

Système Prédictif pour la Synchronisation Production-Transport : Implémentation Pratique chez Cevital

SIDI SAID Hocine¹ et BENYELLES Abderrahim¹

Encadrés par : BOUZID M. Cherif¹

Résumé

Ce projet de fin d'études a porté sur la conception d'un système prédictif permettant d'anticiper les besoins en camions chez Cevital Agroalimentaire, sur un horizon de sept jours au lieu d'un seul. L'objectif principal était de prédire le nombre de camions en s'appuyant sur des techniques d'apprentissage automatique. Le travail a été structuré en plusieurs volets : la prévision de la demande, la durée des arrêts de production et les quantités produites, l'ordonnancement journalier, et la génération automatique des programmes de dispatching. Une boucle a été mise en place pour intégrer toutes ces composantes. Malgré certaines limites, le système propose une amélioration concrète des performances logistiques de l'entreprise.

Mots-clés : Machine Learning, Prévision logistique, XGBoost, Random Forest, Synchronisation production-transport

1. Introduction

L'optimisation des chaînes logistiques constitue un enjeu stratégique majeur pour les entreprises industrielles dans un contexte de concurrence accrue et d'exigences croissantes en termes de réactivité [1]. La coordination entre production et transport représente un défi particulièrement critique, nécessitant une synchronisation précise des activités pour maintenir la compétitivité.

Les approches traditionnelles de planification logistique, basées sur des horizons courts, génèrent des dysfonctionnements opérationnels significatifs. L'émergence des techniques d'apprentissage automatique offre de nouvelles perspectives pour améliorer la précision des prévisions et étendre les horizons de planification [2].

Dans le secteur agroalimentaire, cette problématique s'intensifie par la nature des produits et la complexité des processus de transformation. La planification à court terme (J+1) contraint les prestataires de transport qui peinent à mobiliser leurs ressources dans des délais restreints, engendrant des surcoûts et des dysfonctionnements [3].

Les méthodes d'intelligence artificielle, notamment les algorithmes d'ensemble comme Random Forest et XGBoost, ont démontré leur efficacité pour traiter des données complexes et non-linéaires dans divers domaines d'application [4]. Leur adaptation aux problématiques logistiques industrielles constitue un axe de recherche prometteur.

Cette recherche propose le développement d'un système prédictif intégré exploitant les techniques d'apprentissage automatique pour étendre l'horizon de planification de J+1 à J+7. L'objectif principal consiste à démontrer la faisabilité et l'efficacité d'une approche multi-modèles pour optimiser la synchronisation production-transport dans un contexte industriel agroalimentaire.

La contribution de cette étude réside dans l'intégration de quatre modules prédictifs (demande, arrêts, ordonnancement, production réelle) au sein d'une architecture cohérente, validée sur des données industrielles réelles de la période 2021-2022.

2. Présentation du cas d'étude

L'étude porte sur Cevital Agroalimentaire, complexe agro-industriel basé à Bejaïa., elle s'intéresse à la division margarinerie, qui opère cinq lignes de production automatisées tournant en deux équipes de huit heures. À travers ces installations, ce ne sont pas moins de 28 références qui sont fabriquées, puis distribuées vers 13 Centres de Livraison Régionaux (CLR) répartis à travers tout le pays.

2.1 Problématique

Chez Cevital Agroalimentaire, le processus de dispatching des produits vers les Centres Logistiques Régionaux (CLR) repose sur une planification à J-1 : les transporteurs sont informés des besoins en camions et de leurs destination la veille, et doivent assurer le transport le lendemain. Ceci se fait avec l'envoi de ce plan de dispatching appelé 'programme' qui contient la demande de chaque CLR en termes de palette pour chaque produit ainsi que le nombre de camion nécessaire, pour ainsi avoir à la fin le nombre de camion totale. Toutefois, cette méthode présente rapidement ses limites en pratique :

- Difficultés de mobilisation des transporteurs dans des délais courts
- Envois insuffisants de camions causant des retards de livraison
- Déséquilibres de stock en CLR

Afin de répondre à cette problématique, notre étude propose d'exploiter les données historiques disponibles, en intégrant notamment des informations sur la demande client, la production industrielle, les arrêts (pannes), les stocks et les capacités réelles des flottes de véhicules. L'objectif principal est d'augmenter significativement l'horizon prévisionnel de J+1 à J+7, afin de donner aux prestataires davantage de marge d'anticipation

3. Travaux liés

La planification logistique par des méthodes prédictives a connu un regain d'intérêt significatif ces dernières années, particulièrement dans des secteurs industriels où optimiser la chaîne logistique représente un enjeu stratégique. Plusieurs études se sont intéressées à l'utilisation de l'apprentissage automatique pour anticiper efficacement les besoins en transport, améliorant ainsi les performances opérationnelles.

Les travaux de Feizabadi (2022) ont démontré l'intérêt d'un modèle hybride combinant ARIMAX et réseaux de neurones dans un contexte industriel réel, celui d'un producteur d'acier. En intégrant à la fois les séries chronologiques passées et des variables exogènes, ce modèle a permis de réduire l'erreur de prévision (MAPE) et d'améliorer significativement la rotation des stocks et le cycle de conversion de trésorerie, comparé aux approches traditionnelles comme ARIMA ou les méthodes de lissage exponentielle [5]. De leur côté, Moroff et al. (2021) ont mené une étude comparative de six modèles de prévision, incluant des approches statistiques (ETS, SARIMAX) et des modèles de machine learning comme les réseaux de neurones multicouches (MLP), les forêts aléatoires et XGBoost. Leurs résultats indiquent que bien que les modèles ML apportent souvent une meilleure précision, ils nécessitent un effort d'implémentation important, et leur efficacité dépend fortement du profil de la série temporelle étudiée. Ils insistent sur l'importance d'une analyse produit-spécifique pour choisir l'approche la plus adaptée [6].

L'étude de Jampani et al. (2023) approfondit la portée des algorithmes de machine learning au-delà de la seule prévision de la demande. Elle montre que ces algorithmes sont de plus en plus intégrés dans des fonctions telles que la gestion des stocks, la planification d'itinéraires, ou encore l'évaluation des fournisseurs, grâce à. L'usage de l'apprentissage par renforcement est particulièrement prometteur dans la tarification dynamique et la réaffectation des ressources [7].

Enfin, Tirkolaee et al. (2021) proposent une vision globale des apports du machine learning à tous les maillons de la chaîne logistique : sélection et segmentation des fournisseurs, gestion des risques, prévisions, production, logistique inverse, développement durable, etc. Leur analyse met en évidence que les algorithmes de type arbres de décision ou SVM sont bien adaptés aux tâches de classification, tandis que des techniques plus avancées comme les réseaux profonds (DL) sont performantes pour des tâches complexes de prédiction continue [8].

4. MÉTHODOLOGIE ET ARCHITECTURE SYSTÈME

4.1 Architecture générale

Le système mis en place s'appuie sur une architecture modulaire, structurée autour de quatre composants essentiels :

- Prédiction de la demande : Prédit les volumes de vente par produit et CLR sur 7 jours.
- Prédiction de la durée des arrêts : Estime les durées d'arrêts de production pour chaque ligne.
- Ordonnancement : Génère automatiquement les plans de production journaliers.
- Prédiction de production : Estime les volumes réellement produits.

L'implémentation utilise Python avec les bibliothèques NumPy, Pandas, Scikit-learn, XGBoost et Seaborn.

4.2 Méthodologie création de modèle de prédiction

- Étape 1 : Préparation des données

La préparation débute par le nettoyage des données incluant le traitement des valeurs manquantes et suppression des observations incohérentes.

L'analyse exploratoire (EDA) identifie les patterns temporels, tendances saisonnières et corrélations via visualisations et statistiques descriptives.

- Étape 2 : Construction des variables

Le features engineering exploite la richesse temporelle des données logistiques en créant des variables explicatives pertinentes. Les lags temporels, Les moyennes mobiles et les variables calendaires intègrent jour semaine, mois.

- Étape 3 : Modélisation

Cette étape consiste à séparer les données en ensembles d'entraînement, de validation et de test, à optimiser les hyperparamètres, à implémenter le modèle, puis à l'évaluer à l'aide de plusieurs métriques.

5. Implémentation des Modèles de Prédiction

5.1 Prévission de la Demande

5.1.1 Préparation des données

Le jeu de données de la demande comporte 38 658 observations couvrant la période 2021–2022. La distribution est fortement asymétrique, avec une médiane de 0,87 tonne, une moyenne de 3,19 tonnes et un maximum de 173 tonnes. Plusieurs étapes de préparation ont été réalisées : suppression des valeurs négatives et manquantes, harmonisation des noms de produits ou codes CLR incohérents avec les SKU, et filtrage pour ne conserver que les produits de la gamme margarinerie.

5.1.2 Features engineering

L’analyse exploratoire a mis en évidence des motifs temporels marqués dans les données de demande, justifiant l’enrichissement du jeu de données par des variables dérivées. Deux catégories de variables ont été générées :

- Variables temporelles dérivées :
Afin de capturer les tendances récentes et la dynamique de la demande, plusieurs moyennes mobiles et décalages ont été ajoutés :
 - `Rolling_mean_7`, `rolling_mean_14`, `rolling_mean_30` : moyennes mobiles sur 7, 14 et 30 jours.
 - `Lag_1`, `lag_7`, `lag_30` : valeurs de la demande avec un décalage respectif de 1, 7 et 30 jours.
- Variables calendaires :
Ces variables ont été introduites pour intégrer les effets de la saisonnalité, des cycles hebdomadaires et des événements particuliers :
 - `weekday` : jour de la semaine (pour modéliser les effets hebdomadaires).
 - `Is_weekend` : indicateur des week-ends.
 - `month` et `day` : mois et jour du mois (saisonnalité mensuelle).
 - `is_ramadan` : indicateur du mois de Ramadan, durant lequel une hausse moyenne de 22 % de la demande a été observée.
 - `is_holiday` : indicateur des jours fériés, associés à une baisse moyenne de 21 % de la demande.

5.1.3 Entraînement du modèle

Trois approches de modélisation ont été testé

- Essai 1 - Single Target Model : Modélisation directe avec structure originale (chaque ligne = demande journalière produit × CLR × date).
- Essai 2 - Stacking avec Clustering : Segmentation par clustering hiérarchique basée sur fréquence de demande et tonnage moyen, avec méta-modèle Ridge pour combinaison des prédictions.
- Essai 3 - Multi-Target Model : Restructuration en régression multi-sorties (chaque produit = colonne distincte, chaque ligne = ensemble produits par CLR × date).

Après avoir séparé le dataset en train, validation, test. Et optimisé les hyperparamètres via GridSearch On a obtenu ces résultats

TABLEAU 1. Résultats comparatifs des différents Essais (Demande)

Approche	Modèle	RMSE(Root Mean Squared Error)	MAE(Mean Absolute Error)
Single target	Random Forest	3,38	1.41
Single target	XGBoost	3.77	1.45
Stacked	XGBoost	2.99	1.51
Multi-target	Random Forest	1.04	0.11
Multi-target	XGBoost	0.99	0.11

5.2 Prévisions des temps d'arrêts

5.2.1 Préparation des données

Le jeu de données comprend 15 486 lignes, chacune représentant un arrêt de production sur une ligne spécifique, durant un quart de travail donné.

Plusieurs étapes de prétraitement ont été appliquées pour structurer et nettoyer les données :

- Conversion des dates au format *datetime* afin de faciliter les opérations temporelles.
- Agrégation des données par les variables (*Date*, *N°prod*, *Équipe*, *Quart*, *Ligne*, *Produit*), permettant de regrouper les arrêts et de calculer la durée totale d'arrêt par jour et par ligne de production.
- Conversion des unités de temps en minutes pour homogénéiser toutes les colonnes concernées.
- Suppression des valeurs manquantes afin de garantir la qualité des données utilisées pour l'analyse.

5.2.2 Features engineering

L'analyse exploratoire a mis en évidence des dynamiques temporelles marquées dans les données de demande et d'arrêts de production, ce qui a conduit à la génération de nouvelles variables afin de renforcer la qualité des prédictions. Trois grandes catégories de transformations ont été mises en œuvre :

- Codage des variables catégorielles
Les attributs *Quart*, *Ligne* et *Désignation Produit* ont été convertis en valeurs numériques via un encodage ordinal, permettant une meilleure compatibilité avec les algorithmes de type arbres de décision (Random Forest, XGBoost).
- Extraction de caractéristiques calendaires :
Des variables ont été dérivées à partir des dates pour capturer des effets liés au calendrier :
Day : jour du mois
Month : mois
Year : année
is_weekend : indicateur signalant un week-end
- Variables liées aux arrêts de production :
Pour mieux modéliser le comportement des interruptions, trois indicateurs ont été ajoutés :
prev_stop : durée du dernier arrêt observé
time_since_stop : intervalle de temps depuis ce dernier arrêt
rolling_mean_7 : moyenne glissante sur 7 jours des durées d'arrêt

5.2.3 Entraînement du modèle

Après avoir séparé le dataset en train, validation, test. Et optimisé les hyperparamètres via GridSearch On a obtenu ces résultats

Tableau 2 Comparatif des modèles Random Forest et XGBoost pour la prévision des temps d'arrêts

Métrique	Random Forest	XGBOOST
RMSE	92.01	84.9
MAE	67.33	54.06
R ²	0.687	0.734

Depuis ce Tableau 2 on peut déduire que XGBoost performe le mieux dans ce cas

La Figure suivante permet de visualiser la relation entre les valeurs prédites par le modèle et les valeurs réelles observées pour la durée des arrêts.

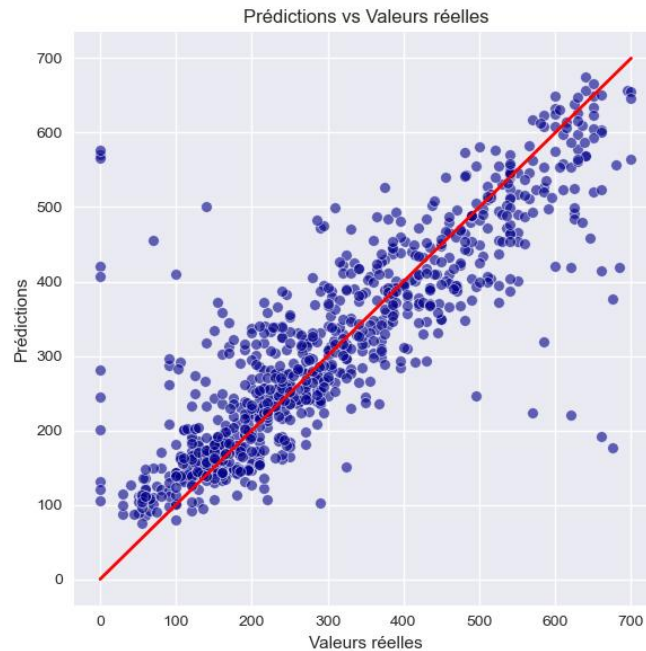


Figure 5-1 Prédictions(XGBoost)vs Valeurs réelles de la durée des arrêts sur l'ensemble de test

5.3 Ordonnancement

L'ordonnancement est formulé comme un problème d'affectation multi-contraintes :

Variables :

$L = \{L_1, L_2, \dots, L_5\}$: lignes de production

$P = \{p_1, p_2, \dots, p_{28}\}$: produits margarinerie

$Capacité[L_j]$: capacité maximale par shift pour ligne j

Objectif :

Minimiser les ruptures de stock en priorisant les produits selon leur couverture :

$$Couverture_p = \frac{Stock_p}{Demande_p} \quad (1)$$

5.3.1 Algorithme d'affectation

Algorithm 1 Ordonnancement de la production pour le jour J+1

```

1: procedure ORDONNANCEMENTPRODUCTION(date_J_plus_1)
2:   for all ligne  $\in$  lignes_production do
3:     produits_eligibles  $\leftarrow$  GET_PRODUITS_LIGNE(ligne)
4:     Trier produits_eligibles par ordre croissant de couverture
5:     for shift  $\in$  {1, 2} do
6:       capacite_restante  $\leftarrow$  capacite_ligne[ligne]
7:       for all produit  $\in$  produits_eligibles do
8:         if couverture[produit] < seuil_securite then
9:           quantite_a_produire  $\leftarrow$  min(stock_target[produit] -
stock_actuel[produit], capacite_restante)
10:          AFFECTER_PRODUCTION(ligne, shift, produit, quantite_a_produire)
11:          capacite_restante  $\leftarrow$  capacite_restante - quantite_a_produire
12:        end if
13:      end for
14:    end for
15:  end for
16: end procedure

```

Figure 5-2 Algorithme d'affectation des produits sur les lignes

5.4 Pr vision de Production R elle

5.4.1 Pr paration des donn es

Le jeu de donn es contient 7 566 observations, chacune correspondant au tonnage produit d'un article sp cifique sur une ligne de production donn e. Ce dataset int gre  galement les pr dictions d'arr ts comme variable explicative. Pour le pr traitement, nous avons pris en charge la gestion des valeurs manquantes afin d'assurer la qualit  des donn es.

5.4.2 Features engineering

Pour la construction des variables explicatives, nous avons int gr  des indicateurs li s   la production et au contexte temporel, ainsi que des caract ristiques sp cifiques aux lignes de production.

Les features utilis es sont les suivantes :

- Dur e d'arr t pr dite (Duree_arret) : dur e estim e des arr ts machine
- Moyenne mobile sur 7 jours de la production (ProdT_roll7_mean) : tendance r cente
-  cart-type mobile sur 7 jours (ProdT_roll7_std) : volatilit  r cente
- Production du jour pr c dent (ProdT_lag1)
- Production du m me jour la semaine pr c dente (ProdT_lag7)

5.4.3 Entra nement du mod le

Apr s avoir s par  le dataset en train, validation, test. Et optimis  les hyperparam tres via GridSearch On a obtenu ces r sultats

Tableau 3.Comparatif entre Random Forest et XGBoost (production)

M�trique	Random Forest	XGBOOST
RMSE	3.48	3.12
MAE	2.27	1.74
R ²	0.93	0.94

On constate que le modèle XGBoost surpasse légèrement le modèle Random Forest sur l'ensemble des métriques évaluées. Il affiche RMSE et MAE plus faibles, ce qui indique une meilleure précision des prédictions. De plus, le coefficient de détermination R^2 est légèrement supérieur, suggérant que XGBoost explique mieux la variance des données. Ces résultats confirment que XGBoost est meilleur dans ce cas.

La Figure 4-3 permet de visualiser la relation entre les valeurs prédites par le modèle et les valeurs réelles observées pour la durée des arrêts

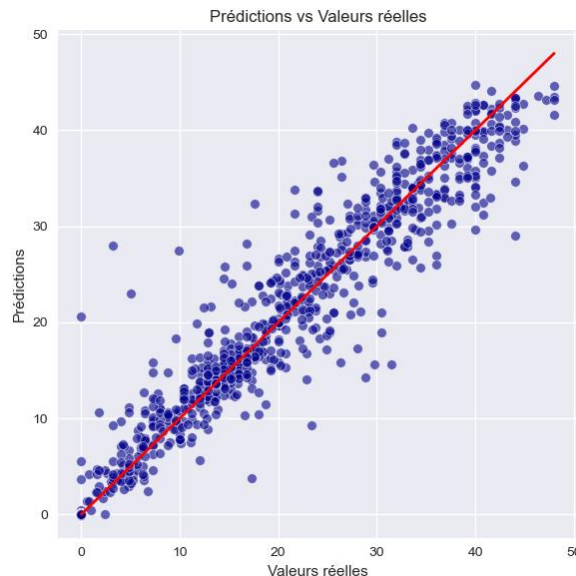


Figure 5-3.Prévision vs réalité modèle XGBoost production

On observe une forte corrélation entre les prédictions et les valeurs réelles, comme en témoigne l'alignement dense des points autour de la diagonale rouge. Cela indique que le modèle parvient à bien capturer la dynamique de production, avec une précision satisfaisante sur l'ensemble des observations.

5.5 Simulation et Intégration des Modèles

Après avoir développé individuellement les modèles de demande, pannes et production, ainsi que l'heuristique d'ordonnancement, nous implémentons une boucle de simulation qui utilise ces composants pour créer un système prédictif intégré permettant de simuler l'activité de production sur un horizon glissant de 7 jours

On commence par faire des prévisions sur la demande sur un horizon de 7 jours, puis on réalise une boucle sur 7 jour en python qui va

- Génération quotidienne du plan de production à partir des prévisions de demande et des stocks disponibles.
- Prédiction des pannes journalières pour chaque ligne et quart de production.
- Prédiction de la production réelle, les temps d'arrêt prédits par le modèle de pannes sont intégrés comme features dans le modèle de production, permettant d'estimer l'impact réel des défaillances sur les volumes produits.
- Mise à jour dynamique du stock. Les quantités produites prédites sont utilisées pour mettre à jour les niveaux de stock, créant les conditions initiales pour la planification du jour suivant

Cette implémentation garantit une cohérence temporelle car les variables de lag et les moyennes mobiles sont recalculées à chaque itération, intégrant les nouvelles observations. Cette mise à jour continue permet aux modèles de s'adapter aux évolutions récentes du système.

5.6 Création des plans de dispatching

Après avoir réalisé les prévisions de la demande et de la production, nous abordons maintenant le cœur de notre problématique : la création des programmes de livraison sur un horizon de J+1 à J+7, afin de donner aux prestataires davantage de marge d'anticipation. Cette solution représente une amélioration majeure par rapport au système actuel de Cevital, qui opère principalement à J+1.

5.6.1 Conversion des Volumes en Palettes

Avant de procéder au dispatching, il faut tout d'abord standardiser les unités de mesure pour faciliter la planification des expéditions. Les prévisions de demande et de production, initialement exprimées en tonnes, doivent être converties en nombre de palettes. Cette conversion s'effectue selon la formule suivante :

$$\text{Nombre de palettes} = \text{Production en tonnes} \div \text{Poids par palette} \quad (2)$$

Sachant que le poids par palette varie selon le type de produit.

5.6.2 Dispatching

Pour créer les programmes de J+1 à J+7, nous avons développé un système de dispatching qui repose sur une architecture décisionnelle à deux niveaux, en fonction de la relation entre la production disponible et la demande totale pour chaque produit. Ce système fonctionne selon les principes suivants :

- Gérer les demandes non satisfaites des jours précédents en leur donnant une priorité plus élevée.
- Répartir les palettes entièrement en cas de production suffisante, ou proportionnellement si la production est insuffisante.
- Reporter les demandes non satisfaites pour les inclure dans la planification du jour suivant.
- Calcul automatisé du nombre de camions nécessaires à l'expédition, basé sur une capacité fixe (24 palettes par camion).

Algorithm 1 Dispatching automatique des camions

```

1: procedure DISPATCHINGAUTOMATIQUE(production_prevue, demandes_prevues)
2:   for all clr ∈ centres_livraison do
3:     palettes_totales ← 0
4:     for all produit ∈ production_prevue do
5:       palettes_produit ← production_prevue[produit] ÷
        poids_par_palette[produit]
6:       palettes_totales ← palettes_totales + palettes_produit
7:     end for
8:     nb_camions[clr] ← [palettes_totales ÷ 24]      ▷ 24 palettes par camion
9:   end for
10:  return nb_camions
11: end procedure

```

Figure 5-4 Algorithme de dispatching de camions

Dans notre projet, nous avons uniquement travaillé sur les produits issus de la margarinerie. Toutefois, dans les cas où le nombre total de palettes expédiées vers une wilaya est inférieur à 24 (moins qu'un camion complet), nous faisons l'hypothèse suivante :

le camion sera complété avec d'autres produits à forte rotation, comme le sucre ou les huiles, également produits par Cevital.

La Figure montre un exemple de programme de livraison généré par notre système pour le jour J+7.

Date: 07/01/2023 - Programme: J+7								
	MARGARINE FEUILLETAGE PLAQ.	MARGARINE FLEURIAL BARQ.	MARGARINE FLEURIAL PLAQ.	SHORTENING 38/40	SMEN EL MEDINA 1.8Kg	SMEN EL MEDINA 500g.	Total	Nb_Camions
CLR ALGER	6	2	0	69	3	5	85	4
CLR ANNABA	0	2	0	2	2	3	9	1
CLR BATNA	0	0	0	0	2	3	5	1
CLR BLIDA	0	2	0	0	3	7	12	1
CLR BOUIRA	5	0	13	0	6	13	37	2
CLR MASCARA	0	2	2	0	2	2	8	1
CLR MEDEA	0	0	4	0	4	7	15	1
CLR MOSTA	0	0	2	0	1	3	6	1
CLR ORAN	0	0	2	0	1	2	5	1
CLR SBA	13	0	2	0	2	7	24	1
CLR SETIF	30	0	6	0	2	3	41	2
CLR TIZI OUZOU	40	1	0	0	3	8	52	3
CLR TLEMCEEN	8	3	0	0	2	2	15	1
TOTAL	102	12	31	71	33	65	314	14

Figure 5-5 exemple de programme de livraison généré par notre système pour le jour J+7.

En comparant le plan présenté dans la Figure 4-5 avec celui de la figure 1-5 qui avait initialement motivé notre question de recherche, nous constatons un gain d'une semaine sur l'horizon de prévision.

Ce temps précieux, obtenu grâce à l'intégration de nos prévisions, permet de réduire la pression sur le fournisseur de transport et augmente significativement les chances de mobiliser un nombre suffisant de camions le jour J.

6. Conclusion

Cette recherche démontre la faisabilité d'un système prédictif intégré pour optimiser la synchronisation production-transport en contexte agroalimentaire. L'approche développée étend significativement l'horizon prévisionnel tout en maintenant une précision satisfaisante.

Le système offre aux entreprises agroalimentaires un outil d'aide à la décision pour améliorer leur coordination logistique et réduire les dysfonctionnements opérationnels. La méthodologie est généralisable à d'autres contextes industriels confrontés aux mêmes défis de synchronisation.

7. Les perspectives d'extension incluent l'intégration temps réel et l'optimisation mathématique des algorithmes

- Intégrer d'autres familles de produits (huiles, sucres...) afin d'obtenir une planification plus globale et réaliste.
- Connecter le système aux bases de données de production pour une mise à jour dynamique.
- Développer une interface utilisateur.

8. Références

- [1] M. Christopher, *Logistics & supply chain management*, Londres: Pearson Education Limited, 2016.
- [2] T. Chen et C. Guestrin, «XGBoost: A Scalable Tree Boosting System,» chez *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, 2016.
- [3] R. Manzini et M. Gamberi, «Logistics optimization in agroalimentary supply chain management,» *International Journal of Logistics Systems and Management*, vol. 6, n° 14, pp. 496-515, 2010.
- [4] L. Breiman, «Random forests,» *Machine Learning*, vol. 45, n° 11, pp. 5-32, 2001.
- [5] M. M. Feizabadi, «Machine learning demand forecasting and supply chain performance,» *Expert Systems with Applications*, vol. 117095, n° 1, Elsevier, p. 198, 2022.
- [6] N. Moroff, L. Röglinger et M. Vetter, «Improving demand forecasting in the process industry: A comparative study of statistical and machine learning algorithms,» chez *Procedia Computer Science*, 2021.
- [7] R. Jampani, P. K. Jain et A. P. S. Rathore, «Machine-Learning Algorithms for Supply Chain Optimisation: A Review,» *Advances in Intelligent Systems and Computing*, vol. 1344, p. 159–175, 2023.
- [8] E. B. Tirkolaei, M. S. Amiri et B. M. Rezaei, «Application of machine learning in supply chain management: a systematic literature review,» *Mathematical Problems in Engineering*, p. 9019071, 2021.