Efficient design :

As explained previously, we use a smart and light algorithm in order to limit environmental impact of our product. In addition, we could consider the use of pipeline to optimize pre, in and post-processing of the data to further improve its efficiency.

Monitor and Ensure quality :

Beyond designing efficient algorithms, its important to validate that production usage is consistent with mathematical theory. To do so, many tools allow to track environmental cost of models. We chose the `CodeCarbon` library to ensure a proper control of carbon emissions.

Conclusion

Considering both performance and environmental aspects, we obtain of final score of 46.16 calculated with the below formula :

$$RMSE_{hfct} = RMSE\left(y_{true}, [0] * len(y_{true})\right)$$

From there, we have many improvement perspectives for our model. For instance, we could modify our target metric to make it a new feature penalized by a production factor in order to limit goods over-production. Another aspect would be to scrap additional variable such as Credit Default Swap per country which is a particularly inflation predictor.

References

[1] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 3149–3157, 2017.

[2] Qi Meng, Guolin Ke, Taifeng Wang, Wei Chen, Qiwei Ye, Zhi-Ming Ma, and Tie-Yan Liu. A communication-efficient parallel algorithm for decision tree. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, pages 1279–1287, 2016.

[3] Yu Shi, Guolin Ke, Zhuoming Chen, Shuxin Zheng, and Tie-Yan Liu. Quantized training of gradient boosting decision trees. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, pages 18822–18833, 2022.