CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

A Machine-Learning-Based System for Predicting Service Level Failures in Supply Chains

Gabrielle Gauthier Melançon
Philippe Grangier
Eric Prescott-Gagnon
Emmanuel Sabourin
Louis-Martin Rousseau

July 2019

CIRRELT-2019-28

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

UNIVERSITÉ LAVAL McGill Université Concordia University ÉTS UQÀM Université du Québec à Montréal HEC MONTRÉAL POLYTECHNIQUE MONTRÉAL Université de Montréal

A Machine-Learning-Based System for Predicting Service Level Failures in Supply Chains

Gabrielle Gauthier Melançon[1,2,ł,*], Philippe Grangier[2,ł], Eric Prescott-Gagnon[2,ł], Emmanuel Sabourin[3], Louis-Martin Rousseau[1]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Mathematics and Industrial Engineering, Polytechnique Montréal, P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C 3A7

[2] Élément AI, 6650 Rue Saint-Urbain #500, Montréal, Canada H2S 3G9

[3] JDA Software, 15059 N Scottsdale Rd, suite 400, Scottsdale, AZ 85254, US

Abstract. Despite advanced supply chain planning and execution systems, manufacturers and distributors tend to observe service levels below their targets. This can be explained by unexpected deviations from the plan or systems that are not properly configured. It is too expensive to have planners continually track all situations at a granular level to ensure that no deviations or configuration problems occur. We present a machine learning system that predicts service level failures a few weeks in advance and alerts the planners. The system includes a user interface that explains the alerts. This project was experimented in co-operation with Michelin.

Keywords: Supply chain management; manufacturing; machine learning; human-computer interface; explainable AI.

[ł] The work described in the paper was done at JDA Software while the first three authors were working on it.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

[*] Corresponding author: gabrielle.gauthier-melancon@polymtl.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2019

Introduction

Supply chain planning is typically done with multiple optimization systems that differ in scope and planning horizon, from strategic sales and operations planning to near-real-time transportation systems. Despite advanced planning and execution systems, manufacturers and distributors tend to observe service levels below their targets. There are two main reasons for this. First, deviations such as drastic changes in demand, delays in transport, or production problems may occur. Since some plans are generated and fixed for a certain period, perhaps a month, the deviations may not be accounted for until the next planning period, resulting in temporary service level failures. Second, when multiple supply chain systems need to be tightly integrated, there is a risk of undetected mismatches or problems in their configurations that may lead to sub-optimal plans. For more information on supply chain planning, the reader is referred to Stadtler and Kilger (2002).

Recent advances in machine learning (ML) and the growing availability of data have initiated a steady stream of research combining machine learning and supply chain. Nguyen et al. (2018) recently published a survey of big-data analytics for supply chains that classifies the studies by supply chain functions, including *demand management*, *manufacturing*, *warehousing*, and *general supply chain management*. The authors highlight that areas such as demand forecasting and machine maintenance are increasingly using ML. The survey also outlines some gaps in the literature, especially in general supply chain management. In this general category there were two descriptive, four prescriptive, and no predictive applications. The problem described in this paper is a predictive application in the general category. Other papers in this category consider managing sustainability in the supply chain (Papadopoulos et al. 2017) and natural disaster risk management (Ong et al. 2015), which both focus on specific aspects of the problem. No papers included in the survey studied the anticipation of service level failures via modeling multiple segments of the supply chain. This may be because of the difficulty of obtaining real data, given the complexity of supply chains and the nature of the service level failures. Apart from this survey, a lot of papers explore supply chain risk management, which can complement our approach. Some papers explore disruptions impact on the supply chain and how risk propagates (Simchi-Levi et al. 2015, Garvey et al. 2015), while other papers are more focused on how to react and mitigate its impact (Schmitt 2011, Paul et al. 2017).

In this paper, we present a system that uses ML to raise alerts when the supply chain conditions, such as combinations of events, small deviations or inadequate systems configurations, may lead to service level failures. The alerts need to anticipate issues in time for the planners to take corrective actions but not so early that the issues would naturally be accounted for in the next plan. The system focuses the attention of the planners on alerts that are *actionable* (it is possible to avoid the failure), *exclusive* (the issues are not detected by other systems), and *significant* (failures concern important items, so that performing the corrective action is worthwhile). To increase confidence in the results, the tool also aims to *explain* alerts, by identifying their underlying causes and providing the context of potential fixes. The model and the user interface (UI) have been developed in co-operation with Michelin, an international tire manufacturer, which provided the business use case and the data.

In the remainder of this paper, the *Problem Description* section will describe why service level failures occur in supply chains, how planners deal with them today and detail some factors that must be accounted for in the model. In the *Machine Learning Model* section, we will outline how we modeled the problem using ML. Then the section entitled *User Interface* will present the UI that we implemented to navigate through the alerts while building confidence and helping planners with potential fixes. In the *Case Study: The Michelin European Supply Chain* section, we will discuss the results and benefits of the case study we have done with Michelin. The *Appendix* contains some details on the model and the data.

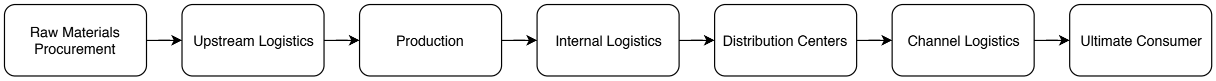
Problem Description

In this section, we discuss why supply chains tend to observe service level below their target, and how planners typically handle this issue. We present how a ML system could help in addressing the problem and the guidelines we followed to ensure that our approach would produce useful alerts.

Service Level Failures

Supply chains can be described as a "... network of organizations that are involved, through upstream and downstream linkages, in the different processes and activities that produce value in the form of products and services in the hands of the ultimate consumer" (Christopher 2005, p.17). The flow of material will typically go through a supply chain such as shown in Figure 1. Organizations may own all segments of their supply chains, or may outsource some sections to reduce complexity.

Figure 1: A typical material flow in a supply chain starts at raw materials procurement and ends when products or services are received at the ultimate consumer.



Notes. The last link can be the end-consumer, retailers or other manufacturers for instance. Some organizations may outsource the management of some sections of the supply chain to third parties.

Typically, the performance of the supply chain can be measured through indicators such as *service level.* For instance, for a given consumer order, the service level can measure whether the desired quantity of products were available at the distribution center (DC) in time for delivery to the *ultimate consumer* requested date. A failure to meet the deadline and/or the desired quantity of items can be cause by either an execution issue or a planning issue. By *execution* issue, we mean that the plan was adequate to fulfill the orders on time, but the plan was not executed as expected, resulting in a failure. These disruptions can be caused by delay in shipments between plants and DCs, or machines failure for instances and are difficult to foresee and prevent. By *planning* issue, we mean that the plan was inadequate in fulfilling the right amount of items on time. These types of issues may be detected early enough to be avoided. Planning issues are usually caused by large deviations in demand that are not accounted for in the most recent plan, or it may be that the supply chain is already at full capacity, hence not able to fulfill the right amount of items. Planning issues can also be the result of inadequate systems configurations. By *configurations*, we refer to the systems parameters and rules that create the supply chain plans, such as safety stock targets, demand forecasting, master plans and such. Systems may not be adequately configured or may not have adapted throughout time. Moreover, as typically manufacturers tend to leverage heterogeneous systems with limited integration, either from different vendors or built in-house, the systems may not be working properly together. These situations may be diluted in aggregated performance metrics since these bad configurations may affect only a subset of items or locations.

Current Process and Need for an Automatic Alerting System

In an ideal world, the planners would continuously monitor the supply chain data and adjust the parameters when they detect situations or patterns that could lead to poor service levels. However, in part because of the high volume of data and the complexity of supply chains, they are usually unable to monitor the data at a granular level. Instead, they adopt a proactive approach for only the most important items, and they resort to a corrective approach (adjusting the parameters only once the service level has fallen significantly below the target) for the vast majority of their products. This is reasonable because supply chains generally operate well, and the experts' time is costly; having the planners investigate low-risk situations would not be profitable. Consequently, there is a need for a system that can identify problematic situations so that the planners can focus their efforts where they are most needed. Moreover, such a system could detect problems that would be missed by human planners. The prediction horizon should cover part or all of the period during which the current plan is frozen. We use the term *alert* to refer to situations that present risk for a given item–location combination.

Useful Alerts

Due to the high number of potential alerts, the system should focus on alerts that are useful for the planners, which we defined as alerts that are exclusive, actionable, and significant, as detailed below. Additionally, the system must provide explanations to increase confidence and help to identify the failures' fix.

By *exclusive*, we mean that the system should generate alerts that are not obvious or detected by other tools. For example, the planners are usually aware of production capacity problems and quality issues. By *actionable*, we mean that alerts should identify situations where the planners can attempt to avoid the failure event. Potential actions include changing the forecast, adjusting the safety-stock levels, and adding transportation options. The prediction horizon should be long enough to ensure a minimal number of available actions. For example, anticipating a stock-out for the next day when the replenishment lead time is one week is unhelpful. By *significant*, we mean that alerts need to concern items and locations that are important (e.g., in terms of volume or strategy), so that the corrective action is worth the effort and cost. Nevertheless, alerts that are not exclusive, actionable or significant may still allow supply chain planners to forewarn customers of delays and highlight recurrent problems that could be alleviated by structural changes in planning processes.

The system also needs to provide explainability around the alerts for two main reasons. First, the users need confidence in the model's results. Second, without the right data, finding the appropriate fix may require a study of multiple planning systems, thus reducing the time saved by the automatic alerting system. A classical spreadsheet approach does not provide the necessary context.

Machine Learning Model

We now discuss how we approached the problem with a classification ML model, summarizing the task and detailing the performance measure and feature set.

Task

We must anticipate service level failures. We use service level targets to convert service levels into a binary classification model: will the service level be above or below this target in the period starting a few weeks from now? These predictions can be made at the most granular level: the item/location/period. Failures are typically less common than successes, leading to two unbalanced classes. Appropriate performance measures for these problems are receiver operating characteristics (ROC) curves and precision/recall curves, sometimes summarized via the area under the curve. Since the experts' time is costly, we choose precision/recall, which allows us to directly answer the question: "How likely is it that this alert will actually be a problem?" by looking at the precision metric.

Feature Engineering

Since raw data cannot be directly input into the model, supply chain conditions are encoded into a set of features representing the different segments as represented in Figure 1. To highlight deviations from the plan, the features generally compare the actual and planned values, as detailed below. They include data from before the period of the prediction, including the service level of the last few periods, and from after the period of prediction, when using projections.

First, *raw materials procurement* can lead to failures when the needed quantities are not received on time, thus affecting production. Relevant features include (1) inventory on-hand vs. production needs and (2) inventory received vs expected inventory.

Second, *production* problems can be good indicators of future failures since the lead time between the plant and the DCs delays their impact, allowing us to foresee potential issues and delays. Relevant features include (1) production plan vs. actual production; (2) inventory on-hand compared to latest forecast; and (3) percentage of production capacity achieved.

Third, between the plant and the DC or between the DC and the ultimate consumer, *logistics* problems may occur in the transportation network or the loading and unloading, delaying the shipment of the items. Although important to understand the past failures, transportation disruptions are less likely to predict future ones. For example, a delay may be weather-related and likely to be resolved within the time horizon. Relevant features include (1) average logistics delays in last period and (2) percentage of logistics capacity achieved.

Finally, *forecasting* deviations and *stock* problems at the DC may indicate that the plan is no longer meeting the demand. Relevant features include (1) cumulative forecasts of the last few periods compared to the customer orders, indicating over- or under-forecasting and (2) inventory on-hand vs. safety-stock target.

Algorithm

We used gradient-boosted decision trees (GBDTs) as implemented in XGBoost (Chen and Guestrin 2016) to solve this problem. GBDTs allow to capture non-linear effects and can also be explainable with the help of approaches such as Shapley values.

GBDTs use machine learning techniques to build trees ensemble that performs a classification or regression task. Each decision tree is trained on the prediction error of the preceding trees. The first tree generally trains on the delta between the predictions and a baseline, such as the average value of the training set. GBDT can perform well in various settings and capture complex non-linear effects, as it is comprised of multiple simpler models (trees) that can learn local behaviors in the data.

One way to explain a model's output is to identify the most important features for each prediction. Additive feature attribution methods (Lundberg and Lee 2017) are approaches that assign a contributing value to each feature for each data point, such that the sum of all contributions is equal to the prediction. Through game theory, this value attribution problem can be seen as a cooperative game, by viewing all players as the different features. The solution to this problem are named Shapley values (Shapley 1953). For each feature, the intuition behind those values is to compare the prediction that is obtained with and without this feature. Unfortunately, computing Shapley values has an exponential complexity, hence approximations must be used. These values can be approximated efficiently for decision trees using a recently proposed method called Tree SHAP (Lundberg et al. 2018), which can be executed in pseudo-polynomial time.

User Interface

We developed a UI to help the planners understand the alerts generated by the system. The UI, shown in Figure 2, provides an interactive dashboard that summarizes the predictions and highlights individual alerts. It is based on both the model output and additional data sources. One of the most important feature of the dashboard, and the motivation for developing it, is that it explains the alerts so that the planners can gain confidence and identify potential corrective actions, as discussed in the subsection entitled *Model Explainability*. Additionally, the dashboard allows the planners to filter the alerts based on various rules so that they can focus on the most useful ones: see the subsection entitled *Navigation and Filtering*. Finally, the UI serves managers by giving them a quick assessment of the health of the supply chain, as detailed in the subsection entitled *Supply Chain Health Check*.

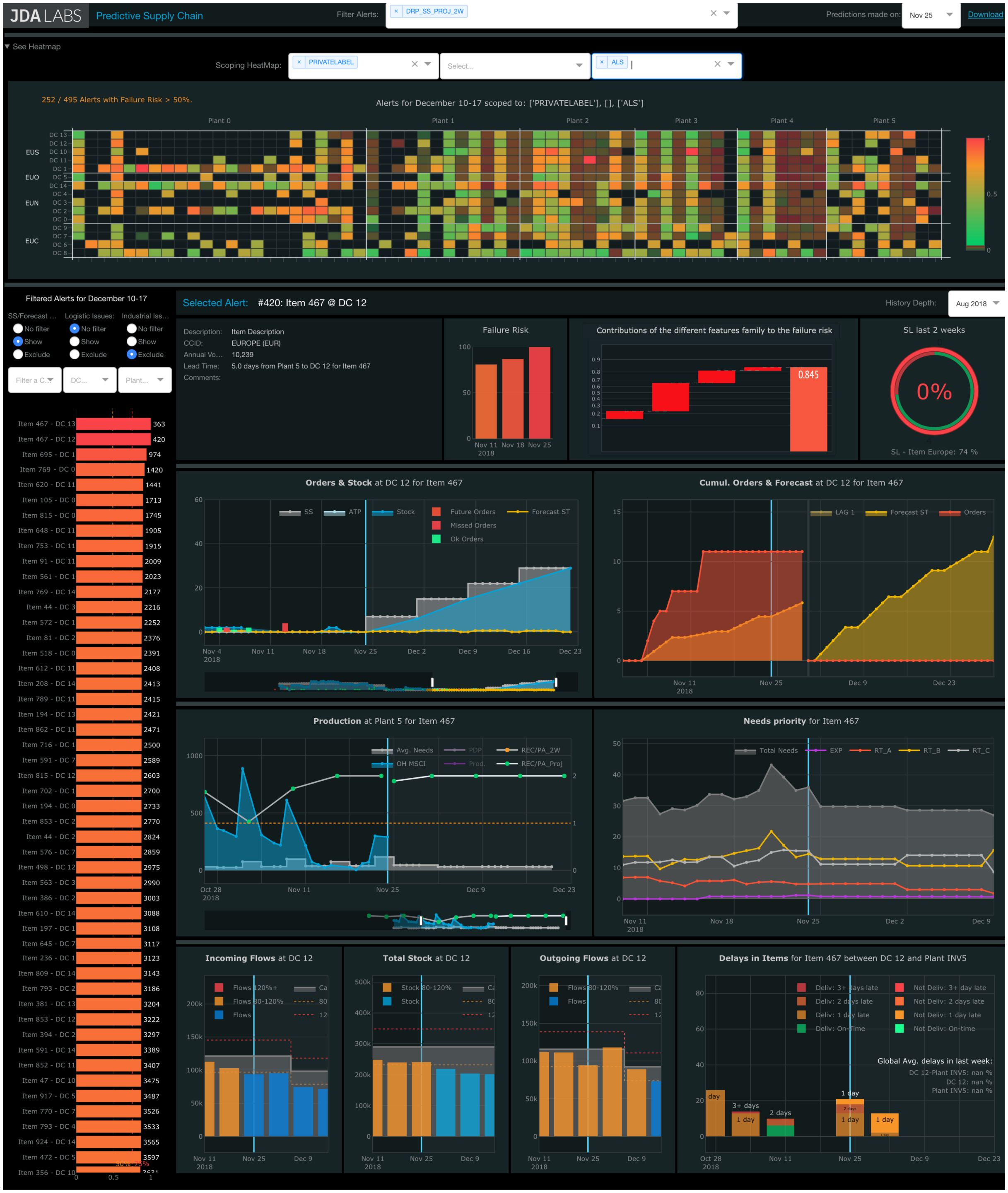
Model Explainability

In ML, it is increasingly recognized that results must be explainable. With the UI, we display the contributing features of each prediction to provide an explanation for the alert and help identify the root causes, we provide context for each alert, and finally we calibrate the model's output so its definition is intuitive for the users and independent of the model.

First, to provide an explanation of an alert and identify its main underlying causes, contributions (Shapley values) of each feature can be summed and grouped by family, each one referring to a different cause and their corresponding corrective actions. In the Appendix, Table 5 indicates the mapping between the features and the families (underlying causes). Because Shapley values are additive, they can be displayed in a waterfall graph such as in Figure 3. In the context of GBDTs, Shapley values actually sum to the difference between the prediction and the baseline. In this example, the rightmost bar displays the overall prediction of 0.747. The other bars illustrate the sum of the contributions for each feature family (sum of the Shapley values by family estimated by Tree SHAP (Lundberg et al. 2018)). Contributions can be positive (in green) and decrease the failure risk or negative (red) and increase it. In classification problems, the prediction is usually computed as log-odds ratio, hence the Shapley values are actually log-odds contributions. So that the length (contributions) of the bars (features family) can be linearly comparable, the graph uses a non-linear (logit) y axis.

Second, the UI provides context on the current conditions via graphs and displays of stock levels, production plans, logistics delays, and so on. These graphs includes the data that is used to create the features. It also displays additional data that the planners need to identify the

Figure 2: We developed an interactive user interface which can be used to have an overview of all alerts and to explore each prediction individually, providing context and explanations to identify potential failure fixes.



JDA LABS Predictive Supply Chain

Filter Alerts: × DRP_SS_PROJ_2W

Predictions made on: Nov 25

Download

▼ See Heatmap

Scoping HeatMap: × PRIVATELABEL | Select... | × ALS

252 / 495 Alerts with Failure Risk > 50%.

Alerts for December 10-17 scoped to: ['PRIVATELABEL'], [], ['ALS']

Filtered Alerts for December 10-17

SS/Forecast ...: ○ No filter ● Show ○ Exclude

Logistic Issues: ● No filter ○ Show ○ Exclude

Industrial Iss...: ○ No filter ○ Show ● Exclude

Selected Alert: #420: Item 467 @ DC 12

History Depth: Aug 2018

Description: Item Description

CCID: EUROPE (EUR)

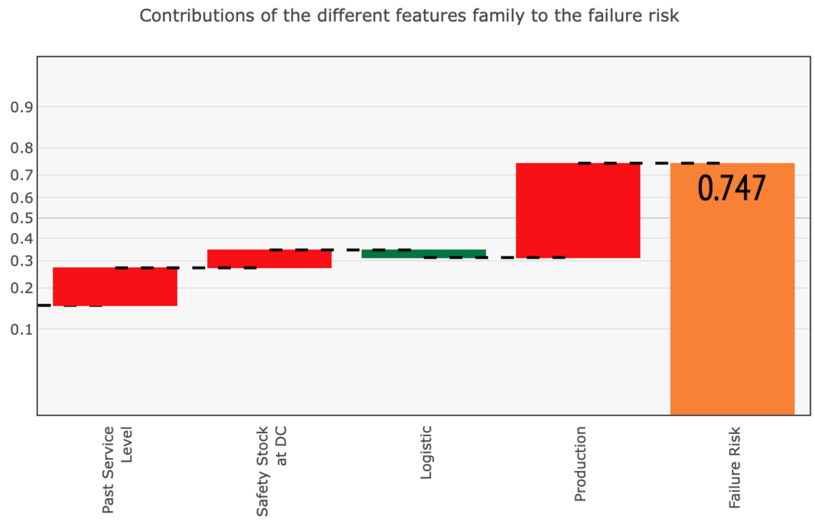
Annual Vo...: 10,239

Lead Time: 5.0 days from Plant 5 to DC 12 for Item 467

Comments:

Notes. The UI was developed using Dash by Plotly. It connects both to the model output and to additional data sources.

Figure 3: This waterfall graph shows the log-odds contributions of each features family, summing to the difference between the baseline (y intercept) and the failure risk (last bar).



Notes. In this example, the main underlying cause is a production problem. The service level in the last few weeks and features representing the safety stock at the DC are also factors which increase the failure risk. Favorable logistic conditions have slightly lowered it.

right corrective actions, so the users do not have to access multiple systems to get the necessary information.

Finally, we calibrate the algorithm's output for the UI so that its meaning is not dependent to the model's features and objective, but also so its interpretation by the system's users matches their perception of risk. Instead of directly using the model output value as the quantification of an alert, we use the precision value of the validation set associated with each output value. We refer to the associated look-up precision as the failure risk. This allows us to compare failure risks from different models and also allows the user to refer to this value as a probability of the model being right, assuming that the new data comes from the same distribution as the validation set.

Navigation and Filtering

Alerts can be filtered out in the UI. We allow planners to provide their own filters for each item/location/period combination. Giving planners the ownership of the filters ensures that the alerts are useful, relevant, and adapted to their supply chain dynamics context. Typical filters include: (1) problematic plants and DCs that are already known to the planners (and so not exclusive to this system), (2) logistics problems causing delays that cannot be avoided (these alerts are not actionable), and (3) low-volume tires and discontinued products (these alerts are not significant). In the UI, it is also possible to filter alerts by underlying cause, to easily deal at the same time with all the situations which can be fixed by the same corrective action.

Supply Chain Health Check

Since the model generates alerts at a granular level, it is possible to group them to gain a quick overview of the supply chain health. In Figure 2, the heatmap in the upper part of the UI displays the alerts organized by item, DC, and plant. The x axis represents the items, grouped by the corresponding plants. The y axis represents the DCs, grouped by the corresponding regions. The color change from green to red represents the failure risk. By glancing at the heatmap, managers can quickly identify if the alerts primarily affect certain items, plants, DCs, or regions. It is also the entry point for exploring each alert in more detail.

Case Study: The Michelin European Supply Chain

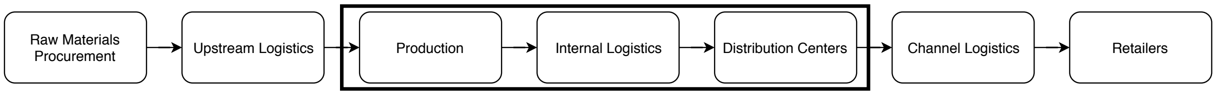
In this section, we discuss how we implemented and tested our ML model and the UI in the context of Michelin's supply chain, specifically the store-and-sell channel. We describe the performance of the model and the benefits that Michelin observed with our approach.

Michelin Context

Michelin is an international manufacturer that produces and sells tires for a vast range of vehicles, from cars and motorcycles to tractors and aircraft. Michelin produces roughly 200M tires per year and has a commercial presence in 170 countries, reaching 13.7% of the global tire market in 2014. In this study, we considered the car tires segment of the Michelin supply chain. For these products, Michelin distinguishes two channels: one for orders placed well in advance (typically large quantities, for car manufacturers or large retailers) and one for orders placed only a few days ahead (typically for local mechanics), called store-and-sell. The latter represents a challenge from a supply perspective since there is little time to plan and potentially a high variability in the demand. We focused on this channel for this case study. We used a prediction horizon of 14 days and a period of a week; meaning we aimed to predict failures for the week starting in 14 days.

Since data gathering and validation is complex and this study is the first, to our knowledge, to use ML to anticipate failures, it was important to limit the data effort and the complexity of the model. We therefore focused on the supply chain segments that were believed to have the most impact on the customer service level. In particular, we excluded raw materials procurement and upstream logistics. Also, for simplicity, we measured the service level at the DC and excluded the channels logistics (i.e., the transport between the DC and the retailer), since this segment usually reflects execution issues more than planning issues. In a nutshell, we considered all operations between the plant and the DCs, indicated in the rectangle in Figure 4.

Figure 4: The rectangular identifies the supply chain segments that were included in the model.



Note. In this visualisation, the ultimate consumer has been identified as retailers to reflect the store-and-sell channel.

For the considered scope, the lack of data remained a challenge. It is not yet standard practice, to our knowledge, to archive all of the supply chain data at the finest granularity. For example, at the beginning of our case study, the metrics *Available To Promise* and *Forecast at the item/DC level* were not archived by Michelin. Since ML models are trained with past data, this led us to discard some data that we believed could have been useful. This indicates that archiving must become the default policy for supply chain data before ML can significantly enter this space. In the *Appendix* can be found the complete set of features that we used.

Model Performance

The project was carried out in two phases. In the first, which lasted seven months, we performed the data gathering and exploration and then developed the model, by focusing on about a hundred 17″ summer tires that had been identified by Michelin as good representatives of the general situation in their supply chain. In the second phase, we performed live tests with the planners on 10 weeks and developed the UI that explains the alerts. We also extended the number of tires in scope to the complete range of Michelin car tires in Europe, about 4000 items, and tested that with the planners for an additional 6 weeks.

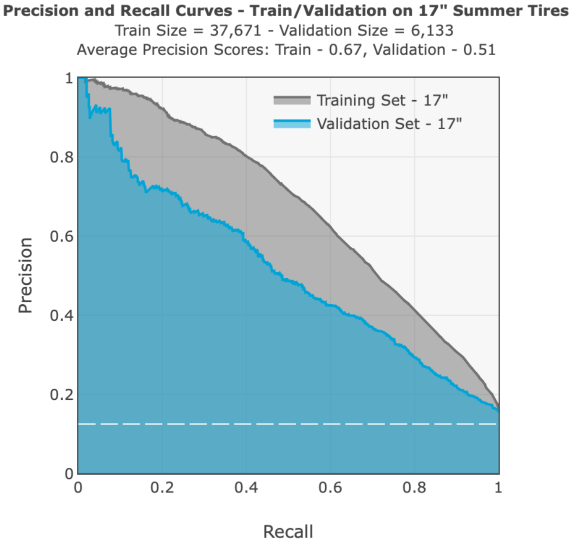
As noted above, the first phase focused on 17″ summer tires. The more than 43k data points covered a period of 14 months and described the supply chain conditions of 95 items, produced at 10 plants (each type of tire was produced in one plant at a time) and stored in 16 DCs. These orders were placed from over 23k customers. We used the first twelve months as the training set and the remaining two months as the validation set. Figure 5 shows the performance of the model, with the x axis representing the recall and the y axis the precision. The mean service level is around 87.5%, so a random classifier would correspond to a horizontal line at $y = 0.125$, as shown with the horizontal white doted line. Our curves are significantly above this, which indicates the predictive value of the proposed model. The optimal hyper-parameters can be found in the *Appendix*.

During the second phase of the project, we decided to explore how the system would perform with more tires. We did not have enough historical data to do a full retraining on all the new tires, so we applied the model trained on the 17″ tires. This test included most of the car tires sold by Michelin in Europe (around 4k items), ranging from 13″ to 22″ and beyond, with 11 months of data. The dataset contains over 500k points, approximately 15 times more than the number in the 17″ training set. Figure 6 shows that the performance of the 17″ model on the full range of tires is similar to its performance for the 17″ tires, thus the model is generalizing well.

User Adoption and Useful Alerts

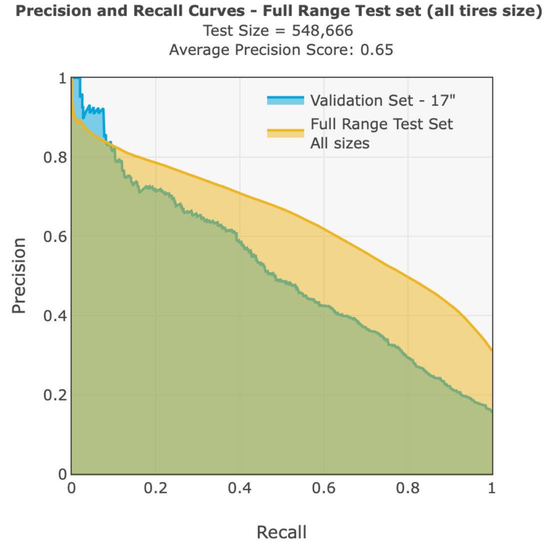
The dynamic tests with the planners in the second phase of the project were crucial for developing a useful system. In total, we met with the planners on 16 weeks (10 weeks on the Summer 17” Tires scope and 6 weeks on the full range scope). We processed the new weekly generated alerts and improved the UI in an iterative way based on their feedback. We added new graphs, to reduce the users’ dependence on other systems to validate the alerts and identify fixes. We also provided better explanations of the alerts, via the graph shown in Figure 3, which was particularly well received. These iterative improvements in the UI empowered the users and helped them to understand and handle the alerts. The users also added filters to hide not exclusive and not actionable alerts, resulting in predictions that were more useful. Figure 7 assesses the model’s

Figure 5: Precision-recall curves for training and validation on 17" Summer Tires.



Notes. The gray area represents the training performance on the first 12 months of data for 17" summer tires. The blue area is the validation performance on the following 2 months of data for the same tires. The horizontal dotted line displays the mean service level in the dataset.

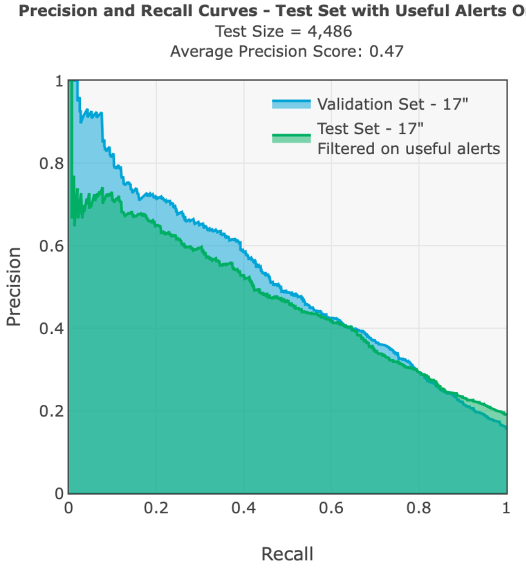
Figure 6: Precision-recall curves on a test set containing the full range of tires.



Notes. The blue area is the validation set as shown on Figure 5. The yellow area displays the performance on all sizes of tires (around 4k items for 11 months of data). The performance is similar, indicating that the model generalize well to new items.

performance on the subset of these useful alerts. The performance is slightly lower than that for all the alerts, and in particular the system does not reach a high precision level. This is because the users have filtered out easy-to-find problems captured by other systems, i.e. not exclusive alerts. Additional filters were also added to classify alerts by significance, identifying big runner, medium runner and long tail products.

Figure 7: Precision-recall curve on a new test set containing only exclusive and actionable alerts.



Notes. The blue area represents the validation set with 17" tires as shown in Figure 5 and 6. The green area displays the performance on a new test set of 17" summer tires, filtered down to the alerts which were exclusive and actionable only. Alerts concerning tires of all significance were included (big runners, medium runners and long tail products).

Capturing Not Adequate System Configurations

Looking at alerts generated at the finest granularity allowed the users to identify recurrent issues that were diluted in the high-level metrics. In the first weeks of test, the system's alerts led to some structural changes in Michelin's processes, identifying systems being either wrongly configured or no more adapted to the current supply chain dynamics. These initial changes had a positive impact beyond the tires in the scope of this project, as they fixed the problems at their source.

First, through multiple alerts, Michelin detected a problem in the computation of the safety stock for their small-volume tires, affecting around 25k units on the week where the issue was discovered. To temporarily fix the problem, the planners manually changed the safety-stock values of the item–DC combinations for which service-level failures were predicted. After a few weeks, it was detected that the overwriting process was faulty as well, so the safety-stock changes were not being taken into account. A month after the discovery of these issues, the amount of items affected by the problem was reduced to around 2.8k tires, reducing the volume of the problem by 89%. Fixing these two issues was one of the first and most significant benefit of the system.

Second, the system detected multiple situations where the underlying cause was a forecasting issue, highlighting that the forecasting algorithm was not sufficiently dynamic. The system identified individual cases of under-forecasting and raised awareness of these issues, motivating further work at Michelin on forecast improvements.

Third, the system detected multiple situations where an item that was to be discontinued was phased out too aggressively given the customer demand. The planners reached out to the appropriate team to address this issue.

Corrective Actions to Address Deviations

Once the most important structural changes were put in place, planners used the alerts on a weekly basis to identify more fine-grained issues on specific items and DCs, such as deviations in the demand, logistics, and production, and intervened to avoid the failures. Of the 16 weeks of tests, the last six weeks (on the full scope of tires) were used extensively to capture the impact at scale of corrective actions on the service level. At first, planners took manual corrective actions which did not affect a big proportion of the tires. However, during the last weeks of the live tests, planners implemented an automatic process to take corrective actions based on the alert's underlying cause. In total, over the course of three weeks, planners were able to act on 23% of the store-and-sell volume. For all tires impacted by corrective actions, they observed a gain of 10 points in their service level (measured out of 100) after 3 weeks. In Table 1 is detailed the results by volume of tires. The total is weighted by the volume of tires in each category.

Table 1: Changes in service level observed over 3 weeks of corrective actions.

Changes in Service Level (/100)	Corrective Actions	No Action
Big Runners	+4	-4
Medium Runners	+14	+3
Long Tail	+14	+5
Total (weighted by volume)	+10	0

Notes. The values represent the changes in service level which are measured out of 100. Service level changes were measured before and after the prediction horizon, hence three weeks apart. In total, tires on which an action was taken as a result of an alert gained in average 10 points of service level over the three weeks of test. Tires on which no action were taken have not gained any points.

Meanwhile, the tires which plans were revised also faced a small decrease in their demand (2%), as compared to tires on which no actions were taken (10%), as shown in Table 2. This indicates that the gains in service levels were not only due to the negative trend in the demand.

Table 2: Changes in demand observed over 3 weeks of corrective actions.

Changes in Demand (%)	Corrective Actions	No Action
Big Runners	0%	-3%
Medium Runners	-7%	-10%
Long Tail	-2%	-21%
Total (weighted by volume)	-2%	-10%

Notes. The tests were done during weeks where the demand of tires tend to naturally decrease. However, for tires on which actions were taken, the demand only reduced of 2%, as opposed to other tires which reduced of 10%. This shows that the gains in service level shown in Table 1 are not only due to the negative trend in the demand.

A gain in service level can be achieved at the expense of an increase in the total inventory and storage cost. However, for tires on which actions were taken, the days of coverage (stock compared to demand) increased of only 2 days, while having a decrease of 2% in their demand. This was considered as stock and cost efficient for Michelin. Details can be find in Table 3.

Table 3: Changes in days of coverage (Stock/Demand) observed over 3 weeks of corrective actions.

Changes in Days of Coverage (Days)	Corrective Actions	No Action
Big Runners	+1	+6
Medium Runners	+2	+9
Long Tail	+4	+15
Total (weighted by volume)	+2	+8

Notes. For tires on which actions were taken, the coverage of the stock as compared to the demand increased of 2 days only, where, as comparison, tires on which no action were taken, it increased of 8 days. This could be due in part to the decrease of the demand. However, the increase of 2 days of coverage for a gain of 10 points in service level was very reasonable and cost efficient for Michelin.

Improving the service level also led to lowering the orders cancellation rate. These results confirmed the benefits of the system, and resulted in increased sales and revenue over the course of the dynamic tests. By identifying situations at risk, planners were able to take corrective actions on risky situations on which they had no visibility before. The system also had good results with big runners, a category in which we suspected gains would be minimal due to the existing manual tracking done by planners on these items.

Due to the new capabilities and disruptive changes that the ML system brings, Michelin needs to adapt its current processes to allow planners to intervene more easily on the supply chain. Today, the available corrective actions for planners are still limited, the main one being adjusting the safety stock at the DC. For example, because Michelin's forecasting process is not done at the most granular level, it is yet impossible for planners to change the forecast of a specific instance. Adjusting safety stock is the most practical lever planners have to re-allocate tires or increase production. Adding new corrective actions could allow the planners to more readily react to deviations and other issues.

Managerial Insights

The tool also had other benefits. Supply chain management can be complex, and the planners are typically organized by function: safety-stock specialists, forecasting experts, logistic planners, and so on. This makes it difficult to gain a product or customer view of the supply chain. Our system provided an opportunity for different specialists to collaborate and discuss specific aspects of the supply chain mechanics. These discussions helped the planners to broaden their understanding of the supply chain. Our experiments suggest that the tool could also help with the training of new employees, since it gives just the right level of information: it is not necessary to understand in detail all the individual data sources. Lastly, this initiative also changed Michelin's perspective on the importance of archiving data at the most granular level. During this project, they implemented a Data Lake in response to the lack of history in some data sources which caused some issues and challenges. This new mindset will surely open the door to numerous ML projects in the years to come.

Conclusion

We have developed a system that uses ML to predict service level failures in a supply chain. Early on in this project, it has been clear that a good performing algorithm is usually not sufficient to ensure users adoption. It needs to be paired with a system that builds confidence and understanding in the model. Our results and the adoption by the planners show the potential of ML systems to complement existing systems for supply chain management. We believe that this type of approach can be applied to more complex supply networks and to other areas such as production planning.

As this system gets used in production over an extensive period of time, it will probably trend towards identifying less structural changes for more fine-grained issues. A natural extension would be to automatically perform corrective actions based on the predicted failures, so that the supply chain becomes a *self-learning* entity, dealing with deviations in *autopilot* mode. As the availability of data improves, such initiatives will lead the way to a new era in supply chain management.

Acknowledgement

We wish to thank Emmanuel Cadet of Michelin for his constant support throughout this project: it led to valuable learning on both sides. Merci beaucoup!

Appendix

Data Format

For each m historical instances, a set of n features (f_n) is computed at the item, DC and week level. The complete list of features can be found in the next section, Features Set. To indicate failures, aggregated service levels are computed with the mean of the corresponding orders weighted according to the product quantities. These aggregated service levels are then compared to the global service level target and converted into a binary variable, y. If the service level is below the target, it is considered as a failure (1), else a 0. In Table 4 is shown the data format. Note that the product, location and week columns were not used explicitly as features, hence the model can be used with new location and products. The columns in grey were hence dropped before running the model.

Table 4: Data format used in the model.

	Product	Location	Week	f_1	f_2	...	f_n		Failure
x_1	AXP	9272	$W1$					y_1	1
x_2	BFR	875	$W1$					y_2	0
x_3	AXP	9272	$W2$					y_3	0
...	...	...	...					...	...
x_m	GFT	7654	$W700$					y_m	1

Note. Grey columns were dropped out of the model, so the system can produce alerts for new products and locations.

Features Set

Table 5 details the complete list of features used in the final model. Each feature type corresponds to either an item, a DC, a plant or a combination of those, as shown in column Aggregation. Each feature type can also be computed for different time steps, from the 3 weeks preceding the instance to the 3 weeks after (for certain projections), as detailed in column Time. The column Nb indicates the number of features as a result of the different time steps for each feature type. Lastly, the column Underlying Cause shows the mapping between feature types and the features family. These mappings are used in particular to produce the cumulative contributions graph in Figure 3, as well as to categorize the alerts by cause in the UI to accelerate their resolution. In total, 35 features were used.

Table 5: Features set used in the final model, according to data availability.

Feature Types	Aggregation	Time	Nb	Underlying Cause
(1) Production Plan vs Actual Production	Item-Plant	t0	1	Production
(2) Inventory and Production Plan vs Needs	Item-Plant	t0:t+2	9	Production
(3) Inventory vs Safety Stock	Item-Plant	t-2:t+2	5	Production
(4) Average logistic delays	Item-Plant-DC	t-3:t0	4	Logistic
(5) Total Stock (all items) vs Capacity	DC	t0	1	Logistic
(6) Inventory vs Safety Stock	Item-DC	t-3:t0	4	Safety Stock at DC
(7) Projected Safety Stock vs current Safety Stock	Item-DC	t1:t3	3	Safety Stock at DC
(8) Past Service Level (Volume tires)	Item-DC	t-3:t0	4	Unknown
(9) Past Service Level (Count orders)	Item-DC	t-3:t0	4	Unknown

Note. For feature type (2), 3 different needs projections were used for each time step, resulting in 3x3 features in total.

Algorithm

The optimal hyperparameters for the GBDT using XGBoost were the following: n_estimators=75, learning_rate=0.1 and max_depth=5. The other hyperparameters were kept at their default value.

References

Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794 (ACM).

Christopher M (2005) *Logistics and Supply Chain Management: Creating Value-adding Networks*. Financial Times Series (FT Prentice Hall), 3rd edition.

Garvey MD, Carnovale S, Yeniyurt S (2015) An analytical framework for supply network risk propagation: A bayesian network approach. *European Journal of Operational Research* 243(2):618–627.

Lundberg SM, Erion GG, Lee SI (2018) Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888* .

Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 4765–4774.

Nguyen T, Li Z, Spiegler V, Ieromonachou P, Lin Y (2018) Big data analytics in supply chain management: A state-of-the-art literature review. *Computers & Operations Research* 98:254–264.

Ong JBS, Wang Z, Goh RSM, Yin XF, Xin X, Fu X (2015) Understanding natural disasters as risks in supply chain management through web data analysis. *International Journal of Computer and Communication Engineering* 4(2):126.

Papadopoulos T, Gunasekaran A, Dubey R, Altay N, Childe SJ, Fosso-Wamba S (2017) The role of big data in explaining disaster resilience in supply chains for sustainability. *Journal of Cleaner Production* 142:1108–1118.

Paul SK, Sarker R, Essam D (2017) A quantitative model for disruption mitigation in a supply chain. *European Journal of Operational Research* 257(3):881–895.

Schmitt AJ (2011) Strategies for customer service level protection under multi-echelon supply chain disruption risk. *Transportation Research Part B: Methodological* 45(8):1266–1283.

Shapley LS (1953) A value for n-person games. *Contributions to the Theory of Games* 2(28):307–317.

Simchi-Levi D, Schmidt W, Wei Y, Zhang PY, Combs K, Ge Y, Gusikhin O, Sanders M, Zhang D (2015) Identifying risks and mitigating disruptions in the automotive supply chain. *Interfaces* 45(5):375–390.

Stadtler H, Kilger C (2002) *Supply chain management and advanced planning*, volume 4 (Springer).