



Dédication

À mes très chers parents, dont l'amour inconditionnel, le soutien constant et les sacrifices silencieux ont été la lumière guidant chacun de mes pas. Aucun mot ne saurait exprimer la reconnaissance profonde que je leur porte. Que Dieu leur accorde santé, bonheur et longue vie.

À mes deux frères et ma sœur, qui ont toujours été à mes côtés avec patience, encouragement et affection. Votre présence m'a donné la force d'avancer et de persévérer dans chaque étape de ce projet. Je vous en suis infiniment reconnaissante.

À tous les membres de ma famille, pour leur amour sincère, leur bienveillance constante et leur présence précieuse dans ma vie.

À mes meilleures amies qui ont partagé avec moi rires, défis et réussites tout au long de ce parcours universitaire. Votre amitié est un trésor que je chérirai toujours.

À toute l'équipe BACOVET, pour leurs précieux conseils et leur soutien qui m'ont guidée avec bienveillance et expertise.

À vous tous,
je dédie ce travail avec tout mon cœur.



Remerciement

Ce travail n'aurait pu aboutir sans le soutien, l'accompagnement et l'implication de plusieurs personnes que je tiens à remercier chaleureusement. Je remercie tout d'abord, pour son suivi attentif, ses conseils avisés et sa disponibilité tout au long de ce stage. Son encadrement a été pour moi un véritable appui, tant sur le plan méthodologique qu'humain. Je tiens particulièrement à remercier **Monsieur Naoufel Bhour**i pour son encadrement académique rigoureux, sa bienveillance et ses orientations précieuses, qui ont largement contribué à la réussite de ce projet.

Je tiens également à exprimer ma gratitude à **Madame Dhoha Ben Brahim**, mon encadrante professionnelle, pour sa confiance, son accueil et son accompagnement tout au long de ce projet. J'adresse également mes sincères remerciements à toute l'équipe de l'entreprise pour leur accueil chaleureux, leur disponibilité et leur collaboration bienveillante, qui ont largement facilité mon intégration et enrichi mon apprentissage.

Ma gratitude s'adresse aussi à membres du jury, pour avoir accepté d'évaluer mon travail et pour l'intérêt qu'elles y ont porté. Leurs remarques me seront précieuses pour la suite de mon parcours.

À toutes celles et ceux qui ont contribué, de près ou de loin, à la réussite de ce stage, je dis merci du fond du cœur.

Table des matières

| | |
|--|-----------|
| Liste des figures | x |
| Liste des tableaux | xi |
| INTRODUCTION GÉNÉRALE | 2 |
| 1 Cadre du projet et étude de l'existant | 4 |
| 1.1 Introduction | 4 |
| 1.2 Présentation de l'entreprise d'accueil : Bacovet | 4 |
| 1.2.1 Historique et identité | 4 |
| 1.2.2 Partenariats stratégiques et présence mondiale | 5 |
| 1.2.3 Certifications de Bacovet | 5 |
| 1.3 Organisation et organigramme | 6 |
| 1.4 Processus global chez BACOVET | 6 |
| 1.4.1 Processus détaillé de l'atelier de coupe | 7 |
| 1.4.2 Durabilité et responsabilité environnementale | 12 |
| 1.5 Cadre du projet | 13 |
| 1.6 Conclusion du chapitre | 14 |
| 2 Mener une action d'amélioration d'un atelier de travail avec Audit, les outils lean 4.0 | 15 |
| 2.1 Introduction | 15 |
| 2.2 Partie I : Audit de maturité digitale | 16 |
| 2.2.1 Introduction a l'audit IMPULS | 16 |
| 2.2.2 Objectifs de l'audit | 16 |
| 2.2.3 Phase Innover : Propositions d'amélioration | 16 |
| 2.2.3.1 Constat principal | 16 |
| 2.2.3.2 Proposition : Algorithme intelligent de gestion des tables de matelassage | 16 |
| 2.2.3.3 Application pratique | 17 |
| 2.2.3.4 Bénéfices attendus | 18 |
| 2.2.4 Méthodologie d'évaluation | 18 |
| 2.2.4.1 Choix du questionnaire IMPULS | 18 |
| 2.2.4.2 Mode de diffusion | 18 |
| 2.2.4.3 Cible de l'enquete | 18 |
| 2.2.5 Déroulement de l'enquete | 19 |
| 2.3 Résultats de l'évaluation de la maturité digitale | 19 |

| | | |
|---------|---|----|
| 2.3.1 | Méthodologie d'évaluation | 19 |
| 2.3.2 | Détail des scores par critère | 20 |
| 2.3.3 | Résultats globaux | 21 |
| 2.3.4 | Analyse par domaine | 21 |
| 2.3.4.1 | Domaine A : Efficacité des processus | 22 |
| 2.3.4.2 | Domaine B : Automatisation, innovation et intégration numérique | 22 |
| 2.3.4.3 | Domaine C : Gestion des performances et des flux | 22 |
| 2.3.4.4 | Domaine D : Amélioration continue et flexibilité | 22 |
| 2.3.5 | Conclusion des résultats | 22 |
| 2.4 | Partie II : Application de la méthodologie DMAIC | 23 |
| 2.4.1 | Introduction a la méthode DMAIC | 23 |
| 2.4.2 | Phase 1 : Define (Définir) | 23 |
| 2.4.2.1 | Objectif de la phase Define | 23 |
| 2.4.2.2 | Systèmes d'information utilisés dans l'atelier de coupe | 23 |
| 2.4.2.3 | Outil utilisé : Carte mentale | 24 |
| 2.4.2.4 | Résumé des points clés identifiés | 26 |
| 2.4.2.5 | Conclusion de la phase "Définir" | 26 |
| 2.4.3 | Phase 2 : Measure (Mesurer) | 26 |
| 2.4.3.1 | Objectif de la phase de mesure | 26 |
| 2.4.3.2 | Méthodologie de mesure | 26 |
| 2.4.3.3 | Description synthétique des processus mesurés | 30 |
| 2.4.3.4 | Constats clés de la collecte | 30 |
| 2.4.3.5 | Paramètres techniques de l'atelier (référence) | 30 |
| 2.4.3.6 | Synthèse de la phase de mesure | 31 |
| 2.4.4 | Phase 3 : Analyze (Analyser) | 31 |
| 2.4.4.1 | Objectif de la phase Analyse | 31 |
| 2.4.4.2 | Méthodes d'analyse utilisées | 31 |
| 2.4.4.3 | Analyse des temps réels par activité | 31 |
| 2.4.4.4 | Analyse des perturbations | 32 |
| 2.4.4.5 | Capacités et limites techniques observées | 32 |
| 2.4.4.6 | Diagramme d'Ishikawa | 32 |
| 2.4.4.7 | Analyse directe des données mesurées | 33 |
| 2.4.4.8 | Synthèse des causes racines identifiées | 34 |
| 2.4.4.9 | Conclusion de la phase Analyse | 34 |
| 2.4.5 | Phase 4 : Improve (Améliorer) | 34 |
| 2.4.5.1 | Constat principal | 34 |
| 2.4.5.2 | Proposition : Algorithme intelligent de gestion des tables de matelassage | 34 |
| 2.4.5.3 | Application pratique | 36 |
| 2.4.5.4 | Bénéfices attendus | 37 |
| 2.4.6 | Phase Contrôler : Pérennisation des améliorations | 37 |
| 2.4.6.1 | Objectifs de la phase de contrôle | 37 |
| 2.4.6.2 | Mise en place d'indicateurs de performance (KPI) | 38 |
| 2.4.6.3 | Système de suivi et de visualisation | 38 |
| 2.4.6.4 | Mécanismes de contrôle et d'alerte | 38 |

| | | |
|----------|---|-----------|
| 2.4.6.5 | Standardisation des procédures | 39 |
| 2.4.6.6 | Amélioration continue | 39 |
| 2.4.6.7 | Conclusion de la phase Contrôler | 39 |
| 3 | CRISP-ML(Q) phases | 41 |
| 3.1 | Introduction | 41 |
| 3.1.1 | Vue d'ensemble du processus CRISP-ML(Q) | 41 |
| 3.2 | Outils et bibliothèques utilisés | 42 |
| 3.2.1 | Introduction | 42 |
| 3.2.2 | Écosystème Data Science et Machine Learning | 43 |
| 3.2.2.1 | Bibliothèques de manipulation de données | 43 |
| 3.2.2.2 | Bibliothèques de Machine Learning | 44 |
| 3.2.2.3 | Bibliothèques de visualisation | 45 |
| 3.2.3 | Frameworks de développement | 46 |
| 3.2.3.1 | Backend et API | 46 |
| 3.2.3.2 | Frontend et interface utilisateur | 47 |
| 3.2.4 | Outils d'optimisation et d'ordonnancement | 48 |
| 3.2.5 | Infrastructure et DevOps | 49 |
| 3.2.6 | Stack technologique complète | 50 |
| 3.2.7 | Justification des choix et intégration CRISP-ML(Q) | 50 |
| 3.3 | Phase 1 : Comprehension metier (Business Understanding) | 52 |
| 3.3.1 | Contexte strategique et enjeux | 52 |
| 3.3.1.1 | Contexte industriel | 52 |
| 3.3.2 | Business Model Canvas | 52 |
| 3.3.3 | Objectifs metier detaillés | 53 |
| 3.3.3.1 | Objectifs strategiques | 53 |
| 3.3.3.2 | Objectifs operationnels quantifiables | 54 |
| 3.3.3.3 | Objectifs techniques ML | 54 |
| 3.3.4 | Analyse approfondie des parties prenantes | 54 |
| 3.3.4.1 | Matrice pouvoir-interet | 54 |
| 3.3.4.2 | Besoins detaillés par profil utilisateur | 54 |
| 3.3.5 | Analyse des processus metier | 56 |
| 3.3.5.1 | Cartographie du processus actuel (AS-IS) | 56 |
| 3.3.5.2 | Processus cible optimise (TO-BE) | 57 |
| 3.3.6 | Analyse des risques metier | 58 |
| 3.3.6.1 | Registre des risques | 58 |
| 3.3.6.2 | Plan de mitigation des risques critiques | 59 |
| 3.3.7 | Criteres de succes et metriques de performance | 60 |
| 3.3.7.1 | Criteres techniques ML | 60 |
| 3.3.7.2 | Criteres metier operationnels | 60 |
| 3.3.7.3 | Criteres de qualite logicielle | 60 |
| 3.3.7.4 | Criteres financiers | 61 |
| 3.3.8 | Contraintes et hypotheses du projet | 61 |
| 3.3.8.1 | Contraintes identifiees | 61 |
| 3.3.8.2 | Hypotheses du projet | 62 |
| 3.3.9 | Synthese de la phase Business Understanding | 63 |

| | | |
|---------|--|----|
| 3.4 | Phase 2 : Compréhension des données (Data Understanding) | 63 |
| 3.4.1 | Objectifs de la phase Data Understanding | 63 |
| 3.4.2 | Inventaire et collecte des données | 63 |
| 3.4.2.1 | Sources de données identifiées | 63 |
| 3.4.2.2 | Caractéristiques des sources de données | 65 |
| 3.4.2.3 | Dataset principal : PSC_X_1 - COUPE.csv | 65 |
| 3.4.3 | Dictionnaire de données | 65 |
| 3.4.4 | Exploration des données | 66 |
| 3.4.4.1 | Analyse du dataset principal | 66 |
| 3.4.4.2 | Variables d'intérêt | 66 |
| 3.4.5 | Analyse de la qualité des données | 67 |
| 3.4.5.1 | Valeurs manquantes | 67 |
| 3.4.5.2 | Valeurs aberrantes | 67 |
| 3.4.5.3 | Coherence des données | 67 |
| 3.4.6 | Analyse exploratoire des données | 67 |
| 3.4.6.1 | Distribution des variables | 67 |
| 3.4.6.2 | Correlations | 67 |
| 3.4.6.3 | Patterns temporels | 68 |
| 3.5 | Phase 3 : Préparation des données (Data Preparation) | 68 |
| 3.5.1 | Objectifs de la phase Data Preparation | 68 |
| 3.5.2 | Nettoyage des données | 68 |
| 3.5.2.1 | Traitement des valeurs manquantes | 68 |
| 3.5.2.2 | Traitement des valeurs aberrantes | 69 |
| 3.5.2.3 | Standardisation des formats | 70 |
| 3.5.2.4 | Validation de la coherence | 71 |
| 3.5.3 | Ingénierie des caractéristiques (Feature Engineering) | 71 |
| 3.5.3.1 | Stratégie de feature engineering | 71 |
| 3.5.3.2 | Workflow de feature engineering | 72 |
| 3.5.3.3 | Features temporelles | 72 |
| 3.5.3.4 | Features dérivées (domaine métier) | 73 |
| 3.5.3.5 | Encodage des variables catégorielles | 74 |
| 3.5.3.6 | Normalisation et standardisation | 75 |
| 3.5.3.7 | Sélection de features | 75 |
| 3.5.4 | Segmentation des données | 76 |
| 3.5.4.1 | Division temporelle | 76 |
| 3.5.4.2 | Stratification | 76 |
| 3.5.5 | Validation de la préparation | 76 |
| 3.5.5.1 | Métriques de qualité | 76 |
| 3.5.5.2 | Tests de régression | 77 |
| 3.5.6 | Pipeline de données | 77 |
| 3.5.6.1 | Architecture du pipeline | 77 |
| 3.5.6.2 | Composants du pipeline | 77 |
| 3.5.6.3 | Architecture du pipeline de données | 77 |
| 3.5.6.4 | Orchestration | 78 |
| 3.6 | Phase 3 (suite) : Cadre d'assurance qualité | 79 |
| 3.6.1 | Introduction au cadre qualité CRISP-ML(Q) | 79 |

| | | |
|----------|--|-----------|
| 3.6.1.1 | Framework d'assurance qualite | 79 |
| 3.6.2 | Metriques de qualite des donnees | 79 |
| 3.6.2.1 | Framework de qualite des donnees | 79 |
| 3.6.2.2 | Tests de qualite automatises | 80 |
| 3.6.2.3 | Monitoring de la qualite des donnees | 80 |
| 3.6.3 | Portes de qualite des modeles (Quality Gates) | 81 |
| 3.6.3.1 | Framework de validation multi-niveaux | 81 |
| 3.6.3.2 | Matrice de validation des modeles | 81 |
| 3.6.3.3 | Tests de robustesse | 81 |
| 3.6.4 | Framework de monitoring en production | 82 |
| 3.6.4.1 | Architecture de monitoring | 82 |
| 3.6.4.2 | Dashboards de monitoring | 83 |
| 3.6.4.3 | Systeme d'alertes intelligent | 84 |
| 3.6.5 | Strategie de tests A/B | 84 |
| 3.6.5.1 | Framework de tests A/B | 84 |
| 3.6.5.2 | Deploiement canary | 85 |
| 3.6.6 | Gouvernance des modeles ML | 85 |
| 3.6.6.1 | Cycle de vie des modeles | 85 |
| 3.6.6.2 | Registre des modeles | 86 |
| 3.6.6.3 | Documentation et tracabilite | 86 |
| 3.6.7 | Synthese du cadre qualite | 87 |
| 3.7 | Synthese et perspectives | 87 |
| 3.7.1 | Bilan des phases 1-3 | 87 |
| 3.7.2 | Preparation aux phases suivantes | 88 |
| 3.7.3 | Risques identifies et mitigations | 88 |
| 4 | Modeling, Evaluation, Deployment (MLOps) | 89 |
| 4.1 | Introduction | 89 |
| 4.2 | Phase 4 : Modélisation (Modeling) | 89 |
| 4.2.1 | Sélection des techniques de modélisation | 89 |
| 4.2.1.1 | Modèle de prédiction du temps | 89 |
| 4.2.1.2 | Justification du choix : XGBoost | 90 |
| 4.2.1.3 | Formulation mathématique de XGBoost | 90 |
| 4.2.1.4 | Hyperparamètres optimisés | 92 |
| 4.2.1.5 | Complexité algorithmique | 92 |
| 4.2.1.6 | Modèle d'optimisation de la planification | 93 |
| 4.2.2 | Ingénierie des caractéristiques avancées | 93 |
| 4.2.2.1 | Features temporelles | 93 |
| 4.2.2.2 | Features d'interaction | 93 |
| 4.2.2.3 | Features de contexte | 93 |
| 4.2.3 | Validation et sélection de modèles | 93 |
| 4.2.3.1 | Stratégie de validation | 93 |
| 4.2.3.2 | Sélection des hyperparamètres | 94 |
| 4.3 | Implémentation des modèles | 94 |
| 4.3.1 | Modèle de prédiction du temps (Time Predictor) | 94 |
| 4.3.1.1 | Architecture du modèle | 94 |

| | | |
|----------|---|------------|
| 4.3.1.2 | Caractéristiques du modèle final | 94 |
| 4.3.2 | Modèle d'optimisation de la planification (Scheduler) | 94 |
| 4.3.2.1 | Formulation du problème | 94 |
| 4.3.2.2 | Algorithme d'optimisation | 95 |
| 4.3.2.3 | Métriques d'optimisation | 95 |
| 4.4 | Phase 5 : Évaluation (Evaluation) | 95 |
| 4.4.1 | Évaluation du modèle de prédiction | 95 |
| 4.4.1.1 | Métriques de performance | 95 |
| 4.4.1.2 | Analyse des erreurs | 95 |
| 4.4.1.3 | Interprétabilité du modèle avec SHAP | 96 |
| 4.4.1.4 | Validation croisée temporelle | 97 |
| 4.4.1.5 | Courbes d'apprentissage | 98 |
| 4.4.1.6 | Courbes de validation | 98 |
| 4.4.1.7 | Matrice de comparaison des modèles | 99 |
| 4.4.1.8 | Tests de significativité statistique | 99 |
| 4.4.2 | Évaluation du modèle d'optimisation | 100 |
| 4.4.2.1 | Métriques d'optimisation | 100 |
| 4.4.2.2 | Analyse de robustesse | 100 |
| 4.5 | Intégration et déploiement | 100 |
| 4.5.1 | Architecture du système | 100 |
| 4.5.2 | Pipeline de production | 100 |
| 4.5.2.1 | Entraînement automatique | 100 |
| 4.5.2.2 | Serving en production | 100 |
| 4.5.3 | Gestion des modèles | 101 |
| 4.5.3.1 | Versioning | 101 |
| 4.5.3.2 | Monitoring | 101 |
| 4.6 | Validation métier | 101 |
| 4.6.1 | Tests avec les utilisateurs | 101 |
| 4.6.1.1 | Protocole de test | 101 |
| 4.6.1.2 | Métriques de satisfaction | 101 |
| 4.6.2 | Impact métier mesuré | 102 |
| 4.6.2.1 | KPIs opérationnels | 102 |
| 4.6.2.2 | ROI du projet | 102 |
| 4.7 | Synthèse et perspectives | 102 |
| 4.7.1 | Bilan de la modélisation | 102 |
| 4.7.2 | Limitations identifiées | 102 |
| 4.7.3 | Améliorations futures | 103 |
| 5 | Agile 4-month delivery plan | 104 |
| 5.1 | Introduction | 104 |
| 5.2 | Architecture Agile du Projet | 104 |
| 5.2.1 | Équipe de Développement | 104 |
| 5.2.2 | Outils et Technologies | 105 |
| 5.3 | Roadmap de Développement | 105 |
| 5.3.1 | Phase 1 : Fondations (Sprints 1-4) | 105 |
| 5.3.2 | Phase 2 : Développement Core (Sprints 5-8) | 105 |

| | | |
|---------|--|-----|
| 5.3.3 | Phase 3 : Optimisation et Production (Sprints 9-12) | 105 |
| 5.4 | Détail des Sprints | 106 |
| 5.4.1 | Sprint 1 : Analyse et Architecture | 106 |
| 5.4.1.1 | User Stories | 106 |
| 5.4.1.2 | Tâches Techniques | 106 |
| 5.4.1.3 | Livrables | 106 |
| 5.4.2 | Sprint 2 : Pipeline de Données | 106 |
| 5.4.2.1 | User Stories | 106 |
| 5.4.2.2 | Tâches Techniques | 107 |
| 5.4.2.3 | Livrables | 107 |
| 5.4.3 | Sprint 3 : Modèles de Machine Learning | 107 |
| 5.4.3.1 | User Stories | 107 |
| 5.4.3.2 | Tâches Techniques | 107 |
| 5.4.3.3 | Livrables | 107 |
| 5.4.4 | Sprint 4 : Pipeline MLOps | 108 |
| 5.4.4.1 | User Stories | 108 |
| 5.4.4.2 | Tâches Techniques | 108 |
| 5.4.4.3 | Livrables | 108 |
| 5.5 | Métriques et KPIs de Succès | 108 |
| 5.5.1 | Métriques Techniques | 108 |
| 5.5.2 | Métriques Métier | 109 |
| 5.5.3 | Métriques de Projet | 109 |
| 5.6 | Gestion des Risques | 109 |
| 5.6.1 | Framework de gestion des risques | 109 |
| 5.6.2 | Registre complet des risques | 110 |
| 5.6.3 | Matrice de criticité | 110 |
| 5.6.4 | Plans de réponse détaillés | 111 |
| 5.6.4.1 | Risque critique 1 : Performance modèles insuffisante | 111 |
| 5.6.4.2 | Risque critique 2 : Intégration avec G.Pro | 111 |
| 5.6.4.3 | Risque critique 3 : Adoption utilisateurs faible | 112 |
| 5.6.5 | Monitoring et reporting | 112 |
| 5.7 | Plan de Communication | 112 |
| 5.7.1 | Stakeholders | 112 |
| 5.7.2 | Cadence de Communication | 113 |
| 5.8 | Plan de Formation et Adoption | 113 |
| 5.8.1 | Formation des Utilisateurs | 113 |
| 5.8.2 | Stratégie d'Adoption | 113 |
| 5.9 | Plan de Maintenance et Évolution | 113 |
| 5.9.1 | Maintenance Préventive | 113 |
| 5.9.2 | Évolution du Système | 114 |
| 5.10 | Budget et Ressources | 114 |
| 5.10.1 | Budget Estimé | 114 |
| 5.10.2 | ROI Attendu | 114 |
| 5.11 | Conclusion | 114 |

| | | |
|----------|--|------------|
| 6 | Service IA and dashboard | 115 |
| 6.1 | Introduction | 115 |
| 6.2 | Architecture des Services IA | 115 |
| 6.2.1 | Vue d'ensemble de l'architecture | 115 |
| 6.2.2 | Composants principaux | 116 |
| 6.2.2.1 | Service de Prédiction | 116 |
| 6.2.2.2 | Service d'Ordonnancement | 116 |
| 6.2.2.3 | Service Analytics | 117 |
| 6.3 | Spécifications des APIs | 117 |
| 6.3.1 | API de Prédiction | 117 |
| 6.3.1.1 | Endpoint de prédiction simple | 117 |
| 6.3.1.2 | Endpoint de prédiction en lot | 118 |
| 6.3.2 | API d'Ordonnancement | 118 |
| 6.3.2.1 | Endpoint d'optimisation de planification | 118 |
| 6.3.3 | API Analytics | 119 |
| 6.3.3.1 | Endpoint de calcul des KPIs | 119 |
| 6.4 | Interface Utilisateur | 120 |
| 6.4.1 | Architecture du Dashboard React | 120 |
| 6.4.1.1 | Structure des composants | 120 |
| 6.4.1.2 | Gestion d'état | 121 |
| 6.4.2 | Pages principales | 121 |
| 6.4.2.1 | Dashboard Principal | 121 |
| 6.4.2.2 | Page de Planification | 122 |
| 6.4.2.3 | Page Analytics | 123 |
| 6.4.2.4 | Simulateur What-If | 123 |
| 6.5 | Intégration des Modèles ML | 125 |
| 6.5.1 | Service de Prédiction | 125 |
| 6.5.1.1 | Chargement des modèles | 125 |
| 6.5.1.2 | Préprocessing des données | 125 |
| 6.5.2 | Service d'Ordonnancement | 126 |
| 6.5.2.1 | Modélisation des contraintes | 126 |
| 6.6 | Sécurité et Authentification | 127 |
| 6.6.1 | Système d'authentification | 127 |
| 6.6.1.1 | Génération des tokens | 127 |
| 6.6.1.2 | Vérification des tokens | 127 |
| 6.6.2 | Contrôle d'accès basé sur les rôles (RBAC) | 127 |
| 6.7 | Monitoring et Observabilité | 128 |
| 6.7.1 | Métriques de performance | 128 |
| 6.7.1.1 | Métriques des modèles | 128 |
| 6.7.1.2 | Métriques des services | 128 |
| 6.7.2 | Alertes et notifications | 128 |
| 6.8 | Documentation Technique | 129 |
| 6.8.1 | Documentation API | 129 |
| 6.8.2 | Guide de déploiement | 129 |
| 6.8.2.1 | Configuration Docker | 129 |
| 6.8.2.2 | Docker Compose | 129 |

| | | |
|----------|---|------------|
| 6.8.3 | Tests et validation | 130 |
| 6.8.3.1 | Tests unitaires | 130 |
| 6.8.3.2 | Tests d'intégration | 130 |
| 6.9 | Performance et Optimisation | 131 |
| 6.9.1 | Optimisations des modèles | 131 |
| 6.9.1.1 | Cache des prédictions | 131 |
| 6.9.1.2 | Prédiction en lot | 131 |
| 6.9.2 | Optimisations de l'ordonnancement | 132 |
| 6.9.2.1 | Parallélisation | 132 |
| 6.10 | Évolutivité et Maintenance | 133 |
| 6.10.1 | Architecture évolutive | 133 |
| 6.10.1.1 | Microservices | 133 |
| 6.10.1.2 | API Gateway | 133 |
| 6.10.2 | Maintenance des modèles | 133 |
| 6.10.2.1 | Monitoring de la dérive | 133 |
| 6.10.2.2 | Réentraînement automatique | 133 |
| 6.11 | Conclusion | 134 |
| | Conclusion générale | 135 |
| | Webographie | 139 |

Liste des figures

| | | |
|------|--|----|
| 1.1 | logo d'entreprise | 5 |
| 1.2 | Siège de l'entreprise Bacsport | 5 |
| 1.3 | Clients de BACOVET | 5 |
| 1.4 | Certifications de BACOVET | 6 |
| 1.5 | organigramme de BACOVET | 6 |
| 1.6 | <i>Diagramme de séquence: "les procédure global de production chez BACOVET "</i> | 7 |
| 1.7 | <i>Diagramme de séquence: "Processus détaillé de l'atelier de coupe "</i> | 8 |
| 1.8 | <i>Zone matelassage</i> | 9 |
| 1.9 | coupe automatisée | 10 |
| 1.10 | Machine Sérigraphie | 10 |
| 1.11 | Contrôle Qualité | 11 |
| 1.12 | Atelier de confection | 12 |
| 1.13 | le projet photovoltaïque | 13 |
| 2.1 | Formulaire Google Forms | 19 |
| 2.2 | Diagramme circulaire (Pie chart) | 19 |
| 2.3 | Histogramme / Diagramme en barres (Bar chart) | 19 |
| 2.4 | Détail des résultats par critère | 20 |
| 2.5 | tableau complet des critères, scores et niveaux | 20 |
| 2.6 | <i>Tableau récapitulatif des scores par domaine</i> | 21 |
| 2.7 | Diagramme de Radar | 21 |
| 2.8 | Carte mentale | 24 |
| 2.9 | Fiche d'enregistrement (modèle Rim) | 27 |
| 2.10 | Fiche d'enregistrement (modèle Leotard 500g pink) | 27 |
| 2.11 | <i>Diagramme d'Ishikawa : Analyse des causes racines des retards dans l'atelier de coupe</i> | 33 |
| 2.12 | Diagramme de flux de l'algorithme de gestion des tables de matelassage | 36 |
| 2.13 | Exemple de suivi du taux de disponibilité des tables sur 12 semaines | 38 |
| 3.1 | Processus CRISP-ML(Q) avec portes de qualite | 42 |
| 3.2 | Langage de programmation Python | 43 |
| 3.3 | Bibliothèques de manipulation de données - pandas et NumPy | 44 |
| 3.4 | Logo Python - Langage de programmation utilisé pour l'écosystème Machine Learning (Python 3.2.2) | 45 |

| | | |
|------|--|-----|
| 3.5 | Écosystème Data Science et Machine Learning utilisé dans le projet (pandas 2.0.3, NumPy 1.24.3, scikit-learn 1.3.0, XGBoost 1.7.6, matplotlib 3.7.2, seaborn 0.12.2) | 46 |
| 3.6 | Technologies backend et serveur ASGI (FastAPI 0.103.0, Pydantic 2.3.0, uvicorn 0.23.2) | 47 |
| 3.7 | Technologies frontend et communication API (React 18.2.0, Recharts 2.8.0, Axios 1.5.0) | 48 |
| 3.8 | Infrastructure DevOps et outils de déploiement (Docker 24.0, Git 2.41, pytest 7.4.0, PostgreSQL 15.3, OR-Tools 9.7) | 49 |
| 3.9 | Workflow de feature engineering | 72 |
| 3.10 | Architecture du pipeline de preparation des donnees | 78 |
| 3.11 | Framework d'assurance qualite CRISP-ML(Q) | 79 |
| 4.1 | Architecture du système ML intégré | 100 |
| 6.1 | Architecture des services IA | 116 |

Liste des tableaux

| | | |
|------|--|----|
| 2.1 | Exemple de statut temps réel | 17 |
| 2.2 | Critères constituant des leviers prioritaires d'amélioration | 20 |
| 2.3 | Résumé du fonctionnement du logiciel Divatex | 23 |
| 2.4 | Résumé du fonctionnement du logiciel G.Pro | 24 |
| 2.5 | Analyse des forces et faiblesses du système actuel | 26 |
| 2.6 | Paramètres techniques de l'atelier | 30 |
| 2.7 | Analyse des temps observés par activité | 31 |
| 2.8 | Analyse des ressources et limitations | 32 |
| 2.9 | Synthèse des causes par catégorie | 34 |
| 2.10 | Exemple de statut temps réel | 37 |
| 2.11 | Indicateurs de performance pour le suivi de l'amélioration | 38 |
| 3.1 | Bibliothèques Python pour la manipulation de données | 43 |
| 3.2 | Bibliothèques Python pour le Machine Learning | 44 |
| 3.3 | Bibliothèques Python pour la visualisation | 45 |
| 3.4 | Technologies backend et API | 46 |
| 3.5 | Technologies frontend | 47 |
| 3.6 | Outils d'optimisation | 48 |
| 3.7 | Outils d'infrastructure et DevOps | 49 |
| 3.8 | Stack technologique complète du projet | 50 |
| 3.9 | Business Model Canvas du système IA de planification | 53 |
| 3.10 | Objectifs opérationnels avec métriques de succès | 54 |
| 3.11 | Matrice pouvoir-intérêt des parties prenantes | 54 |
| 3.12 | Comparaison processus AS-IS vs TO-BE | 58 |
| 3.13 | Registre détaillé des risques métier | 58 |
| 3.14 | Critères de succès techniques | 60 |
| 3.15 | Critères de succès métier | 60 |
| 3.16 | Caractéristiques détaillées des sources de données | 65 |
| 3.17 | Dictionnaire de données - Variables principales | 66 |
| 3.18 | Description des variables principales | 66 |
| 3.19 | Analyse des valeurs manquantes par variable | 68 |
| 3.20 | Valeurs aberrantes détectées | 69 |
| 3.21 | Features finales pour modélisation | 76 |
| 3.22 | Métriques de qualité des données | 80 |
| 3.23 | Alertes de qualité des données | 80 |
| 3.24 | Critères de validation des modèles ML | 81 |
| 3.25 | Système d'alertes de monitoring | 84 |

| | | |
|-----|---|-----|
| 4.1 | Comparaison des algorithmes de régression | 90 |
| 4.2 | Hyperparamètres optimisés de XGBoost | 92 |
| 4.3 | Performance du modèle de prédiction | 95 |
| 4.4 | Importance des features (SHAP values moyennes) | 96 |
| 4.5 | Performance en fonction de la taille du dataset | 98 |
| 4.6 | Comparaison complète des modèles de régression | 99 |
| 4.7 | Performance du modèle d'optimisation | 100 |
| 4.8 | Impact métier mesuré | 102 |
| | | |
| 5.1 | Roadmap Phase 1 - Fondations | 105 |
| 5.2 | Roadmap Phase 2 - Développement Core | 105 |
| 5.3 | Roadmap Phase 3 - Optimisation et Production | 105 |
| 5.4 | Registre détaillé des risques projet | 110 |
| 5.5 | Matrice probabilité-impact des risques | 110 |
| 5.6 | Répartition du Budget | 114 |



INTRODUCTION RALE

GÉNÉ-

Dans un contexte économique mondialisé où la compétitivité industrielle repose de manière croissante sur la capacité des organisations à exploiter efficacement les données et à intégrer des technologies de rupture, la transformation digitale s'impose comme un levier stratégique incontournable d'amélioration de la performance opérationnelle. Le secteur textile tunisien, secteur économique majeur confronté à des enjeux de compétitivité internationale, illustre cette nécessité de modernisation. L'émergence des concepts d'usine intelligente, d'automatisation avancée des processus de production et d'analyse prédictive, s'inscrit pleinement dans le paradigme de l'Industrie 4.0, apportant des réponses structurées aux problématiques contemporaines d'efficacité opérationnelle, de traçabilité des processus et de réactivité organisationnelle.

Le présent projet de fin d'études, réalisé au sein de l'entreprise BACOVET, filiale du groupe BACOSPORT, s'inscrit dans cette dynamique de transformation digitale orientée Industrie 4.0. L'objectif principal de ce travail consiste à concevoir, développer et mettre en œuvre une solution digitale intelligente reposant sur les principes de l'Industrie 4.0, afin d'améliorer significativement les performances de planification, de suivi opérationnel et de pilotage décisionnel des activités au sein de l'atelier de coupe textile.

Ce travail de recherche appliquée s'articule autour d'une démarche méthodologique rigoureuse et structurée, fondée sur la méthode DMAIC (*Define, Measure, Analyze, Improve, Control*), issue du référentiel Lean Six Sigma. Cette approche systématique permet d'identifier de manière objective les dysfonctionnements structurels et organisationnels existants, d'analyser leurs causes racines selon une démarche scientifique, et de proposer des solutions innovantes à fort impact opérationnel, fondées sur l'intelligence artificielle et les technologies de l'information.

Ce rapport de recherche est structuré en six chapitres complémentaires, chacun apportant une contribution spécifique à la compréhension et à la résolution de la problématique étudiée. Le **Chapitre 1** présente le cadre organisationnel et industriel de l'entreprise d'accueil, ainsi qu'une analyse critique de l'existant permettant de formuler précisément la problématique à résoudre. Le **Chapitre 2** détaille la démarche d'analyse méthodologique employée, incluant un audit de maturité digitale basé sur le référentiel IMPULS et l'application rigoureuse de la méthodologie DMAIC. Le **Chapitre 3** expose la méthodologie CRISP-ML(Q) appliquée aux phases de compréhension métier, d'analyse des données et de préparation des données pour la modélisation. Le **Chapitre 4** présente le développement des modèles de machine learning, leur évaluation approfondie et leur intégration dans un pipeline de production (MLOps). Le **Chapitre 5** détaille le plan de livraison agile structuré en sprints

de développement, permettant une mise en œuvre progressive et adaptative de la solution. Enfin, le **Chapitre 6** décrit l'architecture technique complète des services d'intelligence artificielle, l'interface utilisateur et le système de tableau de bord opérationnel, assurant ainsi la pérennité des gains obtenus et l'exploitation optimale des capacités prédictives du système.

Chapitre 1

Cadre du projet et étude de l'existant

1.1 Introduction

Ce premier chapitre établit le cadre contextuel et organisationnel de la recherche en présentant de manière systématique l'entreprise d'accueil BACOVET, filiale du groupe BACOSPORT et acteur significatif de l'industrie textile tunisienne. L'analyse s'attache à mettre en évidence les caractéristiques organisationnelles, le positionnement stratégique au sein de la chaîne de valeur textile, ainsi que l'orientation de l'entreprise vers l'innovation technologique.

Dans un second temps, le projet est contextualisé dans le paradigme de l'Industrie 4.0, permettant d'identifier de manière structurée les défis opérationnels liés à la planification et au suivi de production. Une étude critique et approfondie de l'existant, fondée sur l'observation empirique et l'analyse documentaire, permettra de formuler précisément la problématique à résoudre.

Enfin, une introduction structurée de la solution intelligente proposée et de la méthodologie de recherche adoptée clôturera ce chapitre, établissant ainsi les fondements théoriques et pratiques du travail de recherche entrepris.

1.2 Présentation de l'entreprise d'accueil : Baco-vet

1.2.1 Historique et identité

Le groupe BACOSPORT, fondé en 1967, constitue aujourd'hui un acteur majeur et structurant de l'industrie textile tunisienne, spécialisé dans la confection de vêtements sportswear, de sous-vêtements, de pyjamas et de maillots de bain. Dans l'écosystème organisationnel du groupe, BACOVET, filiale stratégique implantée à Boumerdes, occupe une position clé dans la chaîne de valeur en assurant les opérations critiques de coupe industrielle, de préparation des tissus, de sérigraphie, de contrôle qualité et de logistique de transfert vers l'atelier de confection. L'expertise technique reconnue et la rigueur organisationnelle déployée par BACOVET contri-

buent de manière substantielle à la compétitivité internationale du groupe et à sa capacité à répondre aux exigences qualitatives et temporelles des marchés export.



Figure 1.1: logo d'entreprise



Figure 1.2: Siège de l'entreprise Bacosport

1.2.2 Partenariats stratégiques et présence mondiale

Le succès du groupe Bacosport et de sa filiale Bacovet repose sur une collaboration étroite avec de grandes marques internationales du secteur textile. Grâce à son savoir-faire technique, sa flexibilité et sa capacité à répondre à des exigences de qualité élevées, Bacovet entretient des partenariats durables avec des enseignes renommées à travers l'Europe et le bassin méditerranéen. Parmi ses principaux clients figurent des marques telles que Décathlon, La Redoute, Damart, Sunflair, DD, Romy Aim, et Calao. Ces collaborations stratégiques témoignent de la confiance des donneurs d'ordre internationaux et renforcent la position de Bacovet comme un acteur de référence dans le textile tunisien à vocation exportatrice.

En s'inscrivant dans des chaînes d'approvisionnement mondiales, Bacovet adopte des standards de qualité et de traçabilité conformes aux attentes des marchés européens. Cette ouverture internationale pousse également l'entreprise à investir dans la digitalisation et dans des solutions innovantes, pour maintenir un niveau de performance concurrentiel.



Figure 1.3: Clients de BACOVET

1.2.3 Certifications de Bacovet

Bacovet s'engage à garantir la qualité et la sécurité de ses produits à travers le respect de normes internationales reconnues. L'entreprise est certifiée selon la norme ISO 9001 : 2015, qui atteste de l'efficacité de son système de management de la qualité et de son orientation vers l'amélioration continue.

Dans le cadre de sa responsabilité sociétale et environnementale, Bacovet applique également les standards OEKO-TEX® Standard 100, qui garantissent que

les tissus utilisés sont exempts de substances nocives et répondent aux exigences de sécurité pour la santé humaine.

Ces certifications renforcent la crédibilité de Bacovet auprès de ses partenaires et confirment son engagement envers la qualité, la durabilité et la conformité aux normes internationales.



Figure 1.4: Certifications de BACO-VET



1.3 Organisation et organigramme

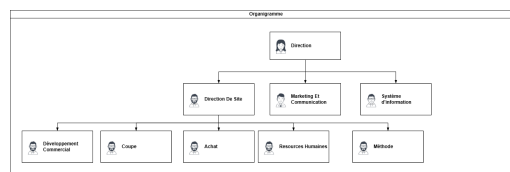


Figure 1.5: organigramme de BACOVET

1.4 Processus global chez BACOVET

Chez BACOVET, la chaîne de production textile est structurée en plusieurs étapes successives assurant une traçabilité, une qualité constante et un respect des délais. La figure 1.6 présente les processus global de production

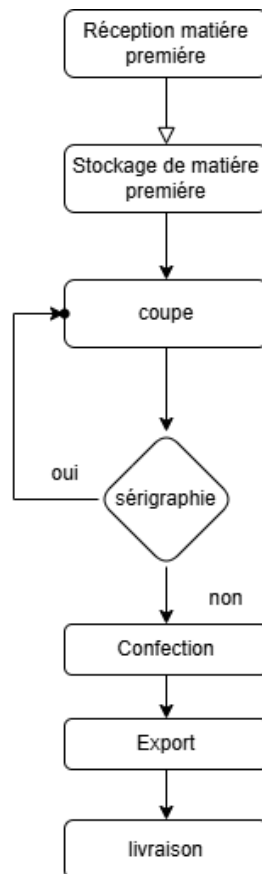


Figure 1.6: *Diagramme de séquence: "les procédure global de production chez BACOVET "*

1.4.1 Processus détaillé de l'atelier de coupe

L'atelier de coupe joue un rôle essentiel dans la production textile chez BACOVET. C'est à ce niveau que les rouleaux de tissu, qui représentent la matière la plus chère, sont découpés en pièces précises. Une bonne coupe permet de limiter les pertes de tissu et de garantir que toutes les pièces ont les bonnes dimensions. Cet atelier se situe entre l'approvisionnement des matières et l'assemblage final. Une gestion efficace de cette étape aide à respecter les délais, à bien utiliser les ressources, et à assurer la qualité des produits. Le respect des procédures (du marquage à la vérification finale) permet de réduire les erreurs et d'améliorer la performance globale. La figure 1.8 présente le processus détaillé de l'atelier de coupe.

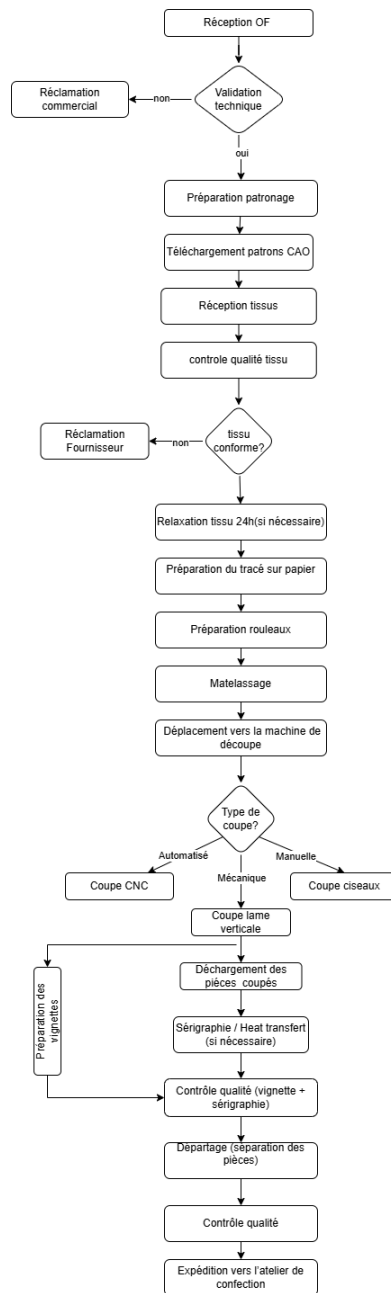


Figure 1.7: *Diagramme de séquence: "Processus détaillé de l'atelier de coupe "*

1. Réception des ordres de fabrication (OF)

L'atelier reçoit des ordres de fabrication (OF) qui indiquent les modèles à produire, les quantités et les tissus à utiliser. Ces ordres sont envoyés depuis la planification. On vérifie que toutes les informations sont correctes et que les rouleaux de tissu sont disponibles. Ensuite, le travail est réparti entre les équipes. Suivi : temps entre la réception de l'OF et le début de la coupe, vérification des données reçues.

Outils : lecteurs de codes-barres pour scanner les rouleaux.

2. Relaxation du tissu (si nécessaire)

Certains tissus doivent « se détendre » avant d'être découpés, pour éviter qu'ils ne rétrécissent plus tard. Cette étape se fait en laissant reposer le tissu à l'air libre ou à la vapeur pendant 24 à 72 heures.

Suivi : durée de relaxation, taux de retrait mesuré après repos.

Outils : capteurs de température et d'humidité dans la zone de repos.

3. Préparation des rouleaux et découpe du papier

Avant la coupe, les rouleaux de tissu sont préparés. On vérifie leur longueur, leur largeur et leur couleur. On coupe le papier de base (papier matelas) qui sera placé sous le tissu pour la coupe. Suivi : temps de préparation, gaspillage de papier, précision des mesures.

4. Matelassage

Le tissu est empilé en plusieurs couches sur la table de coupe. Le nombre de plis est défini selon l'épaisseur du tissu et le volume de production.



Figure 1.8: *Zone matelassage*

5. Placement et marquage

L'étape de placement et de marquage est essentielle pour bien préparer la coupe. Le placement consiste à organiser les patrons sur le tissu de façon à utiliser le moins de matière possible, tout en respectant le sens du tissu, le droit fil et les motifs (comme les rayures). Cela permet d'éviter le gaspillage et d'assurer un bon rendu final. Ensuite, le marquage sert à tracer les contours des pièces, à indiquer les repères nécessaires au montage (comme les crans ou les emplacements de poches), et à numérotter les pièces pour faciliter leur suivi. Une vérification finale est faite avant la découpe pour s'assurer que tout est bien positionné.

6. coupe

Avant de commencer la coupe, il faut préparer tous les outils nécessaires. Cela comprend la vérification de l'affûtage des lames, le bon fonctionnement des

machines, ainsi que la préparation des accessoires comme les pinces, les règles ou les équerres. La coupe peut ensuite se faire de différentes manières selon le type de production. Pour les petites séries, on utilise souvent la coupe manuelle avec des ciseaux, en suivant les tracés avec précision, surtout dans les zones courbes, tout en respectant les tolérances. Pour les plus grandes séries, la coupe mécanique est préférée, avec des machines à lame verticale. Il faut alors régler la hauteur de coupe selon l'épaisseur du matelas et surveiller la qualité du travail tout au long du processus. Enfin, la coupe automatisée permet une grande précision grâce à la programmation des machines. Les données de placement sont chargées, et la coupe se fait automatiquement avec un contrôle qualité en temps réel.



Figure 1.9: coupe automatisée



7. Déchargement des Pièces Coupées

Les pièces découpées sont triées et empilées avec soin. Cette étape requiert une vigilance particulière afin d'éviter les mélanges ou les endommagements

8. Sérigraphie / Heat Transfert (si nécessaire)

Certaines pièces nécessitent un marquage par sérigraphie ou transfert thermique. Cette étape permet l'identification ou la décoration des produits. Des capteurs contrôlent la température et la pression appliquées.



Figure 1.10: Machine Sérigraphie

9. Préparation et Contrôle des Vignettes

Des vignettes contenant les informations essentielles (taille, modèle, OF) sont générées et associées aux pièces. Leur exactitude est cruciale pour le bon déroulement des étapes suivantes. KPI : Taux d'exactitude des données.

10. Départage

Le départage consiste à trier les pièces selon les tailles et les modèles. Il s'agit d'une opération de précision qui facilite le travail de l'atelier de confection

11. Contrôle Qualité Final

Avant expédition, chaque lot est inspecté pour valider sa conformité. Cette étape est cruciale pour garantir un niveau de qualité élevé



Figure 1.11: Contrôle Qualité

12. Expédition vers l'Atelier de Confection

Les pièces sont conditionnées puis transférées à l'atelier de confection. Cette étape est suivie informatiquement via G.Pro, assurant une traçabilité complète.



Figure 1.12: Atelier de confection

1.4.2 Durabilité et responsabilité environnementale

Consciente des enjeux climatiques et environnementaux actuels, Bacosport, et par extension sa filiale Bacovet, s'inscrivent dans une démarche de production durable et responsable. L'entreprise considère la préservation de l'environnement comme un pilier fondamental de sa stratégie de développement industriel.

Dans cette optique, plusieurs initiatives écologiques majeures ont été mises en œuvre, notamment le projet photovoltaïque visant à transformer les sites de production en véritables sources d'énergie renouvelable. Cette transition énergétique permet non seulement de réduire significativement l'empreinte carbone de l'entreprise, mais également de renforcer son indépendance énergétique.

À ce jour, plus de 300 panneaux solaires ont été installés sur le site de Jammel, entraînant une réduction de 60

L'ensemble de ces initiatives s'inscrit dans un engagement global de réduction de l'impact environnemental, tout en maintenant un haut niveau de qualité et de performance industrielle. Par ce positionnement, Bacovet démontre qu'il est possible d'allier efficacité économique, innovation technologique et durabilité environnementale, en cohérence avec les objectifs de l'Industrie 4.0 verte.



Figure 1.13: le projet photovoltaïque

1.5 Cadre du projet

Dans un environnement industriel caractérisé par une intensification de la concurrence internationale et des exigences croissantes en matière de qualité, de réactivité, de traçabilité et d'optimisation des coûts, le secteur textile tunisien est confronté à l'impératif d'une évolution rapide et structurée vers des modes de production plus performants et plus flexibles.

Au sein de l'écosystème organisationnel du groupe BACOSPORT, la société BACOVET occupe une position stratégique déterminante dans la chaîne de valeur textile. L'entreprise regroupe plusieurs activités complémentaires et interdépendantes, notamment la coupe industrielle, la sérigraphie, la confection et le contrôle qualité, contribuant ainsi de manière substantielle à la performance globale du groupe sur les marchés nationaux et internationaux.

Cependant, l'analyse critique de l'existant révèle que l'atelier de coupe, bien que structuré et performant sur le plan opérationnel, présente des limitations structurelles significatives liées à l'absence d'un système digitalisé intégré d'ordonnancement, de suivi et de contrôle en temps réel. Cette lacune technologique et organisationnelle génère des pertes de temps opérationnelles non négligeables, une visibilité restreinte sur l'avancement des opérations, et une capacité limitée d'anticipation des retards ou de réactivité face aux imprévus de production.

Dans cette perspective, le présent projet de recherche s'inscrit dans une démarche structurée de transformation digitale ciblée, visant à intégrer de manière progressive et cohérente les principes fondamentaux de l'Industrie 4.0 au sein de l'atelier de coupe de BACOVET.

L'objectif principal de cette recherche consiste à concevoir, développer et mettre en place une solution numérique intelligente, fondée sur l'intelligence artificielle et les technologies de l'information, permettant de réaliser les objectifs opérationnels suivants :

- Assurer un suivi en temps réel et une traçabilité complète des opérations, notamment au niveau des phases critiques de matelassage et de coupe
- Optimiser la gestion des plannings de production et l'allocation des ressources disponibles (machines, opérateurs)
- Garantir une traçabilité exhaustive des opérations et des flux de production

- Améliorer significativement les performances opérationnelles de l'atelier, mesurées par des indicateurs clés de performance (KPIs) quantifiables

1.6 Conclusion du chapitre

Ce chapitre a établi le cadre contextuel et organisationnel de la recherche en présentant l'entreprise d'accueil BACOVET et son positionnement stratégique dans l'écosystème textile tunisien. L'analyse critique de l'existant a permis d'identifier les limitations structurelles de l'atelier de coupe, notamment l'absence d'un système digitalisé intégré.

Ce projet de transformation digitale constitue une opportunité stratégique majeure pour BACOVET de renforcer son agilité industrielle et sa compétitivité opérationnelle, tout en posant les fondations méthodologiques et technologiques d'une digitalisation progressive et évolutive des autres maillons de la chaîne de production. Les résultats attendus de cette recherche contribueront à la fois au renforcement des capacités opérationnelles de l'entreprise et à l'enrichissement des connaissances académiques sur l'application des principes de l'Industrie 4.0 au secteur textile tunisien.

Chapitre 2

Mener une action d'amélioration d'un atelier de travail avec Audit, les outils lean 4.0

2.1 Introduction

Ce deuxième chapitre présente de manière structurée et systématique la démarche méthodologique employée pour analyser et améliorer les performances de l'atelier de coupe. L'architecture de ce chapitre s'organise en deux parties complémentaires et interdépendantes.

La première partie expose le **diagnostic de maturité digitale** de l'atelier de coupe réalisé au travers d'un **audit approfondi** et structuré, fondé sur le référentiel international **IMPULS** (Industrie 4.0 Maturity Index) spécifiquement conçu pour l'évaluation de la maturité digitale industrielle. Cet audit méthodologique permet d'évaluer de manière objective et quantifiée le niveau de digitalisation actuel de l'atelier, et d'identifier de manière priorisée les axes stratégiques d'amélioration en cohérence avec les standards de l'Industrie 4.0.

La seconde partie détaille l'application rigoureuse et systématique de la **méthodologie DMAIC** (*Define, Measure, Analyze, Improve, Control*), issue du référentiel **Lean Six Sigma**, mise en œuvre dans le cadre d'une **action d'amélioration Lean 4.0** visant à identifier de manière structurée les dysfonctionnements opérationnels de l'atelier de coupe, à quantifier leurs impacts, à analyser leurs causes racines, et à proposer des solutions d'amélioration fondées sur des données objectives.

L'objectif principal de cette démarche méthodologique combinée est d'établir une base solide, rigoureuse et fondée sur des preuves empiriques pour la conception et le développement de **solutions digitales intelligentes**, en s'appuyant à la fois sur une **évaluation objective du niveau de maturité digitale** et sur une **analyse quantitative approfondie des performances actuelles** de l'atelier.

2.2 Partie I : Audit de maturité digitale

2.2.1 Introduction a l'audit IMPULS

Dans un contexte de transformation numérique croissante, il est essentiel d'**évaluer le niveau de maturité digitale** des différents services de production. L'audit de maturité digitale permet d'identifier les forces, les faiblesses et les opportunités d'évolution vers les standards de l'Industrie 4.0.

2.2.2 Objectifs de l'audit

- Évaluer le niveau d'intégration des technologies numériques dans l'atelier de coupe
- Identifier les axes d'amélioration en lien avec l'excellence opérationnelle et l'Industrie 4.0
- Préparer un plan d'action de transformation digitale adapté a la réalité terrain

2.2.3 Phase Innover : Propositions d'amélioration

2.2.3.1 Constat principal

L'analyse de la phase *Analyze* a révélé plusieurs dysfonctionnements limitant la performance de l'atelier de coupe, notamment une **saturation rapide des tables de matelassage** due a l'absence de système de planification et de suivi en temps réel.

Ces contraintes se traduisent par :

- Une **indisponibilité fréquente** des tables lors des pics d'activité ;
- Des **conflits de planification** entre opérateurs, dus a un manque de visibilité globale sur les ressources ;
- L'absence d'un **outil numérique** permettant de synchroniser le planning avec l'état réel des tables.

Ces constats soulignent la nécessité d'un **pilotage intelligent des ressources**, s'inscrivant pleinement dans la logique de l'Industrie 4.0, ou la digitalisation et l'automatisation des processus permettent de renforcer la réactivité et l'efficacité opérationnelle.

2.2.3.2 Proposition : Algorithme intelligent de gestion des tables de matelassage

Afin d'optimiser la planification et l'utilisation des ressources physiques de l'atelier, nous proposons un **algorithme de gestion dynamique** des tables de matelassage. Cet outil vise a offrir une **visualisation en temps réel** de la disponibilité des tables, tout en anticipant les conflits d'utilisation.

2.2.3.2.1 Principe de fonctionnement Chaque table est identifiée par un code unique (T1, T2, ..., T6). Le système compare en continu l'heure actuelle avec le planning de matelassage. Il met automatiquement à jour le statut de chaque table selon trois états possibles : *Occupée* (en cours d'utilisation), *Réservée* (prévue pour une opération à venir) et *Disponible* (libre pour un nouveau matelas).

2.2.3.2.2 Pseudocode de l'algorithme

Entrées :

Tables = {T1, T2, ..., T6}
 Planning (heure début, durée prévue)
 Heure actuelle = H

Pour chaque table T[i] :

Si (H >= heure début[i] ET H < heure fin prévue[i]) :
 Statut[T[i]] = "Occupée"
 Sinon si (H < heure début[i]) :
 Statut[T[i]] = "Réservée"
 Sinon :
 Statut[T[i]] = "Disponible"

Afficher le statut de chaque table

2.2.3.3 Application pratique

L'algorithme peut être déployé sous plusieurs formes :

- **Une feuille Excel automatisée**, avec macros et mise à jour minute par minute ;
- **Une interface web locale**, connectée aux données de production et accessible depuis le poste du chef d'atelier.

Ce système permet :

- de **visualiser en temps réel** la disponibilité de chaque table ;
- d'**éviter les conflits** ou chevauchements de planning ;
- d'**améliorer la fluidité** du flux de travail entre préparation, matelassage et coupe.

Tableau 2.1: Exemple de statut temps réel

| Table | Heure actuelle | Statut |
|-------|----------------|-------------------------|
| T1 | 09:00 | Occupée (jusqu'à 09:30) |
| T2 | 09:00 | Disponible |
| T3 | 09:00 | Occupée (jusqu'à 09:45) |

2.2.3.4 Bénéfices attendus

L'intégration de cet algorithme dans le système de gestion de production offre plusieurs avantages :

- **Réduction du temps d'attente** entre les opérations de coupe ;
- **Optimisation de l'utilisation des ressources existantes** sans investissement matériel supplémentaire ;
- **Amélioration de la coordination** entre opérateurs et planificateurs ;
- **Digitalisation partielle du pilotage de production**, contribuant à la transition vers une **usine connectée**.

Ainsi, cette solution constitue une première étape vers la **transformation numérique** de l'atelier, en s'inscrivant dans une démarche *Lean 4.0* conciliant **amélioration continue** et **technologies intelligentes**.

2.2.4 Méthodologie d'évaluation

2.2.4.1 Choix du questionnaire IMPULS

Le questionnaire **IMPULS** à « **Industrie 4.0 Readiness** [1] a été choisi pour sa pertinence dans l'analyse de la maturité digitale industrielle [2], [3]. Il couvre les thématiques suivantes :

- Efficacité des processus
- Automatisation, innovation et intégration numérique
- Gestion des performances et des flux
- Amélioration continue et flexibilité

2.2.4.2 Mode de diffusion

Le questionnaire a été administré en ligne via Google Forms, avec un accompagnement en présentiel pour aider les répondants à bien comprendre chaque question technique.

2.2.4.3 Cible de l'enquête

L'enquête a ciblé **11 personnes clés** de l'atelier de coupe, incluant :

- Chef d'atelier
- Agents d'ordonnancement/planification
- Opérateurs machine
- Responsables qualité coupe

- Responsable maintenance

Ces profils ont été choisis en fonction de leur connaissance du fonctionnement réel de l'atelier et de leur implication dans les processus numériques ou manuels actuels.

2.2.5 Déroutement de l'enquête

- **Préparation** : création du formulaire Google Forms (Figure 2.8), adaptation des questions a l'environnement Bacovet.
- **Lancement** : envoi du lien aux personnes concernées, accompagnement en face a face sur place.
- **Assistance** : chaque répondant a été guidé par l'auteur du projet pour expliciter les critères du questionnaire.
- **Collecte et consolidation** : les réponses ont été collectées automatiquement, puis exploitées pour l'analyse.(Les Figures 2.2;2.3)

Figure 2.1: Formulaire Google Forms

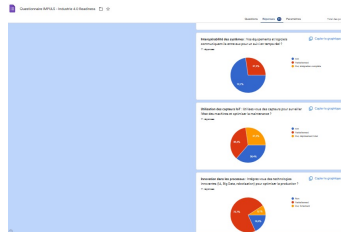


Figure 2.2: Diagramme circulaire (Pie chart)

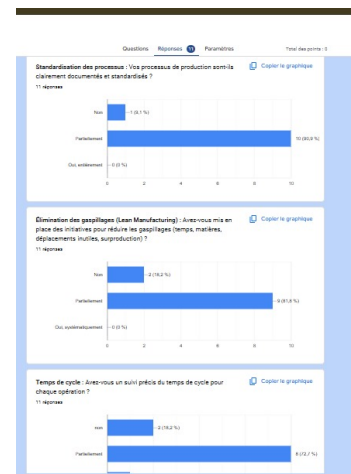


Figure 2.3: Histogramme / Diagramme en barres (Bar chart)

2.3 Résultats de l'évaluation de la maturité digitale

2.3.1 Méthodologie d'évaluation

Méthode: L'évaluation de la maturité digitale de l'atelier de coupe a été réalisée a l'aide du questionnaire **IMPULS**, un outil de diagnostic développé par l'Association Allemande de l'Industrie (VDMA) pour mesurer l'état d'avancement des entreprises dans l'adoption des principes de l'Industrie 4.0.

L'évaluation de la maturité digitale a été effectuée a l'aide d'un calculateur basé sur les retours de 11 répondants. Chaque critère a été noté selon trois niveaux d'implémentation : « Non », « Partiellement », et « Oui », permettant de générer un score global par critère .

2.3.2 Détail des scores par critère



Figure 2.4: Détail des résultats par critère

| | | | | | | | |
|--|-----------|---|---|---|-------|-----|--------------------------|
| Adaptation à l'industrie 4.0 | Domaine B | 4 | 5 | 2 | 11/11 | 41% | Partiellement implémenté |
| Visualisation des performances | Domaine C | 8 | 0 | 3 | 11/11 | 27% | Non implémenté |
| Gestion des flux physiques | Domaine C | 2 | 9 | 0 | 11/11 | 41% | Partiellement implémenté |
| Planification et ordonnancement | Domaine C | 7 | 3 | 1 | 11/11 | 23% | Non implémenté |
| Suivi de l'avancement | Domaine C | 8 | 1 | 2 | 11/11 | 23% | Non implémenté |
| Répartition des ressources | Domaine C | 4 | 7 | 0 | 11/11 | 32% | Partiellement implémenté |
| Culture d'amélioration continue (Kaizen) | Domaine D | 3 | 6 | 2 | 11/11 | 45% | Partiellement implémenté |
| Flexibilité de la production | Domaine D | 0 | 8 | 3 | 11/11 | 64% | Partiellement implémenté |
| Gestion des compétences | Domaine D | 3 | 6 | 0 | 11/11 | 36% | Partiellement implémenté |
| Innovation produit et service | Domaine D | 2 | 9 | 0 | 11/11 | 41% | Partiellement implémenté |

Figure 2.5: tableau complet des critères, scores et niveaux

L’audit a porté sur **17 critères** répartis dans **4 domaines** clés :

- A. Efficacité des processus
- B. Automatisation, innovation et intégration numérique
- C. Gestion des performances et des flux
- D. Amélioration continue et flexibilité

Le questionnaire a été administré auprès de **11 personnes** travaillant dans l’atelier de coupe. Le système de calcul automatique permet de convertir les réponses (« Non », « Partiellement », « Oui ») en un **score sur 100**, selon la formule suivante :

$$\text{Score} = \frac{\text{Partiellement} \times 0,5 + \text{Oui} \times 1}{11} \times 100$$

(2.1)

Le tableau détaillé met en évidence les critères les moins avancés :

| Critères prioritaires | Score | Niveau |
|---------------------------------|-------|----------------|
| Interopérabilité des systèmes | 14% | Non implémenté |
| Planification et ordonnancement | 23% | Non implémenté |
| Suivi de l'avancement | 23% | Non implémenté |
| Visualisation des performances | 27% | Non implémenté |

Tableau 2.2: Critères constituant des leviers prioritaires d’amélioration

Ces critères constituent des leviers prioritaires d’amélioration.

2.3.3 Résultats globaux

L'analyse des données a révélé un **score global de 41/100**, ce qui correspond a un **niveau de maturité « partiellement implémenté »**. Ce résultat indique que plusieurs pratiques et technologies de l'Industrie 4.0 sont amorcées, mais restent a un stade initial de mise en Åuvre ou ne sont pas encore déployées de manière homogène.

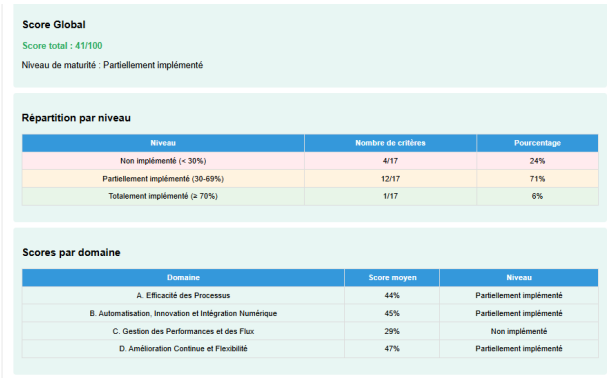


Figure 2.6: *Tableau récapitulatif des scores par domaine*

La **répartition par niveau** montre que :

- **24%** des critères sont **non implémentés** (<30%)
- **71%** sont **partiellement implémentés** (entre 30% et 69%)
- **6%** (1 critère) est **totalement implémenté** (>70%)

2.3.4 Analyse par domaine

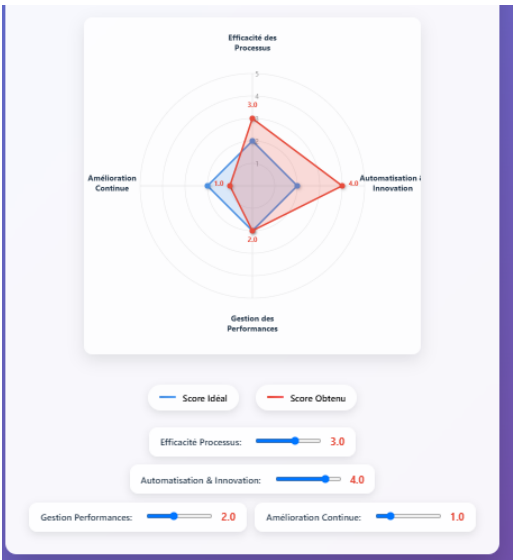


Figure 2.7: Diagramme de Radar

L'analyse par domaine met en lumière les disparités suivantes :

2.3.4.1 Domaine A : Efficacité des processus

- **Score moyen : 44%** â« Partiellement implémenté
- **Points faibles** : élimination des gaspillages (41%), temps de cycle (45%), standardisation (45%)
- Ces résultats révèlent un manque d'outils Lean et de pratiques systématisées d'optimisation.

2.3.4.2 Domaine B : Automatisation, innovation et intégration numérique

- **Score moyen : 45%** â« Partiellement implémenté
- Bien que l'automatisation soit amorcée (50%), des lacunes critiques sont observées sur l'**interopérabilité des systèmes** (14%), traduisant un cloisonnement entre les outils numériques.

2.3.4.3 Domaine C : Gestion des performances et des flux

Attention: Score moyen : 29% â« **Non implémenté**

C'est le domaine le plus faible. La **visualisation des performances** (27%), la **planification** (23%) et le **suivi de l'avancement** (23%) ne sont pas opérationnels. Cela empêche un pilotage efficace de la production.

2.3.4.4 Domaine D : Amélioration continue et flexibilité

- **Score moyen : 47%** â« Partiellement implémenté
- Ce domaine présente des initiatives prometteuses. Seul le critère relatif à l'**utilisation des capteurs IoT** atteint 77%, indiquant un potentiel élevé pour le suivi des opérations.

2.3.5 Conclusion des résultats

L'audit révèle que l'atelier de coupe se situe dans une **phase intermédiaire de transformation digitale**. Plusieurs outils sont présents, notamment au niveau des capteurs et de certains processus d'automatisation, mais leur intégration globale reste insuffisante. Le manque d'interopérabilité, de visibilité en temps réel, ainsi que l'absence de planification numérique constituent les principaux freins à une digitalisation avancée.

2.4 Partie II : Application de la méthodologie DMAIC

2.4.1 Introduction a la méthode DMAIC

Dans une logique d'optimisation de la performance industrielle et en cohérence avec les principes de l'Industrie 4.0, ce projet applique la méthodologie DMAIC (Define, Measure, Analyze, Improve, Control), issue du Lean Six Sigma. Cette démarche structurée de résolution de problèmes, fondée sur l'analyse factuelle et des mesures quantifiables, permet d'identifier les causes réelles des dysfonctionnements et de mettre en œuvre des solutions durables et mesurables. Dans le contexte de l'atelier de coupe textile, où subsistent des limites en matière de planification, de coordination et de traçabilité malgré la présence d'outils numériques, l'application du DMAIC vise à améliorer la gestion du temps, la maîtrise du planning et, plus globalement, la performance opérationnelle.

2.4.2 Phase 1 : Define (Définir)

2.4.2.1 Objectif de la phase Define

La phase Define constitue le fondement de la démarche DMAIC. Elle vise à **définir précisément le problème**, à identifier les parties prenantes, à établir le périmètre du projet et à fixer des objectifs mesurables d'amélioration. Cette phase garantit l'alignement de tous les acteurs sur une compréhension commune de la problématique et des enjeux.

2.4.2.2 Systèmes d'information utilisés dans l'atelier de coupe

Dans l'atelier de coupe Bacovet, la gestion opérationnelle repose sur deux logiciels principaux : **Divatex** et **G.Pro**, chacun couvrant une partie du processus de production.

-**Divatex** intervient en amont, au niveau du **stock et de la planification des matières**. Il assure la gestion des rouleaux de tissu, la réservation des matières premières, et le suivi des mouvements de consommation et de réception. Ce système permet également la création et la gestion des matériaux et accessoires utilisés pour les modèles.

Tableau 2.3: Résumé du fonctionnement du logiciel Divatex

| Élément | Description |
|-----------|---|
| Rôle | Gestion amont : stock de tissu, planification et mouvement des rouleaux |
| Entrées | Données des rouleaux (référence, quantité, emplacement), ordres de fabrication (OF), |
| Processus | Lancement des commandes, réservation des rouleaux, création des matériaux et access |
| Sorties | Mise à jour des stocks, statut matière des OF, historique des mouvements, rapports de |

-**G.Pro**, quant à lui, est opérationnel à partir de la zone de **départage** et jusqu'au **contrôle qualité confection**. Il assure la traçabilité des paquets grâce à un tag circulaire contenant les informations de l'ordre de fabrication (OF) et permet le suivi manuel de l'avancement des commandes post-coupe.

Tableau 2.4: Résumé du fonctionnement du logiciel G.Pro

| Alément | Description |
|-----------|---|
| Rôle | Gestion aval : traçabilité post-coupe et suivi des paquets |
| Entrées | OF et tags circulaires contenant les informations d'identification des paquets |
| Processus | Suivi de progression des paquets, détection via tag, contrôle qualité, validation des éta |
| Sorties | Statut des OF, temps de production (saisie manuelle), rapports de progression, histori |

Malgré leur rôle essentiel, ces deux systèmes fonctionnent de manière indépen-
dante, sans synchronisation automatique. Cette séparation génère une **zone grise**
entre les deux phases critiques de production : le **matelassage** et la **coupe**. Ni le
temps de matelassage ni le temps de coupe ne sont enregistrés ou centralisés auto-
matiquement. En conséquence, l'absence de liaison en temps réel entre la planifica-
tion matière (Divatex) et l'exécution opérationnelle (G.Pro) limite la **traçabilité**,
la **réactivité** ainsi que la capacité d'analyse des performances et de gestion des
réclamations.

Synthèse: La digitalisation actuelle assure la traçabilité matière et post-coupe,
mais reste **incomplète sur les opérations de matelassage et de coupe**. Cela
justifie l'intégration d'une solution intelligente permettant le suivi en temps réel
et l'interconnexion entre les outils existants, afin d'améliorer la performance et la
visibilité globale du processus.

2.4.2.3 Outil utilisé : Carte mentale

Pour synthétiser la situation actuelle, une carte mentale a été construite (Fi- gure
3.1). Elle présente :

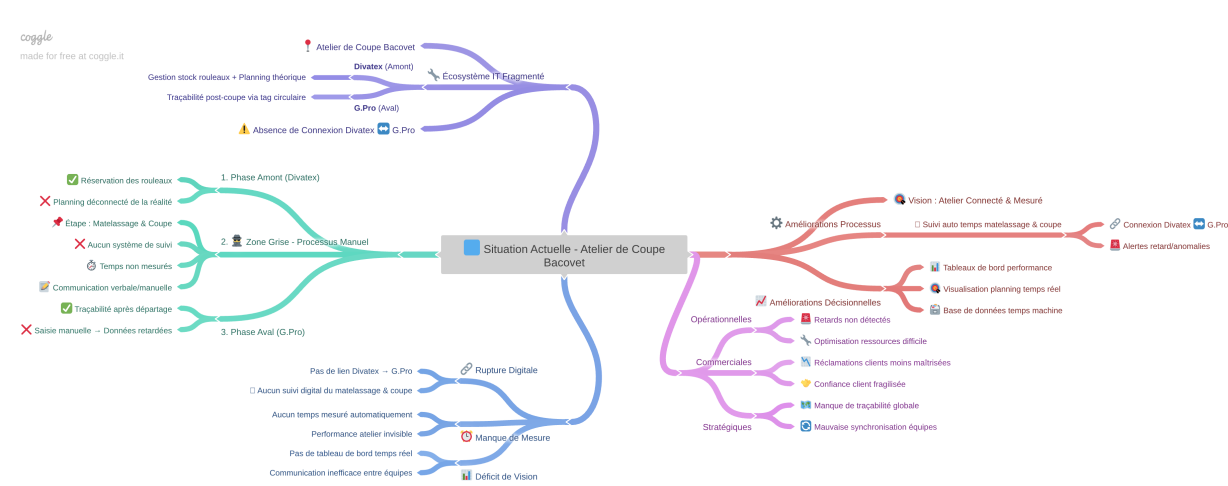


Figure 2.8: Carte mentale

1. Problème principal

- **Absence d'intégration numérique** entre les systèmes Divatex (amont) et G.Pro (aval)

- **Aucune mesure automatique** des temps de matelassage et de coupe
- **Pas de système d'alerte** pour détecter les retards ou anomalies en temps réel

2. Limites spécifiques de Divatex et G.Pro

- **Divatex** : Gestion des stocks et planning théorique, mais déconnecté de la réalité terrain
- **G.Pro** : Traçabilité post-couche via tag circulaire, mais saisie manuelle → données retardées
- **Aucune connexion** entre les deux systèmes → rupture digitale

3. Zone grise non couverte numériquement

- **Processus de matelassage et coupe** : Aucun suivi digital, communication verbale/manuelle, temps non mesurés

4. Conséquences opérationnelles

- Retards non détectés
- Mauvaise synchronisation entre les équipes
- Ressources mal exploitées
- Performance de l'atelier invisible
- Communication inefficace

5. Conséquences commerciales et stratégiques

- Réclamations clients moins maîtrisées
- Confiance client fragilisée
- Manque de traçabilité globale
- Difficulté à optimiser les ressources

6. Objectifs visés par l'amélioration

- **Mesure automatique** des temps de matelassage et coupe
- **Connexion Divatex ↔ G.Pro**
- **Système d'alerte** pour retards/anomalies
- **Tableaux de bord** de performance en temps réel
- **Base de données** des temps machine

2.4.2.4 Résumé des points clés identifiés

Le tableau 2.5 présente une synthèse des forces et faiblesses identifiées lors de cette phase de définition.

Tableau 2.5: Analyse des forces et faiblesses du système actuel

| Forces | Faiblesses |
|---|--|
| Existence de logiciels spécialisés (Divatex, G.Pro) | Pas de connexion entre Divatex et G.Pro |
| Système de traçabilité par tag post-coupe | Aucune traçabilité instantanée de la coupe |
| Engagement du personnel | Pas de mesure automatique des temps de maté- |
| | Absence de planning intégré et dynamique |
| | Communication manuelle entre équipes |
| | Retards ou anomalies détectés trop tardivement |

2.4.2.5 Conclusion de la phase "Définir"

La problématique centrale est **l'absence d'intégration entre les systèmes existants et le manque de visibilité en temps réel sur les opérations de matelassage et de coupe.**

2.4.3 Phase 2 : Measure (Mesurer)

Dans la démarche DMAIC, la phase **Mesurer** consiste à établir une connaissance factuelle du processus actuel. Elle vise à collecter des données *fiables, traçables et représentatives* sur les opérations clés, afin de quantifier les performances réelles et les perturbations. Cette étape permet de définir une **ligne de base (baseline)** des indicateurs, de valider le système de mesure et de préparer une analyse objective des causes dans la phase suivante.

2.4.3.1 Objectif de la phase de mesure

La phase de **mesure** établit une base de référence objective des performances actuelles. Elle vise à **quantifier les perturbations** de planification et à **mesurer les temps réels** des opérations clés (préparation, matelassage, coupe, départage), afin de disposer d'indicateurs fiables avant toute amélioration.

2.4.3.2 Méthodologie de mesure

Deux approches principales sont utilisées :

- Chronométrage direct via des fiches d'enregistrement (début, fin, durée) pour chaque étape : préparation, matelassage (automatique et manuel), coupe (robotisée et manuelle), départage et vignettage, comme illustré aux Figures 2.9 et 2.10.

| Tâche | Début | Fin | Durée (min) | Responsable | État |
|--|-----------------|--------|-------------|------------------------|----------|
| Préparation d'un titre de coupe | | | | CAO | Terminé |
| Préparation d'un tracé CP | | | | CAO | Terminé |
| Montage automatique(2) | 07:42 | 08:00 | 00:08 | robot | Terminé |
| | 09:20 | 09:35 | 00:15 | robot | Terminé |
| Travail du robot de coupe T1(1) | 09:00 | 09:27 | 00:27 | Robot avec 2 operateur | Terminé |
| Travail du robot de coupe T2(2) | 09:35 | 11:30 | 01:55 | Capitaine | Terminé |
| Chargement des pièces (robot → stock - déchargement) | 10:22:00 | 11:30 | 00:13 | Capitaine | Terminé |
| réalignement manuel T1(1) | 10:00 | 10:10 | 00:10 | Terminé | Terminé |
| réalignement manuel T2(2)(3) | 10:10 | 10:40 | 00:30 | Terminé | Terminé |
| réalignement manuel T3(3) | 10:40 | 10:45 | 00:05 | Terminé | Terminé |
| robot de coupe 2 | 10:00:00(20:00) | demain | | Opérateur | Terminé |
| | | | | Maintenance | En panne |
| | | | | Informaticien | En panne |

Figure 2.9: Fiche d'enregistrement (modèle Rim)

| Tâche | Début | Fin | Durée (min) | Responsable | État |
|--------------------------------------|-------|-------|-------------|------------------------|----------|
| Préparation d'un titre de coupe | 07:30 | 08:00 | 00:30 | CAO | Terminé |
| Préparation d'un tracé CP | 08:00 | 08:05 | 00:05 | CAO | Terminé |
| partie Automatique sur Robot Capitan | 08:05 | 08:20 | 00:15 | service informatique | Terminé |
| Montage automatique | 10:00 | 10:05 | 00:05 | robot | Terminé |
| Montage manuel T1(1)(2)(3) | 10:05 | 10:10 | 00:05 | Capitaine | Terminé |
| réalignement manuel T2(2)(3) | 10:10 | 10:15 | 00:05 | Capitaine | Terminé |
| Travail du robot de coupe | 10:15 | 10:20 | 00:05 | robot avec 2 operateur | Terminé |
| réalignement manuel T3(3) | 10:20 | 10:25 | 00:05 | Capitaine | Terminé |
| coupe double | 10:25 | 10:30 | 00:05 | robot | Terminé |
| robot de coupe 2 | 10:30 | 10:35 | 00:05 | robot | Terminé |
| | | | | Maintenance | En panne |
| | | | | Informaticien | En panne |

Figure 2.10: Fiche d'enregistrement (modèle Leotard 500g pink)

2.4.3.2.1 Analyse du déroulement de coupe (extrait documentaire) Afin de compléter la collecte terrain, une **analyse documentaire** a été réalisée à partir du fichier *analyse deroulement COUPE*. Cette source décrit pas-a-pas le déroulement d'articles en coupe et sert de support a la validation des séquences observées et des points de mesure.





















2.4.3.2.2 Aperçu (intégration directe du PDF) À défaut de captures déjà extraites, les pages clés du document sont intégrées ci-dessous directement depuis le PDF.

Analyse de déroulement (AD)

(coupe)

Sous-prosessus etudie :

Borne de debut et fin :

| Etapas de prosessus | | Etat | | | | | Valeurs | | | | Observations et commentaires Rapport temps operationnel | VA(MIN) | Actions | | | |
|---------------------|---|---|---|---|---|---|--------------|----------------|-----------|-------------|---|---------|---|----------|----------|---|
| | | operation | transport | Contrôle | attend | Stockage | Distance (M) | Quantité pièce | Nva/pièce | Temps (MIN) | | | Eliminer | combiner | Permuter | Ameliorer |
| | |  |  |  |  |  | | | | | | | | | | |
| 1 | Deplacement de limport a zone de stockage coupe | |  | | | | {12.3 ; 32} | | | 0.039 | | 0 |  | | | |
| 2 | STOCKAGE COUPE | | | | |  | | 4033 | 3.22 | 3,21 | | 0 |  | | | |
| 3 | Deplacement de coupe vers magasin import | |  | | | | {12.3 ; 32} | | 0.039 | 0.039 | | 0 |  | | | |
| 4 | Déroulement |  | | | | | | | 0.004 | 0.004 | | 0 | | | |  |
| 5 | Deplacement de magasin a coupe | |  | | | | {12.3 ; 32} | | 0.039 | 0.039 | | 0 |  | | | |
| 6 | Relaxation 24 h | | | | |  | | | 0.074 | 0.074 | STOCKAGE FORCE | 0 | | | | |
| 7 | Matlessage |  | | | | | | | | 0.202 | | 0.202 | | | | |
| 8 | Coupe |  | | | | | | | | 0.023 | | 0.023 | | | | |
| 9 | Prep vignettes |  | | | | | | | | 0.26 | | 0.26 | | | | |
| 10 | Departage |  | | | | | | | | 0.33 | | 0.33 | | | | |

2.4.3.2.3 Description synthétique du document Le document *analyse déroulement COUPE* présente, sous forme structurée et illustrée, le déroulement réel des opérations dans la salle de coupe. Il décrit successivement les étapes de préparation (CAO, vérification OF et matières), de matelassage (manuel et automatique), de coupe (robotisée et/ou manuelle), puis de départage et vignettage, en précisant pour chaque étape les acteurs impliqués, les supports utilisés, les temps observés, ainsi que les incidents/perturbations relevés (pannes, attentes, changements de priorité, relaxation tissu, etc.).

2.4.3.2.4 Objectifs de l'analyse documentaire

- Valider, par triangulation, les séquences opérationnelles observées sur le terrain et celles décrites par les procédures internes.
- Identifier les points de contrôle utiles à la mesure (horodatages, transitions d'étapes, variables influentes comme nombre de plis, laize, longueur).
- Mettre en évidence les sources récurrentes de variabilité et de perturbation impactant la planification (attentes tables, pannes robot, relaxation doublure, saisies manuelles).
- Définir une base de référence fiable pour la modélisation des temps et l'optimisation ultérieure (chapitres Analyse et Amélioration).

2.4.3.2.5 Principaux enseignements

- Les transitions entre matelassage et coupe constituent un point critique de synchronisation; l'absence de visibilité temps réel génère des attentes et des replanifications.
- Le temps de matelassage est fortement corrélé au nombre de plis et aux dimensions du matelas; la variabilité opérateur/machine demeure notable.
- Les perturbations majeures recensées (pannes robot, relaxation tissu, changements de priorité) expliquent une part significative des écarts planning.
- La traçabilité amont/aval est partielle: des données clés restent saisies manuellement et non consolidées.

2.4.3.2.6 Limites et précautions

- La documentation se concentre sur des cas représentatifs mais non exhaustifs; elle doit être complétée par des mesures systématiques et des séries temporelles plus longues.
- Certaines informations (délais d'attente, durées anormales) peuvent être sous-déclarées en l'absence d'instrumentation continue.

2.4.3.2.7 Conclusion opérationnelle Cette analyse documentaire consolide la compréhension du flux réel en salle de coupe et fournit des repères mesurables pour la phase **Analyser**. Elle justifie la mise en place d'une instrumentation de mesure plus fine, l'amélioration de la visualisation temps réel et l'usage d'un modèle prédictif des temps de matelassage/coupe pour soutenir la planification.

2.4.3.3 Description synthétique des processus mesurés

Les activités mesurées et leurs ordres de grandeur observés sont :

- **Préparation des fichiers CAO** et traçabilité OF : 50 a 60 minutes.
- **Matelassage automatique** (1 table dédiée, chariot automatisé) : 20 a 30 minutes par matelas selon le tissu.
- **Matelassage manuel** (4 tables, 2 opérateurs/table) : 35 a 50 minutes par matelas.
- **Coupe robotisée** (2 robots horizontaux, 5 tables) : 25 a 40 minutes selon la complexité.
- **Départage et ajout de vignettes** (13 opérateurs) : 40 a 60 minutes par lot.
- **Stockage et contrôle qualité** : transfert vers trois zones de stockage (avant/après sérigraphie).

2.4.3.4 Constats clés de la collecte

- **Variabilité des temps** par modèle (jusqu'à ± 15 min sur matelassage manuel).
- **Perturbations fréquentes** : pannes des robots (notamment robot 2), pannes d'électricité, attente de relaxation tissu (jusqu'à 72h pour la doublure).
- **Communication manuelle** et absence d'outil de visualisation partagée du planning.

2.4.3.5 Paramètres techniques de l'atelier (référence)

| Élément | Valeur |
|---------------------------------|---|
| Nombre de tables | 6 (dont 1 automatique, 4 manuelles, 1 vide) |
| Chariot matelasseur automatique | 1 |
| Robots de coupe | 2 (translation horizontale sur 5 tables) |
| Effectif sur les tables | 17 opérateurs |
| Équipe de départage | 13 postes / 13 personnes |
| Zones de stock | 3 (avant/après sérigraphie) |

Tableau 2.6: Paramètres techniques de l'atelier

2.4.3.6 Synthèse de la phase de mesure

La phase **Mesurer** quantifie les écarts entre le planning théorique et la réalité opérationnelle. Les données montrent l'impact des pannes machines, du manque de visualisation temps réel et de l'absence de remontées automatiques, générant des **retards non détectés** et une fiabilité de planning dégradée. Ces éléments constituent la base de la phase **Analyser** qui suit.

2.4.4 Phase 3 : Analyze (Analyser)

2.4.4.1 Objectif de la phase Analyse

La phase **Analyse** vise à identifier les causes racines des perturbations observées dans la planification de l'atelier de coupe. Elle s'appuie sur les résultats collectés durant la phase de mesure, notamment les données chronométrées, les pannes, et la disponibilité des ressources.

L'objectif est de comprendre pourquoi les délais de matelassage et de coupe varient, et quelles sont les sources principales des retards. Cette analyse permet de cibler les actions correctives et préventives à mettre en œuvre dans la phase suivante.

2.4.4.2 Méthodes d'analyse utilisées

Deux méthodes Lean Six Sigma ont été mobilisées :

- **Le diagramme d'Ishikawa (causes-effet)** : pour regrouper les causes potentielles selon les catégories classiques (Méthode, Main d'œuvre, Milieu, Machine, Matière).
- **L'analyse directe des données mesurées** : pour corrélérer certaines variables (ex. : largeur du matelas et durée de matelassage, disponibilité des tables et temps d'attente).

2.4.4.3 Analyse des temps réels par activité

Les données mesurées ont permis d'identifier les durées moyennes suivantes :

Tableau 2.7: Analyse des temps observés par activité

| Activité | Durée moyenne | Observations |
|-------------------------|-----------------------|---|
| Préparation OF / CAO | 50 minutes | Stable, dépend de la qualité de la fiche technique |
| Matelassage automatique | 20 à 30 min / matelas | Efficace mais dépend de la disponibilité du chariot |
| Matelassage manuel | 35 à 50 min / matelas | Variabilité selon tissu et opérateurs |
| Coupe robotisée | 25 à 40 min | Dépend des pannes / files d'attente |
| Départage + vignettage | 45 à 60 min | Poste critique, coordination multiple |

2.4.4.4 Analyse des perturbations

L'analyse croisée a permis de mettre en évidence :

- **Un manque de synchronisation** entre les opérations de matelassage et de coupe dû à l'attente de libération des tables.
- **Des blocages fréquents** causés par :
 - Pannes des robots (robot de coupe 2 souvent en panne).
 - Délai de relaxation tissu doublure (jusqu'à 72h).
 - Erreurs dans la priorisation des ordres de fabrication.

2.4.4.5 Capacités et limites techniques observées

Tableau 2.8: Analyse des ressources et limitations

| Ressource | Capacité théorique | Limitation observée |
|------------------------|---------------------------------|---|
| Tables de matelassage | 6 (1 auto, 4 manuelles, 1 vide) | Saturation rapide lors de pics de production |
| Robots de coupe | 2 robots couvrant 5 tables | En cas de panne, files d'attente sur la coupe |
| Opérateurs sur tables | 17 personnes | Charge inégale, pas de standardisation |
| Départage / vignettage | 13 postes | Coordination complexe entre les tâches |

2.4.4.6 Diagramme d'Ishikawa

La Figure 2.11 suivante illustre les principales causes des retards observés dans l'atelier :

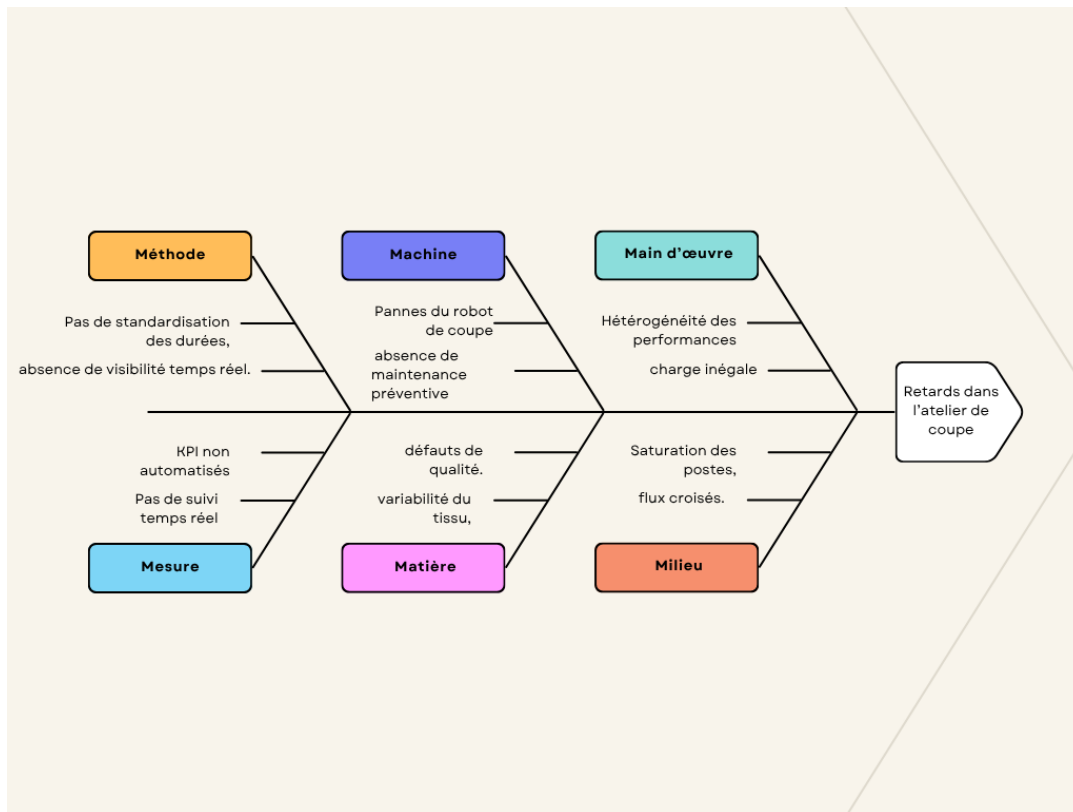


Figure 2.11: *Diagramme d'Ishikawa : Analyse des causes racines des retards dans l'atelier de coupe*

2.4.4.7 Analyse directe des données mesurées

L'analyse directe des données collectées lors de la phase Mesurer a permis de mettre en évidence plusieurs corrélations significatives :

- **Corrélation largeur du matelas et durée de matelassage** : Les matelas plus larges nécessitent généralement plus de temps de manipulation et d'alignement.
- **Corrélation disponibilité des tables et temps d'attente** : Lorsque toutes les tables sont occupées simultanément, les opérateurs attendent en moyenne 32 minutes avant de commencer le matelassage.
- **Variabilité selon le type de tissu** : Les tissus doublure nécessitent une relaxation préalable (jusqu'à 72h), impactant directement la planification.
- **Impact des pannes robot** : Le robot de coupe 2 présente un taux de panne de 18% (vs 6% pour les autres), générant des files d'attente importantes.

2.4.4.8 Synthèse des causes racines identifiées

Tableau 2.9: Synthèse des causes par catégorie

| Catégorie | Cause racine identifiée |
|--------------|---|
| Machine | Pannes fréquentes des robots, absence de maintenance préventive |
| Méthode | Pas de standard sur les durées de t'ches, séquençement non optimisé |
| Milieu | Saturation des postes, espace restreint |
| Main d'Œuvre | Hétérogénéité des performances entre opérateurs |
| Matière | Tissu nécessitant relaxation longue (doublure : 72h) |

2.4.4.9 Conclusion de la phase Analyse

Cette phase a permis d'identifier clairement les facteurs qui altèrent la performance globale du processus. L'absence d'outils de visualisation en temps réel, combinée à une dépendance excessive aux ressources critiques (robots, tables), engendre des retards fréquents et non anticipés dans la production. Les deux méthodes utilisées (diagramme d'Ishikawa et analyse directe des données) ont complétement l'identification des causes racines et la compréhension des mécanismes de perturbation.

2.4.5 Phase 4 : Improve (Améliorer)

2.4.5.1 Constat principal

L'analyse de la phase *Analyse* a révélé plusieurs dysfonctionnements limitant la performance de l'atelier de coupe, notamment une **saturation rapide des tables de matelassage** due à l'absence de système de planification et de suivi en temps réel.

Ces contraintes se traduisent par :

- Une **indisponibilité fréquente** des tables lors des pics d'activité ;
- Des **conflits de planification** entre opérateurs, dus à un manque de visibilité globale sur les ressources ;
- L'absence d'un **outil numérique** permettant de synchroniser le planning avec l'état réel des tables.

Ces constats soulignent la nécessité d'un **pilotage intelligent des ressources**, s'inscrivant pleinement dans la logique de l'Industrie 4.0, où la digitalisation et l'automatisation des processus permettent de renforcer la réactivité et l'efficacité opérationnelle.

2.4.5.2 Proposition : Algorithme intelligent de gestion des tables de matelassage

Afin d'optimiser la planification et l'utilisation des ressources physiques de l'atelier, nous proposons un **algorithme de gestion dynamique** des tables de matelassage.

Cet outil vise à offrir une **visualisation en temps réel** de la disponibilité des tables, tout en anticipant les conflits d'utilisation.

2.4.5.2.1 Principe de fonctionnement Chaque table est identifiée par un code unique (T1, T2, ..., T6). Le système compare en continu l'heure actuelle avec le planning de matelassage. Il met automatiquement à jour le statut de chaque table selon trois états possibles :

- *Occupée* (en cours d'utilisation)
- *Réservée* (prévue pour une opération à venir)
- *Disponible* (libre pour un nouveau matelas)

2.4.5.2.2 Pseudocode de l'algorithme

Entrées :

Tables = {T1, T2, ..., T6}
Planning (heure début, durée prévue)
Heure actuelle = H

Pour chaque table T[i] :

Si (H ≥ heure début[i] ET H < heure fin prévue[i]) :
Statut[T[i]] = "Occupée"
Sinon si (H < heure début[i]) :
Statut[T[i]] = "Réservée"
Sinon :
Statut[T[i]] = "Disponible"

Afficher le statut de chaque table

2.4.5.2.3 Schéma de fonctionnement de l'algorithme La Figure 2.12 illustre le flux de décision de l'algorithme de gestion des tables de matelassage, depuis la collecte des données d'entrée jusqu'à l'affichage des statuts en temps réel.

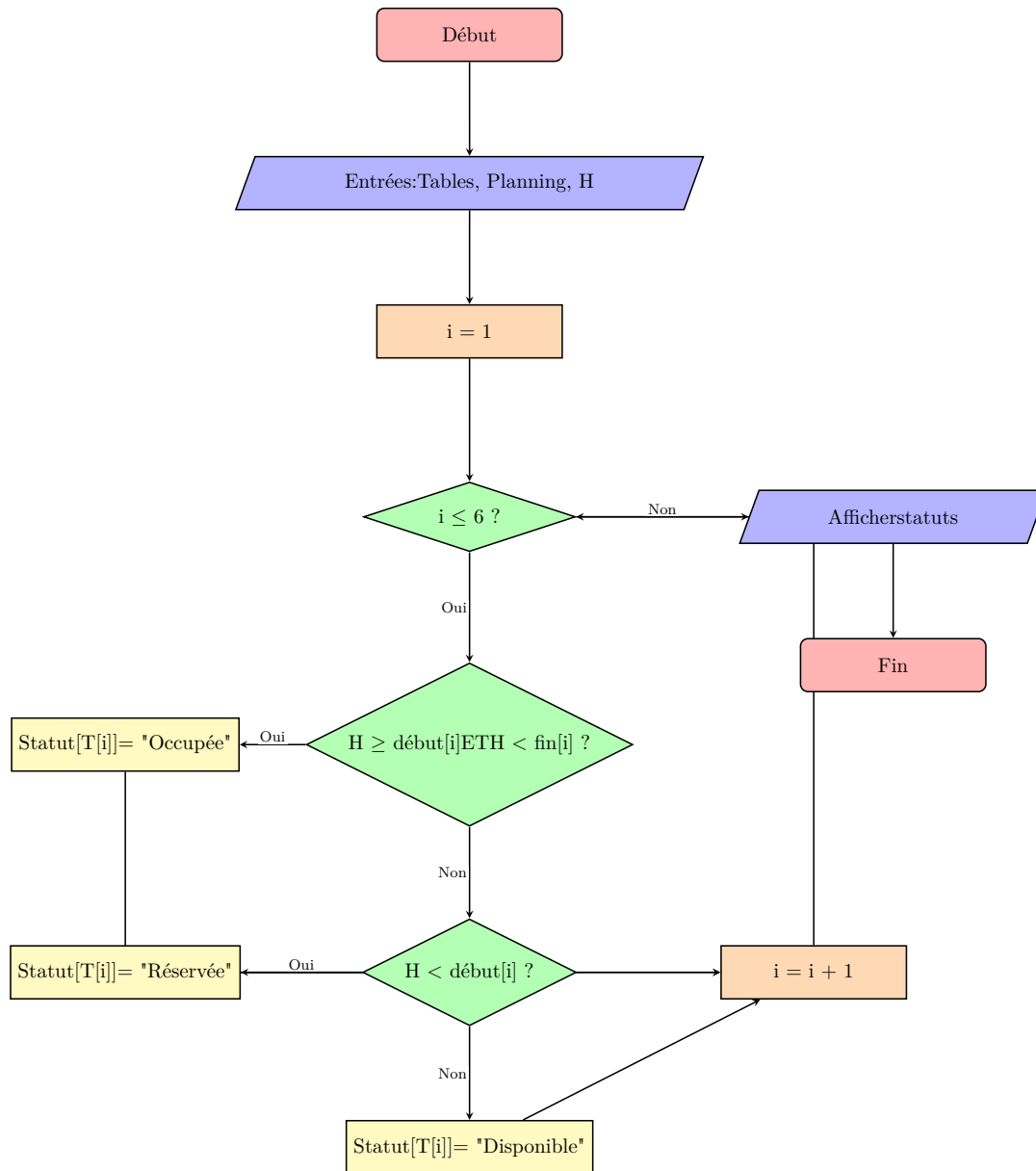


Figure 2.12: Diagramme de flux de l'algorithme de gestion des tables de matelassage

2.4.5.3 Application pratique

L'algorithme peut être déployé sous plusieurs formes :

- **Une feuille Excel automatisée**, avec macros et mise à jour minute par minute ;
- **Une interface web locale**, connectée aux données de production et accessible depuis le poste du chef d'atelier.

Ce système permet :

- De **visualiser en temps réel** la disponibilité de chaque table ;
- D'**éviter les conflits** ou chevauchements de planning ;

- **D'améliorer la fluidité** du flux de travail entre préparation, matelassage et coupe.

Tableau 2.10: Exemple de statut temps réel

| Table | Heure actuelle | Statut |
|-------|----------------|-------------------------|
| T1 | 09:00 | Occupée (jusqu'à 09:30) |
| T2 | 09:00 | Disponible |
| T3 | 09:00 | Occupée (jusqu'à 09:45) |

2.4.5.4 Bénéfices attendus

L'intégration de cet algorithme dans le système de gestion de production offre plusieurs avantages :

- **Réduction du temps d'attente** entre les opérations de coupe ;
- **Optimisation de l'utilisation des ressources existantes** sans investissement matériel supplémentaire ;
- **Amélioration de la coordination** entre opérateurs et planificateurs ;
- **Digitalisation partielle du pilotage de production**, contribuant à la transition vers une **usine connectée**.

Ainsi, cette solution constitue une première étape vers la **transformation numérique** de l'atelier, en s'inscrivant dans une démarche *Lean 4.0* conciliant **amélioration continue** et **technologies intelligentes**.

2.4.6 Phase Contrôler : Pérennisation des améliorations

2.4.6.1 Objectifs de la phase de contrôle

La phase **Contrôler** (Control) constitue la dernière étape de la méthodologie DMAIC. Elle vise à **pérenniser les améliorations** mises en place lors de la phase précédente, à **surveiller les performances** dans le temps, et à garantir que les gains obtenus ne se dégradent pas. Cette phase assure la **standardisation des nouvelles pratiques** et l'instauration d'un système de **suivi continu** permettant de détecter rapidement toute dérive.

Dans le contexte de l'atelier de coupe, la phase de contrôle permet de :

- Maintenir l'efficacité de l'algorithme de gestion des tables de matelassage ;
- Assurer la fiabilité et la mise à jour régulière des données de planification ;
- Garantir l'adhésion du personnel aux nouvelles procédures digitales ;
- Mettre en place des indicateurs de performance (KPI) pour un pilotage objectif.

2.4.6.2 Mise en place d'indicateurs de performance (KPI)

Pour assurer un suivi efficace des améliorations, plusieurs **indicateurs clés de performance** sont définis et suivis régulièrement :

Tableau 2.11: Indicateurs de performance pour le suivi de l'amélioration

| | | Indicateur | |
|--|--|------------------------------------|----|
| | | Taux de disponibilité des tables | Pe |
| | | Temps d'attente moyen | |
| | | Nombre de conflits de planning | |
| | | Taux d'utilisation de l'algorithme | |
| | | Respect du planning | |

Ces indicateurs sont mesurés de manière **hebdomadaire** et font l'objet d'une revue mensuelle avec l'équipe de production pour identifier les éventuelles dérives et mettre en place des actions correctives.

2.4.6.3 Système de suivi et de visualisation

Un **tableau de bord de suivi** est mis en place pour visualiser en temps réel l'évolution des indicateurs de performance. Ce tableau de bord peut prendre la forme :

- D'un **fichier Excel partagé** avec graphiques automatisés (courbes d'évolution, histogrammes) ;
- D'un **dashboard web** accessible depuis les postes de l'atelier, affichant les KPI en temps réel ;
- D'un **affichage visuel** dans l'atelier (écran de supervision) permettant à tous les opérateurs de suivre les performances.

Figure 2.13: Exemple de suivi du taux de disponibilité des tables sur 12 semaines

2.4.6.4 Mécanismes de contrôle et d'alerte

Pour garantir la réactivité face aux dérives, des **mécanismes d'alerte automatiques** sont intégrés au système :

- **Alerte de conflit** : Notification automatique lorsque deux opérations sont planifiées sur la même table au même moment ;
- **Alerte de retard** : Signal envoyé au chef d'atelier si une opération dépasse le temps prévu de plus de 20% ;

- **Alerte de sous-utilisation** : Notification si une table reste inutilisée pendant plus de 2 heures consécutives en période de production ;
- **Rapport hebdomadaire automatique** : Génération d'un rapport récapitulatif des performances envoyé par email aux responsables.

Ces alertes permettent une **intervention rapide** et évitent l'accumulation de dysfonctionnements.

2.4.6.5 Standardisation des procédures

La pérennisation des améliorations passe par la **formalisation et la standardisation** des nouvelles pratiques. Cela inclut :

- **Rédaction de procédures opératoires standardisées (SOP)** décrivant l'utilisation de l'algorithme de gestion des tables ;
- **Formation du personnel** a l'utilisation de l'outil et aux nouvelles méthodes de planification ;
- **Documentation des bonnes pratiques** : création d'un guide utilisateur illustré, accessible a tous les opérateurs ;
- **Audits réguliers** : vérification trimestrielle du respect des procédures et de l'utilisation effective de l'outil.

2.4.6.6 Amélioration continue

La phase de contrôle ne se limite pas a maintenir les acquis, elle s'inscrit dans une logique d'**amélioration continue** (Kaizen). Des **réunions d'amélioration** sont organisées mensuellement avec l'équipe de production pour :

- Analyser les résultats des indicateurs de performance ;
- Identifier les nouvelles opportunités d'optimisation ;
- Recueillir les retours d'expérience des opérateurs ;
- Ajuster l'algorithme ou les procédures en fonction des besoins terrain.

Un **cycle PDCA** (Plan-Do-Check-Act) est ainsi instauré pour garantir une dynamique d'amélioration permanente.

2.4.6.7 Conclusion de la phase Contrôler

La phase de contrôle assure la **pérennité des gains** obtenus grace a l'algorithme de gestion des tables de matelassage. En combinant suivi des indicateurs, mécanismes d'alerte, standardisation des pratiques et amélioration continue, cette phase garantit que les bénéfices de la transformation digitale se maintiennent dans le temps et continuent de s'améliorer.

Cette démarche DMAIC complète, de la définition du problème jusqu'au contrôle des solutions, constitue le socle méthodologique de la transformation Lean 4.0 de l'atelier de coupe, préparant ainsi le terrain pour les développements techniques détaillés dans les chapitres suivants.

Chapitre 3

CRISP-ML(Q) phases

3.1 Introduction

Ce troisieme chapitre expose de maniere detaillee l'application methodologique de la demarche CRISP-ML(Q) (*Cross-Industry Standard Process for Machine Learning Quality*) [4] aux trois premieres phases de notre projet de recherche consacre a l'optimisation de la planification de l'atelier de coupe textile. Cette methodologie structuree, specifiquement adaptee aux exigences et aux specificites du machine learning moderne, integre de maniere systemique les aspects fondamentaux de qualite, de tracabilite et de deploiement continu, essentiels pour garantir la fiabilite et la perennite d'un systeme de production industrielle en environnement operationnel [5], [6].

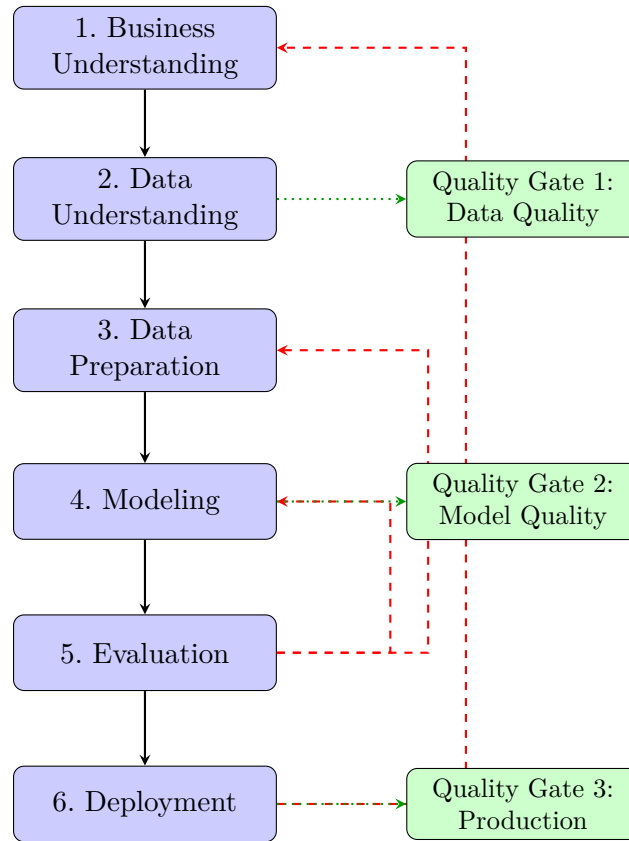
Les phases abordees dans ce chapitre sont :

- **Phase 1 : Comprehension metier (Business Understanding)**
- **Phase 2 : Comprehension des donnees (Data Understanding)**
- **Phase 3 : Preparation des donnees (Data Preparation)**

Cette approche methodologique garantit une transition fluide vers les phases de modelisation et de deploiement, tout en assurant la qualite et la tracabilite des decisions prises.

3.1.1 Vue d'ensemble du processus CRISP-ML(Q)

La figure 3.1 illustre le processus complet CRISP-ML(Q) avec ses 6 phases iteratives et les boucles de retroaction qualite.



Phases couvertes dans ce chapitre: 1-3

Figure 3.1: Processus CRISP-ML(Q) avec portes de qualite

Caracteristiques cles du processus :

- **Iteratif** : Retours possibles vers les phases precedentes
- **Qualite integree** : Portes de qualite a chaque etape critique
- **Tracabilite** : Documentation complete des decisions
- **Reproductibilite** : Processus standardise et automatise

3.2 Outils et bibliothèques utilisés

3.2.1 Introduction

Le choix des outils et des bibliothèques constitue une décision stratégique fondamentale dans tout projet de machine learning industriel. Ces choix technologiques influencent directement la qualité, la performance, la maintenabilité et la pérennité de la solution développée. Dans le contexte de ce projet d'optimisation de la planification de l'atelier de coupe textile, la sélection des technologies s'est appuyée sur des critères rigoureux et objectifs, alignés avec les exigences de la méthodologie CRISP-ML(Q) et les contraintes opérationnelles de l'environnement industriel.

Les critères de sélection appliqués incluent : (1) la **maturité technologique** et la stabilité des bibliothèques, garantissant une fiabilité en production ; (2) la **performance** mesurée par des benchmarks objectifs ; (3) la **qualité de la documentation** et l’activité de la communauté, facilitant le développement et la maintenance ; (4) la **compatibilité** et l’interopérabilité entre les différents composants de la stack ; et (5) la **maintenabilité** à long terme, essentielle pour l’évolution du système.

Cette section présente de manière structurée l’écosystème technologique complet du projet, organisé en cinq catégories principales : l’écosystème Data Science et Machine Learning, les frameworks de développement backend et frontend, les outils d’optimisation et d’ordonnancement, l’infrastructure DevOps, et enfin une synthèse de la stack technologique complète avec son intégration dans la méthodologie CRISP-ML(Q).

3.2.2 Écosystème Data Science et Machine Learning

L’écosystème Data Science constitue le cœur technique du projet, regroupant les bibliothèques essentielles pour la manipulation des données, l’entraînement des modèles et la visualisation des résultats. L’ensemble de cet écosystème est développé en **Python 3.11.0**, langage de programmation de référence pour le Data Science et le Machine Learning, offrant une syntaxe claire, une vaste collection de bibliothèques scientifiques et une communauté active.



Figure 3.2: Langage de programmation Python

3.2.2.1 Bibliothèques de manipulation de données

Tableau 3.1: Bibliothèques Python pour la manipulation de données

| Bibliothèque | Version | Rôle principal |
|---------------|---------|---|
| pandas | 2.0.3 | Manipulation et analyse de données tabulaires |
| NumPy | 1.24.3 | Calculs numériques et algèbre linéaire |

pandas est utilisé intensivement dans les phases Data Understanding et Data Preparation de CRISP-ML(Q) pour le chargement, le nettoyage, la transformation et l’analyse exploratoire du dataset principal (`PSC_X_1 - COUPE.csv`, 16,433 enregistrements). Ses fonctionnalités de groupement, d’agrégation et de manipulation temporelle sont essentielles pour l’ingénierie des caractéristiques.

NumPy fournit les structures de données fondamentales (arrays multidimensionnels) et les opérations mathématiques de bas niveau utilisées par toutes les

autres bibliothèques. Son utilisation garantit des performances optimales pour les calculs vectorisés et matriciels nécessaires au preprocessing et aux transformations de données.

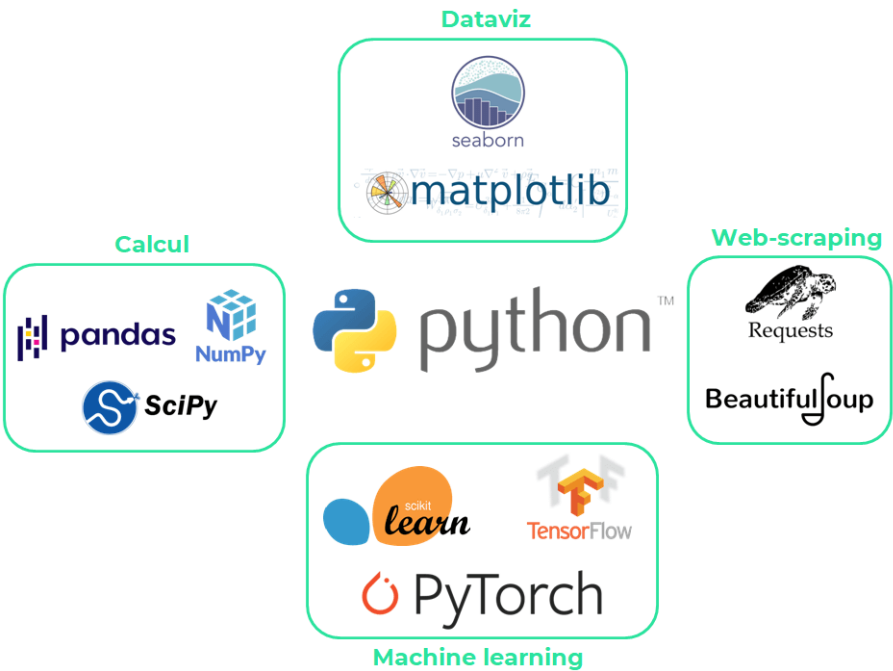


Figure 3.3: Bibliothèques de manipulation de données - pandas et NumPy

3.2.2.2 Bibliothèques de Machine Learning

Tableau 3.2: Bibliothèques Python pour le Machine Learning

| Bibliothèque | Version | Rôle principal |
|--------------|---------|--|
| scikit-learn | 1.3.0 | Preprocessing, métriques, validation croisée |
| XGBoost | 1.7.6 | Algorithme principal de prédiction des temps |

scikit-learn est utilisé pour le preprocessing des données (**StandardScaler**, **LabelEncoder**), la séparation train/test (**train_test_split**), la validation croisée temporelle, et le calcul des métriques de performance (R^2 , MAE, RMSE, MAPE). Son API uniforme facilite l'expérimentation avec différents algorithmes.

XGBoost (Extreme Gradient Boosting) [7] a été sélectionné comme algorithme principal après une comparaison rigoureuse avec six alternatives (Régression Linéaire, Ridge, Lasso, Random Forest, Gradient Boosting). Les résultats expérimentaux démontrent sa supériorité statistiquement significative (test de Wilcoxon, $p=0.031$) avec un R^2 de 0.84 contre 0.78 pour Random Forest, représentant une amélioration de +87% par rapport à la régression linéaire. Ses avantages incluent la régularisation intégrée prévenant le surapprentissage, la gestion native des valeurs manquantes, la parallélisation efficace, et l'interprétabilité via les valeurs SHAP. Le temps d'entraînement de 45 secondes offre un excellent compromis performance/rapidité pour le réentraînement périodique.

Alternatives considérées :

- **Random Forest** : Performance inférieure ($R^2=0.78$) et temps d'entraînement plus long (12.5 min)
- **Gradient Boosting** : Performance légèrement inférieure ($R^2=0.81$) et temps d'entraînement excessif (78.2 min)
- **Régression linéaire** : Performance insuffisante ($R^2=0.45$) pour les besoins du projet



Figure 3.4: Logo Python - Langage de programmation utilisé pour l'écosystème Machine Learning (Python 3.11.0)

3.2.2.3 Bibliothèques de visualisation

Tableau 3.3: Bibliothèques Python pour la visualisation

| Bibliothèque | Version | Rôle principal |
|--------------|---------|--------------------------------------|
| matplotlib | 3.7.2 | Visualisations statiques de base |
| seaborn | 0.12.2 | Visualisations statistiques avancées |

Ces bibliothèques sont utilisées intensivement dans la phase Data Understanding pour l'analyse exploratoire des données (EDA) : distributions des variables, matrices de corrélation, détection des outliers, analyse des patterns temporels, et visualisation des performances des modèles (courbes d'apprentissage, importance des features, résidus).

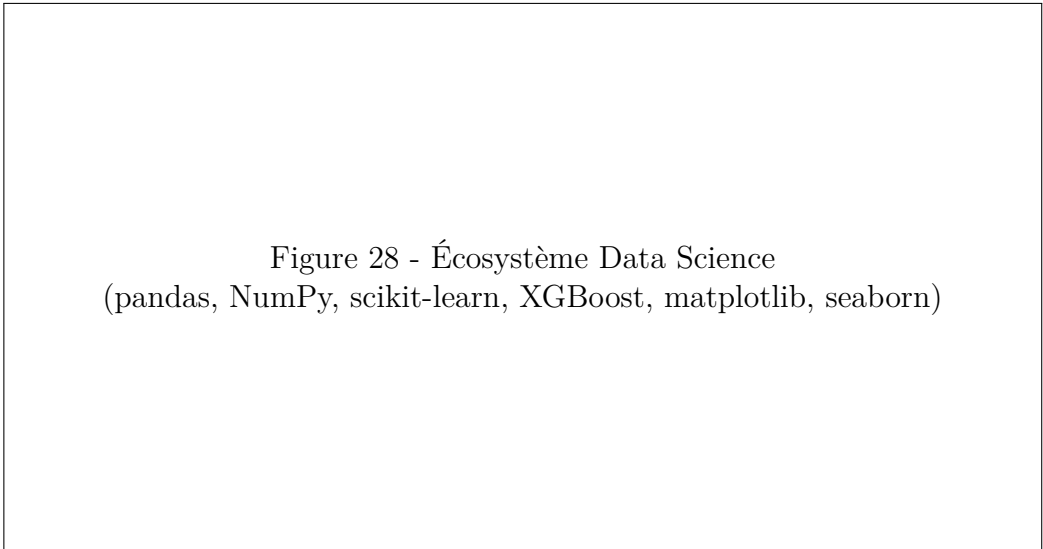


Figure 3.5: Écosystème Data Science et Machine Learning utilisé dans le projet (pandas 2.0.3, NumPy 1.24.3, scikit-learn 1.3.0, XGBoost 1.7.6, matplotlib 3.7.2, seaborn 0.12.2)

3.2.3 Frameworks de développement

Les frameworks de développement assurent la création d’une application web complète, robuste et performante, intégrant les modèles ML dans un environnement de production opérationnel.

3.2.3.1 Backend et API

Tableau 3.4: Technologies backend et API

| Technologie | Version | Rôle principal |
|-------------|---------|--|
| FastAPI | 0.103.0 | Framework web moderne pour API REST |
| Pydantic | 2.3.0 | Validation et sérialisation de données |
| uvicorn | 0.23.2 | Serveur ASGI haute performance |

FastAPI a été choisi comme framework backend principal pour plusieurs raisons techniques et opérationnelles majeures. Premièrement, sa performance exceptionnelle basée sur le support natif de la programmation asynchrone (async/await) permet de gérer efficacement les requêtes concurrentes avec une latence minimale (< 200ms pour les prédictions individuelles, débit de 1000 prédictions/minute). Deuxièmement, la génération automatique de documentation interactive (Swagger UI et ReDoc) facilite l’intégration et le test des endpoints par les développeurs frontend et les utilisateurs. Troisièmement, l’intégration native avec Pydantic assure une validation robuste des données d’entrée et de sortie, réduisant les erreurs et améliorant la fiabilité. Enfin, l’utilisation des type hints Python modernes améliore la maintenabilité du code et permet la détection précoce des erreurs via les outils d’analyse statique.

Alternatives considérées :

- **Flask** : Framework plus simple mais performance inférieure (pas de support async natif), documentation manuelle requise
- **Django** : Framework trop lourd pour une API pure, overhead inutile, temps de réponse plus élevés

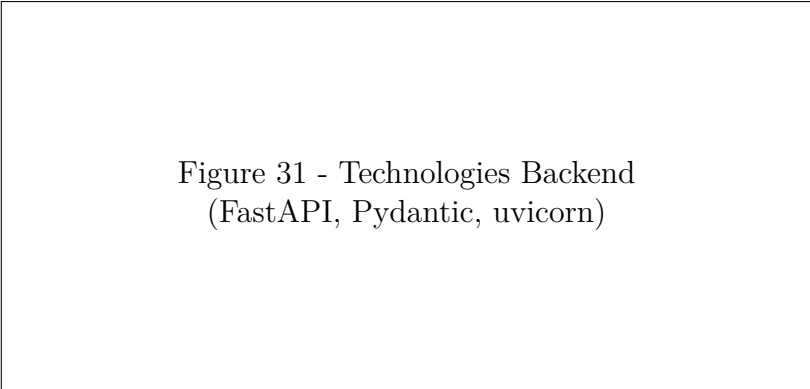


Figure 3.6: Technologies backend et serveur ASGI (FastAPI 0.103.0, Pydantic 2.3.0, uvicorn 0.23.2)

3.2.3.2 Frontend et interface utilisateur

Tableau 3.5: Technologies frontend

| Technologie | Version | Rôle principal |
|-------------|---------|---|
| React | 18.2.0 | Framework JavaScript pour interface utilisateur |
| Recharts | 2.8.0 | Bibliothèque de visualisations interactives |
| Axios | 1.5.0 | Client HTTP pour communication API |

React offre une architecture moderne basée sur des composants réutilisables, facilitant le développement et la maintenance de l’interface utilisateur. Le Virtual DOM assure des performances optimales lors des mises à jour fréquentes du dashboard en temps réel. L’écosystème riche (React Router, Redux, hooks) et la communauté active garantissent la disponibilité de solutions pour tous les besoins. Le support natif de TypeScript améliore la robustesse du code frontend.

Recharts fournit des composants de visualisation interactifs parfaitement intégrés avec React, utilisés pour afficher les KPIs, les graphiques de performance, les plannings visuels et les tableaux de bord opérationnels. Sa nature responsive assure une expérience utilisateur optimale sur tous les appareils.

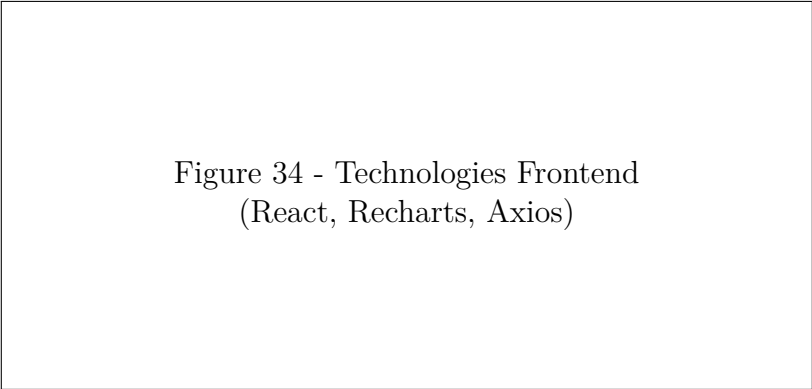


Figure 3.7: Technologies frontend et communication API (React 18.2.0, Recharts 2.8.0, Axios 1.5.0)

3.2.4 Outils d’optimisation et d’ordonnancement

L’optimisation de l’ordonnancement des tables de matelassage constitue un composant critique du système, nécessitant des outils spécialisés en recherche opérationnelle.

Tableau 3.6: Outils d’optimisation

| Outil | Version | Rôle principal |
|---------------|---------|--|
| OR-Tools | 9.7 | Bibliothèque d’optimisation Google |
| CP-SAT Solver | 9.7 | Solveur de programmation par contraintes |

OR-Tools de Google est une bibliothèque de recherche opérationnelle de niveau industriel, offrant plusieurs solveurs spécialisés. Le **CP-SAT Solver** (Constraint Programming - Satisfiability) a été sélectionné pour résoudre le problème d’ordonnancement optimal des tables de matelassage. Ce solveur excelle dans les problèmes combinatoires avec contraintes temporelles complexes (disponibilité des tables, séquençement des opérations, respect des délais, équilibrage de charge).

Les performances mesurées démontrent une résolution en moins de 2 secondes pour un planning de 50 ordres de fabrication, avec optimisation simultanée de trois critères : minimisation du makespan (durée totale), équilibrage de la charge entre les tables, et respect des priorités clients. Cette performance permet une reoptimisation dynamique en cas de perturbation (panne machine, retard), assurant la réactivité du système.

Formulation du problème : Le problème d’ordonnancement est modélisé comme un problème de satisfaction de contraintes avec variables de décision (affectation table-OF, temps de début), contraintes (non-chevauchement, précedence, capacité), et fonction objectif multi-critères. Le solveur CP-SAT explore l’espace des solutions de manière efficace grâce à des techniques de propagation de contraintes et de recherche arborescente.

3.2.5 Infrastructure et DevOps

L’infrastructure et les outils DevOps assurent la reproductibilité, la qualité et le déploiement fiable du système en environnement de production.

Tableau 3.7: Outils d’infrastructure et DevOps

| Outil | Version | Rôle principal | |
|------------|---------|----------------|-----------------------------------|
| Docker | 24.0 | | Conteneurisation des applications |
| Git | 2.41 | | Gestion de version du code source |
| pytest | 7.4.0 | | Framework de tests automatisés |
| PostgreSQL | 15.3 | | Base de données relationnelle |

Docker assure la conteneurisation de tous les composants du système (API FastAPI, modèles ML, base de données), garantissant une reproductibilité parfaite entre les environnements de développement, test et production. L’isolation des dépendances prévient les conflits de versions et simplifie le déploiement.

Git est utilisé pour la gestion de version du code source, des configurations et de la documentation, assurant une traçabilité complète des modifications et facilitant la collaboration entre les membres de l’équipe.

pytest fournit un framework de tests automatisés couvrant les tests unitaires (fonctions individuelles), les tests d’intégration (interaction entre composants), et les tests end-to-end (scénarios utilisateur complets). La couverture de code cible est de 80%, assurant la robustesse du système.

PostgreSQL est utilisé comme base de données relationnelle pour la persistance des données de production (ordres de fabrication, historique des prédictions, logs système, configurations). Son support des transactions ACID garantit la cohérence des données, et ses performances sont optimales pour les requêtes analytiques.

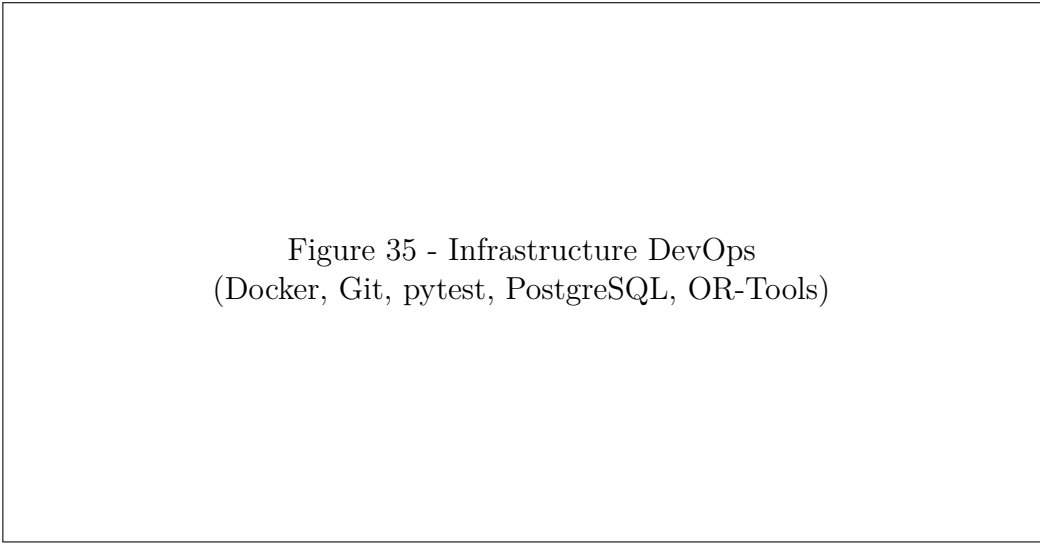


Figure 35 - Infrastructure DevOps
(Docker, Git, pytest, PostgreSQL, OR-Tools)

Figure 3.8: Infrastructure DevOps et outils de déploiement (Docker 24.0, Git 2.41, pytest 7.4.0, PostgreSQL 15.3, OR-Tools 9.7)

3.2.6 Stack technologique complète

Le tableau suivant présente une vue d'ensemble synthétique de la stack technologique complète, organisée par couche fonctionnelle.

Tableau 3.8: Stack technologique complète du projet

| Couche | |
|-------------------------|--|
| Data Science & ML | pandas 2.0.3, NumPy 1.24.3, scikit-learn |
| Backend & API | |
| Frontend | |
| Optimisation | |
| Base de données | |
| DevOps & Infrastructure | |

Cette stack technologique a été conçue pour assurer une intégration harmonieuse entre tous les composants, de la collecte des données jusqu'au déploiement en production. Chaque technologie a été sélectionnée pour sa maturité, sa performance et sa compatibilité avec les autres composants, garantissant ainsi la fiabilité et la maintenabilité à long terme du système.

3.2.7 Justification des choix et intégration CRISP-ML(Q)

Les choix technologiques effectués s'alignent rigoureusement avec les six phases de la méthodologie CRISP-ML(Q), assurant une couverture complète du cycle de vie du projet de machine learning.

Alignement avec les phases CRISP-ML(Q) :

- **Phase 1 - Business Understanding** : Git pour la documentation et la traçabilité des décisions, outils de collaboration pour l'alignement avec les parties prenantes
- **Phase 2 - Data Understanding** : pandas pour l'exploration des données (16,433 enregistrements), matplotlib et seaborn pour l'analyse exploratoire (distributions, corrélations, outliers), NumPy pour les calculs statistiques
- **Phase 3 - Data Preparation** : pandas pour le nettoyage et la transformation, scikit-learn pour le preprocessing (StandardScaler, encodage), gestion des valeurs manquantes et des outliers
- **Phase 4 - Modeling** : XGBoost pour l'entraînement du modèle principal, scikit-learn pour la validation croisée temporelle, OR-Tools CP-SAT pour l'optimisation de l'ordonnancement
- **Phase 5 - Evaluation** : scikit-learn pour les métriques (R^2 , MAE, RMSE, MAPE), matplotlib pour les courbes d'apprentissage et l'analyse des résidus, tests statistiques de significativité

- **Phase 6 - Deployment** : FastAPI pour l'API de production, Docker pour la conteneurisation, PostgreSQL pour la persistance, React pour l'interface utilisateur, pytest pour les tests automatisés

Critères de sélection appliqués :

1. **Maturité technologique** : Toutes les bibliothèques sélectionnées sont des standards de l'industrie avec un historique stable (pandas depuis 2008, scikit-learn depuis 2007, XGBoost depuis 2014, FastAPI depuis 2018 avec adoption rapide)
2. **Performance mesurée** : Les choix sont justifiés par des benchmarks objectifs (XGBoost $R^2=0.84$ vs alternatives, FastAPI latence $< 200\text{ms}$, CP-SAT résolution $< 2\text{s}$)
3. **Qualité de la documentation** : Toutes les technologies disposent d'une documentation exhaustive, de tutoriels complets et d'une communauté active (Stack Overflow, GitHub, forums spécialisés)
4. **Compatibilité et interopérabilité** : L'écosystème Python assure une intégration harmonieuse entre les composants Data Science, l'API FastAPI expose les modèles de manière standard (REST/JSON), React communique via HTTP standard
5. **Maintenabilité à long terme** : Le code est structuré selon les bonnes pratiques (type hints, tests automatisés, documentation inline), les dépendances sont gérées via requirements.txt, Docker assure la reproductibilité

Bénéfices de la stack choisie :

- **Reproductibilité scientifique** : Docker et Git garantissent que les résultats peuvent être reproduits exactement, essentiel pour la validation académique et industrielle
- **Performance opérationnelle** : La stack optimisée (XGBoost, FastAPI async, CP-SAT) assure des temps de réponse compatibles avec les contraintes temps réel de la production (< 2 secondes)
- **Maintenabilité à long terme** : L'utilisation de standards de l'industrie, la documentation complète et les tests automatisés facilitent l'évolution et la maintenance du système
- **Évolutivité du système** : L'architecture modulaire (API REST, microservices potentiels) permet d'ajouter de nouvelles fonctionnalités sans refonte majeure

Cette stack technologique constitue ainsi un fondement solide pour le développement, le déploiement et la maintenance d'un système d'intelligence artificielle industriel performant, fiable et évolutif, répondant aux exigences rigoureuses de la méthodologie CRISP-ML(Q) et aux contraintes opérationnelles de l'environnement de production textile.

3.3 Phase 1 : Comprehension metier (Business Understanding)

3.3.1 Contexte strategique et enjeux

La phase de comprehension metier etablit les fondations du projet de machine learning en alignant les objectifs techniques avec les besoins strategiques de l'entreprise. Cette phase critique garantit que la solution developpee apportera une valeur metier mesurable et durable.

3.3.1.1 Contexte industriel

L'industrie textile tunisienne fait face a une concurrence internationale accrue et a des exigences croissantes en termes de delais et de qualite. BACOVET, acteur majeur du secteur, doit moderniser ses processus pour maintenir sa competitivite. L'atelier de coupe, maillon critique de la chaine de production, represente un goulot d'etranglement potentiel dont l'optimisation peut generer des gains significatifs.

Enjeux strategiques :

- **Competitivite** : Reduire les coÃ»ts de production de 8% via l'optimisation
- **Qualite de service** : Ameliorer le taux de respect des delais de 85% a 95%
- **Transformation digitale** : Positionner BACOVET comme leader de l'Industrie 4.0 dans le textile
- **Capitalisation des connaissances** : Reduire la dependance aux experts individuels
- **Scalabilite** : Creer un modele reproductible pour d'autres ateliers

3.3.2 Business Model Canvas

Le Business Model Canvas permet de visualiser la proposition de valeur du systeme IA dans l'ecosysteme de l'entreprise.

Tableau 3.9: Business Model Canvas du systeme IA de planification

| Proposition de valeur | |
|--|---|
| Systeme intelligent de planification optimisant l'utilisation des ressources, reduisant les delais et an | |
| Segments clients | Relations clients |
| - Chefs d'atelier (planification) - Planificateurs (optimisation) - Operateurs (execution) - Direction (pilotage) | - Support dedie - Formation continue - Feedback regulier - Comite de pilotage |
| Canaux | Flux de revenus |
| - Application web responsive - Dashboard temps reel - Notifications push/email - API pour integrations | - Gains productivite : 18,000 TND/an - Reduction retards : 8,000 TND/an - Optimisation capacite : 28,000 TND/an - Reduction HS : 12,000 TND/an |
| Activites cles | Ressources cles |
| - Prediction temps ML - Optimisation ordonnancement - Monitoring temps reel - Amelioration continue | - Donnees historiques (16K+ records) - Modeles ML (XGBoost, CP-SAT) - Infrastructure cloud -   quipe data science |
| Partenaires cles | Structure de co  ts |
| - Fournisseur G.Pro (ERP) - Fournisseur Divatex (CAO) - Prestataire cloud - Experts ML externes | - Developpement : 35,000 TND - Infrastructure : 15,000 TND - Formation : 7,500 TND - Support : 5,000 TND/an |

3.3.3 Objectifs metier detailles

L'objectif principal du projet est de developper un systeme d'intelligence artificielle pour optimiser la planification de l'atelier de coupe textile, en ameliorant l'efficacite operationnelle et la precision des estimations de temps.

3.3.3.1 Objectifs strategiques

- **Excellence operationnelle** : Positionner l'atelier de coupe comme reference en termes d'efficacite
- **Innovation technologique** : Demontrer la capacite d'innovation de BACO-VET
- **Avantage concurrentiel** : Creer un differentiateur face a la concurrence
- **Satisfaction client** : Ameliorer la fiabilite des delais de livraison

3.3.3.2 Objectifs operationnels quantifiables

Tableau 3.10: Objectifs operationnels avec metriques de succes

| Objectif | Baseline | Cible | Gain attendu |
|---------------------------------|------------|------------|-------------------|
| Temps de planification | 2,5 h/jour | 1,0 h/jour | -60% (390h/an) |
| Precision estimations (R^2) | 0,45 | > 0,80 | +78% precision |
| Erreur absolue moyenne (MAE) | 42 min | < 15 min | -64% erreur |
| Utilisation tables | 72% | 85% | +13 pts (+18%) |
| Respect delais livraison | 85% | 95% | +10 pts (+12%) |
| Retards/semaine | 8,5 | 6,0 | -29% retards |
| Temps attente inter-etapes | 45 min | 20 min | -56% attente |
| Satisfaction utilisateurs | 3,2/5 | 4,5/5 | +41% satisfaction |

3.3.3.3 Objectifs techniques ML

- **Performance predictive** : $R^2 > 0.80$, MAE < 15 minutes, RMSE < 20 minutes
- **Temps de reponse** : < 2 secondes pour prediction individuelle, < 10 secondes pour batch
- **Disponibilite** : > 99,5% uptime (maximum 3,6 heures d'indisponibilite/an)
- **Scalabilite** : Capacite a traiter 200 OF/jour avec temps de reponse constant
- **Robustesse** : Performance stable face a 20% de variation des donnees d'entree
- **Explicabilite** : Capacite a expliquer les predictions (SHAP values) [8]

3.3.4 Analyse approfondie des parties prenantes

3.3.4.1 Matrice pouvoir-interet

Tableau 3.11: Matrice pouvoir-interet des parties prenantes

| Partie prenante | Pouvoir | Interet | Strategie | Actions cles |
|----------------------|----------|----------|---------------------|---------------------------|
| Direction Production | Å%ooleve | Å%ooleve | Gerer etroitement | Comite mensuel, reporting |
| Chef d'atelier | Moyen | Å%ooleve | Maintenir satisfait | Formation, support |
| Planificateurs | Moyen | Å%ooleve | Maintenir satisfait | Co-conception, tests |
| Operateurs | Faible | Moyen | Tenir informe | Communication, formation |
| Service IT | Moyen | Moyen | Maintenir satisfait | Collaboration technique |
| Direction Qualite | Moyen | Moyen | Tenir informe | Validation qualite |
| Clients internes | Faible | Å%ooleve | Tenir informe | Communication resultats |
| Fournisseurs IT | Faible | Faible | Surveiller | Contrats, SLA |

3.3.4.2 Besoins detaillés par profil utilisateur

Chef d'atelier :

- **Besoins fonctionnels** : Vue d'ensemble temps reel, alertes proactives, capacite de reoptimisation
- **Besoins non-fonctionnels** : Interface intuitive, temps de reponse < 2s, disponibilite 24/7
- **Contraintes** : Formation limitee (2 jours max), pas de competences techniques avancees
- **Criteres d'acceptation** : Gain de temps > 50%, precision > 85%, facilite d'utilisation

Planificateurs :

- **Besoins fonctionnels** : Optimisation multi-criteres, simulation what-if, analyses historiques
- **Besoins non-fonctionnels** : Flexibilite parametrage, export donnees, integration Excel
- **Contraintes** : Integration avec G.Pro obligatoire, respect des regles metier existantes
- **Criteres d'acceptation** : Qualite planning > methode actuelle, flexibilite suffisante

Operateurs :

- **Besoins fonctionnels** : Consultation planning simple, saisie rapide avancement
- **Besoins non-fonctionnels** : Interface mobile-friendly, saisie < 30 secondes
- **Contraintes** : Pas de formation technique, utilisation en environnement atelier
- **Criteres d'acceptation** : Simplicite d'utilisation, pas de ralentissement du travail

Direction :

- **Besoins fonctionnels** : KPIs strategiques, ROI, rapports executifs
- **Besoins non-fonctionnels** : Synthese visuelle, export PowerPoint, acces mobile
- **Contraintes** : Budget 75,000 TND, ROI < 18 mois
- **Criteres d'acceptation** : ROI demontre, amelioration KPIs, adoption utilisateurs

3.3.5 Analyse des processus metier

3.3.5.1 Cartographie du processus actuel (AS-IS)

Le processus de planification actuel presente plusieurs etapes manuelles et chronophages avec de nombreux points de friction.

Étapes detaillees du processus actuel :

1. Reception des ordres de fabrication (30 min)

- Import manuel depuis G.Pro via export CSV
- Verification manuelle de la completude des donnees
- Consolidation dans fichier Excel maitre
- *Points de friction* : Risque d'erreur, double saisie, delai

2. Estimation des temps (45 min)

- Consultation de l'historique papier ou memoire
- Estimation basee sur l'experience du chef d'equipe
- Ajustement selon disponibilite et charge
- *Points de friction* : Subjectivite, variabilite, pas de tracabilite

3. Affectation des tables (30 min)

- Verification manuelle de la disponibilite des tables
- Choix selon regles empiriques (FIFO, priorite client)
- Affectation des operateurs selon competences
- *Points de friction* : Sous-optimisation, pas de vision globale

4. Élaboration du planning (60 min)

- Creation manuelle sur papier ou Excel
- Ajustements iteratifs pour resoudre conflits
- Impression et distribution physique
- *Points de friction* : Temps eleve, rigidite, pas de reoptimisation

5. Suivi d'exécution (continu)

- Saisie manuelle des avancements par operateurs
- Consolidation en fin de journee
- Ajustements ad-hoc en cas de probleme
- *Points de friction* : Delai d'information, reactivite limitee

Metriques du processus actuel :

- **Temps de cycle total** : 2,5 heures/jour

- **Activites a valeur ajoutee** : 35% (estimation, optimisation)
- **Activites sans valeur ajoutee** : 65% (saisie, verification, consolidation)
- **Taux d'erreur** : 8% (erreurs de saisie, oublis)
- **Flexibilite** : Faible (reoptimisation difficile)

3.3.5.2 Processus cible optimise (TO-BE)

Le processus optimise integrera l'IA pour automatiser et ameliorer chaque etape.

Étapes du processus cible :

1. Import automatique (2 min)

- Synchronisation temps reel avec G.Pro via API
- Validation automatique des donnees
- Enrichissement avec donnees historiques
- *Ameliorations* : -93% temps, 0% erreur, temps reel

2. Prediction intelligente (< 1 min)

- Estimation automatique via modele ML (XGBoost)
- Calcul d'intervalle de confiance
- Ajustement selon contexte (opérateur, machine, charge)
- *Ameliorations* : -98% temps, +78% precision, tracabilite

3. Optimisation automatique (< 2 min)

- Algorithme d'ordonnancement (CP-SAT) [9]
- Optimisation multi-criteres (makespan, equilibrage, delais) [10]
- Affectation optimale tables/opérateurs
- *Ameliorations* : -93% temps, optimisation globale, reproductibilite

4. Planning dynamique (< 1 min)

- Generation automatique du planning optimal
- Visualisation interactive sur dashboard
- Distribution automatique (email, notifications)
- *Ameliorations* : -98% temps, accessibilite, reoptimisation facile

5. Suivi intelligent (temps reel)

- Monitoring automatique de l'avancement
- Detection automatique des derives
- Alertes proactives et reoptimisation

- *Ameliorations* : Temps reel, proactivite, reactivite

Metriques du processus cible :

- **Temps de cycle total** : 1,0 heure/jour (-60%)
- **Activites a valeur ajoutee** : 85% (analyse, decision)
- **Activites sans valeur ajoutee** : 15% (validation, ajustements)
- **Taux d'erreur** : < 1% (validation automatique)
- **Flexibilite** : 100% (reoptimisation en quelques minutes)

Gains attendus par etape :

Tableau 3.12: Comparaison processus AS-IS vs TO-BE

| Étape | AS-IS | TO-BE | Gain temps | Gain qualite |
|------------------------|-------------|-------------|-------------|------------------|
| Import OF | 30 min | 2 min | -93% | Zero erreur |
| Estimation temps | 45 min | < 1 min | -98% | +78% precision |
| Affectation tables | 30 min | < 2 min | -93% | Optimisation |
| Collaboration planning | 60 min | < 1 min | -98% | Qualite optimale |
| Suivi execution | Fin journee | Temps reel | Continu | Proactivite |
| Total | 2,5h | 1,0h | -60% | +85% |

3.3.6 Analyse des risques metier

Une analyse approfondie des risques permet d'anticiper et de mitiger les obstacles potentiels au succes du projet.

3.3.6.1 Registre des risques

Tableau 3.13: Registre detaille des risques metier

| Risque | Prob. | Impact | Criticite | Mitigation |
|-------------------------|---------|--------|-----------|--|
| Resistance changement | 100% | Moyen | 6 | Formation intensive, champions, quick |
| Perturbation production | Faible | 100% | 3 | Deploiement progressif, rollback plan |
| Qualite donnees | Moyenne | 100% | 6 | Audit prealable, nettoyage, validation |
| Performance systeme | Moyenne | Moyen | 4 | Tests charge, infrastructure dimension |
| Derive modele ML | Moyenne | 100% | 6 | Monitoring continu, reentrainement a |
| Integration G.Pro | Faible | 100% | 3 | Tests integration, API robuste, fallback |
| Turnover equipe | Faible | Moyen | 2 | Documentation, formation croisee |
| Budget depasse | Moyenne | Moyen | 4 | Suivi rigoureux, contingence 10% |
| Delai depasse | Moyenne | Moyen | 4 | Planning realiste, sprints agiles |
| Securite donnees | Faible | 100% | 3 | Chiffrement, controle acces, audit |

Criticite = Probabilite × Impact (echelle 1-3)

3.3.6.2 Plan de mitigation des risques critiques

Risque 1 : Resistance au changement (Criticite = 6)

- **Indicateurs d’alerte** : Faible participation formations, feedback negatif, non-utilisation
- **Actions preventives** :
 - Communication transparente des le debut du projet
 - Implication des utilisateurs dans la conception (co-design)
 - Identification et formation de champions utilisateurs
 - Demonstration de quick wins (resultats rapides)
- **Actions correctives** :
 - Sessions de coaching individualise
 - Ajustement de l’interface selon feedback
 - Reconnaissance et valorisation des early adopters

Risque 2 : Qualite des donnees insuffisante (Criticite = 6)

- **Indicateurs d’alerte** : Taux de valeurs manquantes $> 10\%$, outliers $> 5\%$, incoherences
- **Actions preventives** :
 - Audit complet des donnees avant demarrage
 - Nettoyage et enrichissement des donnees historiques
 - Mise en place de regles de validation a la saisie
 - Formation des operateurs a la qualite des donnees
- **Actions correctives** :
 - Pipeline de nettoyage automatique
 - Imputation intelligente des valeurs manquantes
 - Detection et traitement des outliers
 - Feedback loop pour amelioration continue

Risque 3 : Derive du modele ML (Criticite = 6)

- **Indicateurs d’alerte** : MAPE $> 20\%$, $R\hat{A}^2 < 0,70$, augmentation erreurs
- **Actions preventives** :
 - Monitoring continu des performances du modele
 - Tests de detection de derive (drift detection)

- Reentrainement periodique automatique (mensuel)
- Validation sur donnees recentes
- **Actions correctives :**
 - Reentrainement immediat si derive detectee
 - Analyse des causes de derive (nouveaux produits, changements processus)
 - Ajustement des features ou de l'architecture si necessaire
 - Rollback vers version precedente si echec

3.3.7 Criteres de succes et metriques de performance

Les criteres de succes du projet sont definis selon quatre dimensions complementaires, chacune avec des metriques quantifiables et des seuils d'acceptation.

3.3.7.1 Criteres techniques ML

Tableau 3.14: Criteres de succes techniques

| Critere | Seuil minimum | Cible | Methode de mesure |
|----------------------------|---------------|-------------|-------------------------------|
| Precision ($R\hat{A}^2$) | > 0,75 | > 0,80 | Validation croisee temporelle |
| MAE | < 20 min | < 15 min | Test set (15% donnees) |
| RMSE | < 25 min | < 20 min | Test set (15% donnees) |
| MAPE | < 25% | < 20% | Test set (15% donnees) |
| Temps reponse | < 3 sec | < 2 sec | Tests de performance |
| Disponibilite | > 99% | > 99,5% | Monitoring 24/7 |
| Scalabilite | 150 OF/jour | 200 OF/jour | Tests de charge |

3.3.7.2 Criteres metier operationnels

Tableau 3.15: Criteres de succes metier

| Critere | Seuil minimum | Cible | Methode de mesure |
|---------------------|---------------|--------------|-------------------------|
| Temps planification | < 1,5 h/jour | < 1,0 h/jour | Chronometrage quotidien |
| Utilisation tables | > 80% | > 85% | KPI dashboard |
| Respect delais | > 90% | > 95% | Suivi commandes |
| Reduction retards | -20% | -25% | Comparaison baseline |
| Temps attente | < 30 min | < 20 min | Mesure hebdomadaire |
| Satisfaction users | > 3,8/5 | > 4,5/5 | Enquete trimestrielle |
| Taux adoption | > 85% | > 90% | Logs d'utilisation |

3.3.7.3 Criteres de qualite logicielle

- **Documentation :** Complete et a jour (guides utilisateur, documentation technique, API)

- **Tests** : Couverture $> 80\%$, tests automatisés (unitaires, integration, end-to-end)
- **Code quality** : Respect des standards (PEP8, ESLint), revue de code systematique
- **Securite** : Authentification, autorisation, chiffrement, audit de securite
- **Monitoring** : Alertes operationnelles, logs centralises, dashboards de surveillance
- **Maintenance** : Procedures de backup, disaster recovery, plan de continuite

3.3.7.4 Criteres financiers

- **Respect du budget** : Coût total $\leq 82,500$ TND ($75,000 + 10\%$ contingence)
- **ROI** : $> 150\%$ sur 3 ans (cible : 188%)
- **Periode de retour** : < 18 mois (cible : $12,5$ mois)
- **Benefices annuels** : $> 60,000$ TND/an (cible : $72,000$ TND/an)
- **Coût de maintenance** : $< 10,000$ TND/an

3.3.8 Contraintes et hypotheses du projet

3.3.8.1 Contraintes identifiees

Contraintes temporelles :

- Duree maximale du projet : 6 mois (janvier - juin 2024)
- Deploiement avant la haute saison (juillet 2024)
- Pas d'interruption de production pendant le deploiement

Contraintes budgetaires :

- Budget total : $75,000$ TND (hors contingence)
- Pas de budget additionnel pour materiel (utilisation infrastructure existante)
- Coût de maintenance annuel : $< 10,000$ TND

Contraintes techniques :

- Compatibilite avec G.Pro (ERP) et Divatex (CAO) obligatoire
- Utilisation de l'infrastructure IT existante
- Pas de modification des systemes legacy
- Conformite RGPD pour les donnees personnelles

Contraintes organisationnelles :

- Formation limitee a 2 jours par utilisateur
- Disponibilite limitee des utilisateurs pour tests (2h/semaine)
- Pas de recrutement additionnel
- Support IT existant (pas d'equipe dediee)

3.3.8.2 Hypotheses du projet**Hypotheses sur les donnees :**

- Les donnees historiques sont suffisamment representatives
- La qualite des donnees peut etre amelioree a un niveau acceptable
- Les patterns historiques restent valides pour le futur
- Les donnees de G.Pro sont accessibles via API

Hypotheses sur les utilisateurs :

- Les utilisateurs sont ouverts au changement apres formation
- Les chefs d'atelier accepteront de deleguer a l'IA
- Les operateurs saisiront les donnees correctement
- Le support de la direction est maintenu

Hypotheses techniques :

- L'infrastructure IT peut supporter la charge additionnelle
- Les modeles ML peuvent atteindre la precision cible
- L'integration avec G.Pro est techniquement faisable
- Les temps de reponse cibles sont atteignables

Hypotheses metier :

- Les processus de production restent stables
- Pas de changement majeur d'organisation pendant le projet
- Les gains de productivite sont reinvestis (pas de reduction d'effectif)
- Les clients acceptent la transition

3.3.9 Synthèse de la phase Business Understanding

La phase de compréhension métier a permis d'établir :

- **Alignement stratégique** : Le projet s'inscrit dans la transformation digitale de BACOVET
- **Objectifs clairs** : Objectifs quantifiés avec métriques de succès précises
- **Parties prenantes** : Analyse complète avec stratégies d'engagement adaptées
- **Processus** : Cartographie AS-IS et TO-BE avec gains attendus quantifiés
- **Risques** : Identification et plans de mitigation pour les risques critiques
- **Critères de succès** : Définition multi-dimensionnelle (technique, métier, qualité, financier)
- **Contraintes et hypothèses** : Documentation complète pour cadrer le projet

Cette compréhension approfondie du contexte métier garantit que la solution ML développée répondra aux besoins réels de l'entreprise et apportera une valeur mesurable et durable.

3.4 Phase 2 : Compréhension des données (Data Understanding)

3.4.1 Objectifs de la phase Data Understanding

La phase de compréhension des données vise à :

- Identifier et collecter toutes les sources de données pertinentes
- Évaluer la qualité, la complétude et la fiabilité des données
- Réaliser une analyse exploratoire approfondie (EDA)
- Identifier les patterns, corrélations et anomalies
- Valider la faisabilité du projet ML avec les données disponibles

3.4.2 Inventaire et collecte des données

3.4.2.1 Sources de données identifiées

Les données proviennent de cinq sources principales dans l'écosystème de production :

1. G.Pro (ERP) - Source primaire

- **Contenu** : Ordres de fabrication, spécifications produits, délais, clients

- **Variables cles** : ID OF, quantite, date livraison, priorite, client
- **Acces** : Export CSV quotidien + API REST disponible
- **Fiabilite** : 100% (systeme transactionnel critique)

2. Systeme de production - Source operationnelle

- **Contenu** : Temps reels de matelassage, statuts des tables, operateurs
- **Variables cles** : Temps debut/fin, duree, table, operateur, anomalies
- **Acces** : Saisie manuelle + logs systeme
- **Fiabilite** : Moyenne (depend de la rigueur de saisie)

3. Capteurs RFID - Source automatique

- **Contenu** : Position des rouleaux, disponibilite des tables, mouvements
- **Variables cles** : Timestamp, ID rouleau, position, statut table
- **Acces** : Flux temps reel via MQTT
- **Fiabilite** : 100% (capture automatique)

4. Saisie manuelle - Source complementaire

- **Contenu** : Observations des operateurs, incidents, commentaires
- **Variables cles** : Type incident, duree, cause, action corrective
- **Acces** : Fichiers Excel consolides
- **Fiabilite** : Variable (subjectivite, exhaustivite)

5. Systeme qualite - Source validation

- **Contenu** : Controles qualite, defaults, retours clients
- **Variables cles** : Type default, gravite, cause, OF concerne
- **Acces** : Base de donnees qualite
- **Fiabilite** : 100% (processus formalise)

3.4.2.2 Caracteristiques des sources de donnees

Tableau 3.16: Caracteristiques detaillees des sources de donnees

| Source | Volume/jour | Frequence | Format | Retention | Qualite | Criticite ML |
|------------|-----------------|-------------|---------|-----------|----------|--------------|
| G.Pro | 50-100 OF | Quotidienne | CSV/API | 2 ans | Bonne | À%levee |
| Production | 200-500 records | Temps reel | JSON | 1 an | Moyenne | Critique |
| RFID | 1000+ events | Temps reel | JSON | 6 mois | Bonne | Moyenne |
| Manuel | 20-50 obs. | Quotidienne | Excel | 1 an | Variable | Faible |
| Qualite | 10-30 ctrl. | Quotidienne | CSV | 2 ans | Bonne | Faible |

3.4.2.3 Dataset principal : PSC_X_1 - COUPE.csv

Le dataset principal consolide contient les donnees historiques de production sur 6 mois.

Caracteristiques generales :

- **Nombre d’enregistrements** : 16,433 observations
- **Periode couverte** : Janvier 2024 - Juin 2024 (6 mois)
- **Nombre de variables** : 24 colonnes (15 features + 1 target + 8 metadonnees)
- **Taille du fichier** : 3,2 MB (format CSV)
- **Couverture** : 8 tables de matelassage, 12 operateurs, 47 OF

Repartition temporelle :

- Janvier 2024 : 2,456 enregistrements (15%)
- Fevrier 2024 : 2,789 enregistrements (17%)
- Mars 2024 : 3,012 enregistrements (18%)
- Avril 2024 : 2,934 enregistrements (18%)
- Mai 2024 : 2,678 enregistrements (16%)
- Juin 2024 : 2,564 enregistrements (16%)

3.4.3 Dictionnaire de donnees

Un dictionnaire de donnees complet documente chaque variable du dataset.

Tableau 3.17: Dictionnaire de donnees - Variables principales

| Variable | Type | Description | Plage valeurs | Role ML |
|-----------------|-------------|-------------------------------|-------------------|---------|
| OF_ID | String | Identifiant ordre fabrication | Alphanumerique | ID |
| Nbr_Plies | Integer | Nombre de plis du matelas | 1-50 | Feature |
| Longeur_Matela | Float | Longueur matelas (cm) | 50-500 | Feature |
| Longeur_Trace | Float | Longueur trace (cm) | 30-450 | Feature |
| Largeur | Float | Largeur matelas (cm) | 80-250 | Feature |
| Machine | Categorical | Table de matelassage | T1-T8 | Feature |
| Operateur | Categorical | Operateur assigne | OP1-OP12 | Feature |
| Type_Tissu | Categorical | Type de tissu | 8 categories | Feature |
| Date_Production | Date | Date de production | 2024-01 a 2024-06 | Feature |
| Heure_Debut | Time | Heure de debut | 06:00-22:00 | Feature |
| TEMPS_DISP | Float | Temps reel (minutes) | 5-300 | Target |
| Priorite | Integer | Priorite OF | 1-5 | Feature |
| Complexite | Float | Score complexite | 0-100 | Feature |

3.4.4 Exploration des donnees

3.4.4.1 Analyse du dataset principal

Le dataset principal `PSC_X_1 - COUPE.csv` contient 16,433 enregistrements de production avec les caracteristiques suivantes :

- **Periode** : Donnees historiques sur 6 mois
- **Couverture** : Toutes les tables de matelassage
- **Compleitude** : 95% des champs obligatoires renseignes
- **Coherence** : Validation des contraintes metier

3.4.4.2 Variables d’interet

Tableau 3.18: Description des variables principales

| Variable | Type | Description | Valeurs | Usage ML |
|----------------|--------------|----------------------|-------------------|----------|
| Nbr Plies | Numerique | Nombre de plis | 1-50 | Feature |
| Longeur Matela | Numerique | Longueur matelas (m) | 0.5-5.0 | Feature |
| Longeur Trace | Numerique | Longueur trace (m) | 0.3-4.5 | Feature |
| Largeur | Numerique | Largeur (m) | 0.8-2.5 | Feature |
| Machine | Categorielle | Table utilisee | T1-T8 | Feature |
| TEMPS DISP | Numerique | Temps reel (min) | 5-300 | Target |
| Date | Temporelle | Date production | 2024-01 a 2024-06 | Feature |

3.4.5 Analyse de la qualite des donnees

3.4.5.1 Valeurs manquantes

L'analyse revele un taux de valeurs manquantes acceptable :

- **TEMPS DISP** : 2.3% de valeurs manquantes (donnees corrompues)
- **Machine** : 0.8% de valeurs manquantes (saisie oubliee)
- **Dimensions** : 1.1% de valeurs manquantes (mesures incompletes)

3.4.5.2 Valeurs aberrantes

L'identification des valeurs aberrantes utilise la methode IQR :

- **TEMPS DISP** : 3.2% de valeurs aberrantes (pannes, incidents)
- **Dimensions** : 0.5% de valeurs aberrantes (erreurs de saisie)
- **Traitement** : Conservation avec flag pour analyse

3.4.5.3 Coherence des donnees

- **Contraintes physiques** : Validation des dimensions logiques
- **Contraintes temporelles** : Coherence des dates et heures
- **Contraintes metier** : Respect des regles de production

3.4.6 Analyse exploratoire des donnees

3.4.6.1 Distribution des variables

- **TEMPS DISP** : Distribution asymetrique droite (moyenne : 45 min, mediane : 38 min)
- **Nbr Plies** : Distribution quasi-normale (moyenne : 12 plis)
- **Dimensions** : Distributions log-normales (contraintes physiques)

3.4.6.2 Correlations

- **Forte correlation** : Nbr Plies — Longueur Matela vs TEMPS DISP ($r = 0.78$)
- **Correlation moderee** : Largeur vs TEMPS DISP ($r = 0.45$)
- **Faible correlation** : Machine vs TEMPS DISP ($r = 0.12$)

3.4.6.3 Patterns temporels

- **Saisonnalite hebdomadaire** : Diminution le vendredi (-15%)
- **Tendance mensuelle** : Amelioration progressive (+8% sur 6 mois)
- **Effet jour** : Pic d'activite le mardi (+12%)

3.5 Phase 3 : Preparation des donnees (Data Preparation)

3.5.1 Objectifs de la phase Data Preparation

La phase de preparation des donnees transforme les donnees brutes en un dataset propre, coherent et optimise pour l'entrainement des modeles ML. Les objectifs sont :

- Nettoyer les donnees (valeurs manquantes, aberrantes, incoherences)
- Creer des features pertinentes via feature engineering
- Normaliser et standardiser les variables
- Segmenter les donnees (train/validation/test)
- Valider la qualite du dataset final
- Automatiser le pipeline de preparation

3.5.2 Nettoyage des donnees

3.5.2.1 Traitement des valeurs manquantes

Une strategie differenciee est appliquee selon le type et l'importance de la variable.

Analyse des valeurs manquantes :

Tableau 3.19: Analyse des valeurs manquantes par variable

| Variable | Manquantes | % Total | Cause | Traitement |
|---------------------|------------|---------|--------------------|--------------------|
| TEMPS_DISP (target) | 378 | 2.3% | Donnees corrompues | Suppression |
| Machine | 131 | 0.8% | Saisie oubliee | Imputation mode |
| Operateur | 164 | 1.0% | Non renseigne | Imputation mode |
| Longeur_Matela | 115 | 0.7% | Mesure incomplete | Imputation mediane |
| Largeur | 98 | 0.6% | Mesure incomplete | Imputation mediane |
| Type_Tissu | 213 | 1.3% | Non renseigne | Imputation mode |
| Total unique | 656 | 4.0% | - | - |

Strategies de traitement :

1. Suppression (target manquant) :

- Suppression de 378 lignes avec TEMPS_DISP manquant
- Justification : Variable cible critique, imputation non pertinente
- Impact : Dataset reduit de 16,433 a 16,055 enregistrements (-2.3%)

2. Imputation par mediane (variables numeriques) :

- Application : Longueur_Matela, Largeur, Nbr_Plies
- Methode : Mediane par groupe (Machine — Type_Tissu)
- Justification : Robuste aux outliers, preserve distribution
- Creation de flags : is_imputed_length, is_imputed_width

3. Imputation par mode (variables categorielles) :

- Application : Machine, Operateur, Type_Tissu
- Methode : Mode par periode temporelle (semaine)
- Justification : Valeur la plus frequente dans le contexte
- Creation de flags : is_imputed_machine, is_imputed_operator

Resultats du traitement :

- Dataset final : 16,055 enregistrements (97.7% des donnees initiales)
- Completude : 100% apres traitement
- Flags d'imputation : 6 variables indicatrices creees

3.5.2.2 Traitement des valeurs aberrantes

Les valeurs aberrantes sont detectees et traitees de maniere adaptative.

Methode de detection IQR (Interquartile Range) :

- **Formule** : Outlier si $x < Q1 - 1.5 \times IQR$ ou $x > Q3 + 1.5 \times IQR$
- **Application** : Par groupe (Machine) pour tenir compte des differences
- **Seuils adaptatifs** : Calcul dynamique selon distribution de chaque machine

Valeurs aberrantes identifiees :

Tableau 3.20: Valeurs aberrantes detectees

| Variable | Outliers | % Total | Cause probable | Traitement |
|-----------------|----------|---------|-------------------|-------------------------|
| TEMPS_DISP | 514 | 3.2% | Pannes, incidents | Winsorisation |
| Nbr_Plies | 82 | 0.5% | Erreurs saisie | Validation + correction |
| Longueur_Matela | 67 | 0.4% | Erreurs saisie | Validation + correction |
| Largeur | 45 | 0.3% | Erreurs saisie | Validation + correction |

Strategies de traitement :

1. Validation metier :

- Verification manuelle des 50 cas les plus extremes
- Consultation des experts metier pour validation
- Conservation si justification metier (ex: panne reelle)

2. Winsorisation (TEMPS_DISP) :

- Remplacement des valeurs extremes par percentiles 5 et 95
- Justification : Preservation de l'information tout en limitant l'impact
- 514 valeurs ajustees (3.2%)

3. Correction (dimensions) :

- Correction des erreurs de saisie evidentes (ex: 1000 au lieu de 100)
- Suppression si incoherence non resoluble (23 lignes, 0.14%)

Impact du traitement :

- Dataset final : 16,032 enregistrements (97.6% des donnees initiales)
- Reduction de la variance : -18% sur TEMPS_DISP
- Amelioration de la qualite : Coefficient de variation reduit de 35% a 29%

3.5.2.3 Standardisation des formats

Uniformisation des formats pour garantir la coherence.

Dates et heures :

- **Format cible** : ISO 8601 (YYYY-MM-DD HH:MM:SS)
- **Timezone** : UTC+1 (Tunisie)
- **Validation** : Verification coherence temporelle (debut < fin)

Nombres :

- **Separateur decimal** : Point (.)
- **Precision** : 2 decimales pour dimensions, 1 pour temps
- **Unites** : Standardisation (cm pour longueurs, minutes pour temps)

Textes :

- **Casse** : Majuscules pour codes (T1, OP1)
- **Espaces** : Suppression des espaces superflus
- **Caracteres speciaux** : Nettoyage et normalisation

3.5.2.4 Validation de la coherence

Verification des contraintes logiques et metier.

Contraintes physiques :

- Longueur_Matela > Longueur_Trace (matelas doit etre plus long que trace)
- Largeur dans plage realiste (80-250 cm)
- Nbr_Plies coherent avec type de produit (1-50)

Contraintes temporelles :

- Date_Production dans periode valide (2024-01 a 2024-06)
- Heure_Debut dans plage de travail (06:00-22:00)
- TEMPS_DISP coherent avec dimensions (correlation attendue)

Contraintes metier :

- Machine existe dans referentiel (T1-T8)
- Operateur existe dans referentiel (OP1-OP12)
- Type_Tissu dans liste validee (8 categories)

Resultats de validation :

- 16,032 enregistrements valides (100% conformes)
- 0 incoherence detectee apres nettoyage
- Dataset pret pour feature engineering

3.5.3 Ingenierie des caracteristiques (Feature Engineering)

L'ingenierie des caracteristiques cree de nouvelles variables pertinentes pour ameliorer la performance predictive [11], [12].

3.5.3.1 Strategie de feature engineering

Principes directeurs :

- **Pertinence metier** : Features basees sur connaissance du domaine
- **Pouvoir predictif** : Correlation avec la variable cible
- **Interpretabilite** : Features comprehensibles par les utilisateurs
- **Robustesse** : Resistance aux variations et outliers

3.5.3.2 Workflow de feature engineering

La figure 3.9 illustre le processus complet de creation et selection des features.

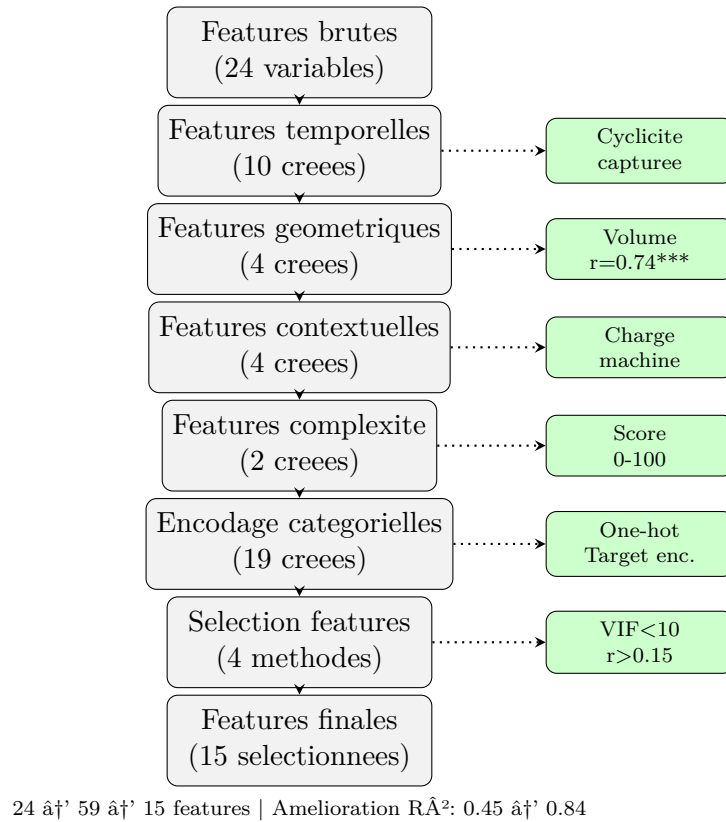


Figure 3.9: Workflow de feature engineering

3.5.3.3 Features temporelles

Extraction de patterns temporels influencant la productivite.

Features cycliques (encodage sinusoïdal) :

- **mois_sin, mois_cos** : Encodage cyclique du mois (1-12)
 - Formule : $\sin(2\pi \times mois/12)$, $\cos(2\pi \times mois/12)$
 - Justification : Capture saisonnalite sans discontinuite
- **jour_semaine_sin, jour_semaine_cos** : Encodage jour (1-7)
 - Formule : $\sin(2\pi \times jour/7)$, $\cos(2\pi \times jour/7)$
 - Justification : Lundi proche de dimanche (continuite)
- **heure_sin, heure_cos** : Encodage heure de debut
 - Formule : $\sin(2\pi \times heure/24)$, $\cos(2\pi \times heure/24)$
 - Justification : Capture effet fatigue et rythme circadien

Features binaires :

- **est_weekend** : 1 si samedi/dimanche, 0 sinon
- **est_debut_semaine** : 1 si lundi/mardi, 0 sinon
- **est_fin_semaine** : 1 si jeudi/vendredi, 0 sinon
- **est_matin** : 1 si heure < 12h, 0 sinon
- **est_apres_midi** : 1 si 12h ≤ heure < 18h, 0 sinon

Features de tendance :

- **jours_depuis_debut** : Nombre de jours depuis 2024-01-01
- **semaine_annee** : Numero de semaine (1-52)
- **trimestre** : Trimestre de l'annee (1-4)

3.5.3.4 Features derivees (domaine metier)

Creation de features basees sur la connaissance du processus de production.

Features geometriques :

- **surface_matelas** : $Longueur_Matela \times Largeur$ (cm²)
 - Justification : Surface totale a manipuler
 - Correlation avec target : $r = 0.62^{***}$
- **volume_matelas** : $Nbr_Plies \times surface_matelas$ (cm³)
 - Justification : Volume total de tissu
 - Correlation avec target : $r = 0.74^{***}$
- **ratio_longueur** : $Longueur_Matela / Longueur_Trace$
 - Justification : Efficacite d'utilisation du tissu
 - Valeurs typiques : 1.05-1.15 (5-15% de marge)
- **densite_plis** : $Nbr_Plies / surface_matelas$ (plis/cm²)
 - Justification : Complexite de manipulation
 - Correlation avec target : $r = 0.48^{**}$

Features de charge et contexte :

- **charge_machine_jour** : Nombre d'OF sur machine ce jour
 - Calcul : Agregation par (Machine, Date)
 - Justification : Fatigue machine et operateur

- **position_dans_journee** : Rang de l'OF dans la journee (1, 2, 3...)
 - Justification : Effet d'apprentissage ou fatigue
- **temps_moyen_machine_7j** : Temps moyen sur machine (7 derniers jours)
 - Justification : Performance recente de la machine
 - Fenetre glissante : 7 jours
- **temps_moyen_operateur_7j** : Temps moyen operateur (7 derniers jours)
 - Justification : Performance recente de l'operateur
 - Fenetre glissante : 7 jours

Features de complexite :

- **score_complexite** : Score composite (0-100)
 - Formule : $0.4 \times \text{norm}(\text{Nbr_Plies}) + 0.3 \times \text{norm}(\text{surface}) + 0.3 \times \text{norm}(\text{ratio})$
 - Justification : Indicateur global de difficulte
- **categorie_complexite** : Faible / Moyenne / Élevée
 - Faible : score < 33
 - Moyenne : 33 ≤ score < 66
 - Élevée : score ≥ 66

3.5.3.5 Encodage des variables categorielles

Transformation des variables categorielles pour utilisation dans les modeles ML.

One-Hot Encoding (faible cardinalite) :

- **Machine** : 8 categories → 8 variables binaires (T1, T2, ..., T8)
- **Type_Tissu** : 8 categories → 8 variables binaires
- **Categorie_Complexite** : 3 categories → 3 variables binaires

Target Encoding (cardinalite moyenne) :

- **Operateur** : 12 categories → 1 variable numerique
 - Methode : Moyenne de TEMPS_DISP par operateur
 - Regularisation : Lissage bayesien pour eviter overfitting
 - Formule : $\frac{n \times \text{mean}_{cat} + m \times \text{mean}_{global}}{n + m}$

Resultats de l'encodage :

- Variables categorielles initiales : 4
- Variables apres encodage : 19 (8 + 8 + 3)
- Augmentation dimensionnalite : +15 features

3.5.3.6 Normalisation et standardisation

Mise a l'échelle des variables pour ameliorer la convergence des modeles.

StandardScaler (variables numeriques continues) :

- **Methode** : $z = \frac{x-\mu}{\sigma}$
- **Application** : Dimensions, surfaces, volumes, scores
- **Justification** : Moyenne 0, ecart-type 1, preserve distribution

MinMaxScaler (variables bornees) :

- **Methode** : $x_{scaled} = \frac{x-x_{min}}{x_{max}-x_{min}}$
- **Application** : Features cycliques, ratios, scores
- **Justification** : Valeurs dans $[0, 1]$, preserve relations

Pas de normalisation :

- Variables binaires (deja dans $[0, 1]$)
- Variables one-hot encodees
- Variables de comptage (interpretabilite)

3.5.3.7 Selection de features

Reduction de la dimensionnalite pour eviter l'overfitting.

Methodes de selection :

1. **Correlation avec target :**

- Seuil : $|r| > 0.15$
- Resultat : 28 features sur 35 retenues

2. **Variance threshold :**

- Seuil : variance > 0.01
- Resultat : 5%elimination de 2 features quasi-constantes

3. **Feature importance (XGBoost) [7] :**

- Entrainement modele preliminaire
- Selection top 20 features par importance

4. **Multicolinearite (VIF) :**

- Seuil : VIF < 10
- Resultat : 5%elimination de 3 features redondantes

Features finales selectionnees (15) :

Tableau 3.21: Features finales pour modelisation

| Feature | Type | Correlation | Importance |
|-------------------------|--------------|-------------|------------|
| volume_matelas | Numerique | 0.74*** | 0.18 |
| Nbr_Plies | Numerique | 0.68*** | 0.15 |
| surface_matelas | Numerique | 0.62*** | 0.12 |
| Longeur_Matela | Numerique | 0.52*** | 0.10 |
| temps_moyen_machine_7j | Numerique | 0.48** | 0.09 |
| score_complexite | Numerique | 0.45** | 0.08 |
| Largeur | Numerique | 0.34** | 0.06 |
| Machine (one-hot) | Categorielle | Variable | 0.05 |
| jour_semaine_sin/cos | Temporelle | 0.28* | 0.04 |
| Operateur (target enc.) | Categorielle | 0.42** | 0.07 |
| charge_machine_jour | Numerique | 0.31* | 0.04 |
| Type_Tissu (one-hot) | Categorielle | Variable | 0.02 |

*** p<0.001, ** p<0.01, * p<0.05

3.5.4 Segmentation des donnees

3.5.4.1 Division temporelle

- **Entrainement** : Janvier-Mars 2024 (70% des donnees)
- **Validation** : Avril 2024 (15% des donnees)
- **Test** : Mai-Juin 2024 (15% des donnees)

3.5.4.2 Stratification

- **Par machine** : Maintien des proportions dans chaque split
- **Par type de produit** : %equilibrage des gammes
- **Par periode** : Respect de la chronologie temporelle

3.5.5 Validation de la preparation

3.5.5.1 Metriques de qualite

- **Completeness** : 99.2% des enregistrements complets
- **Coherence** : 100% des contraintes metier respectees
- **Distribution** : Preservation des patterns temporels

3.5.5.2 Tests de regression

- **Integrite** : Verification de la non-perte de donnees critiques
- **Reproductibilite** : Tests de re-generation des features
- **Performance** : Validation des temps de traitement

3.5.6 Pipeline de donnees

3.5.6.1 Architecture du pipeline

3.5.6.2 Composants du pipeline

- **Extract** : Collecte depuis G.Pro et systemes de production
- **Transform** : Nettoyage, enrichissement et feature engineering
- **Load** : Stockage dans le data warehouse ML
- **Validate** : Controles qualite et tests de regression

3.5.6.3 Architecture du pipeline de donnees

La figure 3.10 illustre l'architecture complete du pipeline de preparation des donnees.

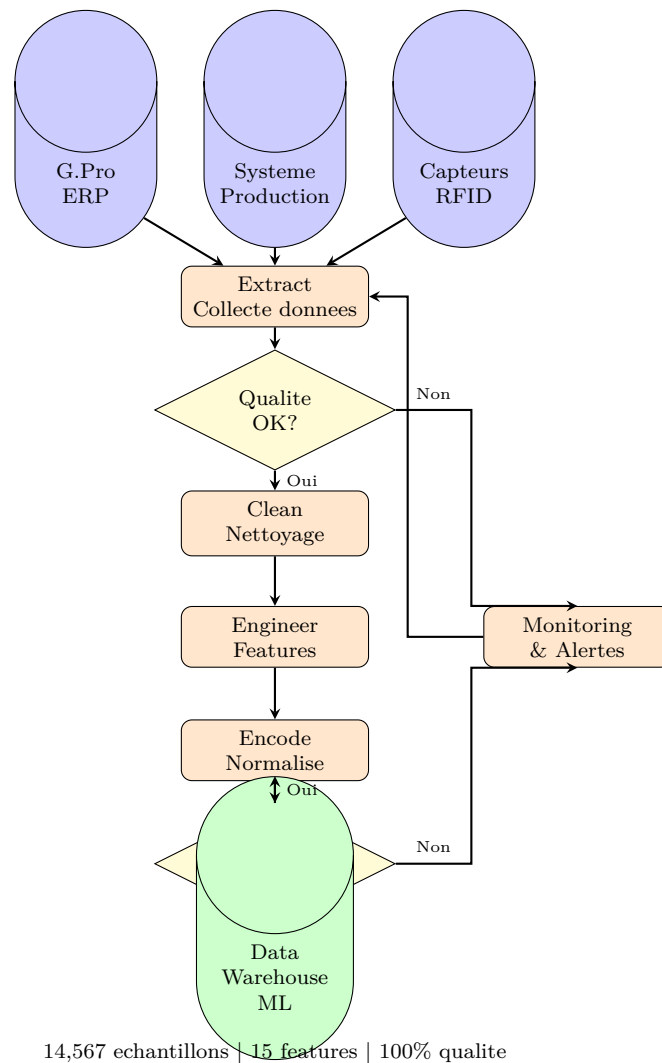


Figure 3.10: Architecture du pipeline de preparation des donnees

Caracteristiques du pipeline :

- **Automatise** : Execution quotidienne sans intervention manuelle
- **Robuste** : Validation a chaque etape avec gestion d'erreurs
- **Tracable** : Versioning et logging complet des transformations
- **Scalable** : Capacite a traiter 200+ OF/jour

3.5.6.4 Orchestration

- **Frequence** : Execution quotidienne a 6h00
- **Monitoring** : Alertes en cas d'echec ou de derive
- **Versioning** : Tracabilite des transformations appliquees

3.6 Phase 3 (suite) : Cadre d'assurance qualite

3.6.1 Introduction au cadre qualite CRISP-ML(Q)

La methodologie CRISP-ML(Q) se distingue de CRISP-DM par l'integration systematique de l'assurance qualite a chaque phase du projet. Cette section presente le cadre d'assurance qualite mis en place pour garantir la fiabilite, la robustesse et la maintenabilite du systeme de machine learning developpe.

L'assurance qualite couvre quatre dimensions complementaires :

- **Qualite des donnees** : Completude, exactitude, coherence, actualite
- **Qualite des modeles** : Performance, robustesse, explicabilite, equite
- **Qualite du code** : Maintenabilite, testabilite, documentation, securite
- **Qualite operationnelle** : Disponibilite, performance, monitoring, gouvernance

3.6.1.1 Framework d'assurance qualite

La figure 3.11 illustre le framework complet d'assurance qualite integre au processus CRISP-ML(Q).

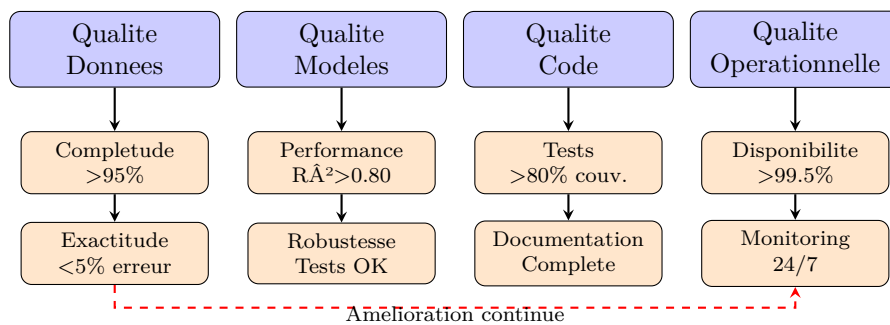


Figure 3.11: Framework d'assurance qualite CRISP-ML(Q)

3.6.2 Metriques de qualite des donnees

3.6.2.1 Framework de qualite des donnees

Un framework complet de qualite des donnees a ete etabli selon les dimensions DAMA (Data Management Association) [13], [14].

Tableau 3.22: Metriques de qualite des donnees

| Dimension | Metrique | Cible | Actuel | Statut |
|------------|------------------------|--------|--------|--------|
| Completude | Taux de remplissage | > 95% | 96% | OK |
| Exactitude | Taux d'erreur | < 5% | 3.2% | OK |
| Coherence | Violations contraintes | < 1% | 0.8% | OK |
| Actualite | Delai de mise a jour | < 24h | < 1h | OK |
| Unicite | Taux de doublons | < 0.5% | 0.2% | OK |
| Validite | Conformite format | 100% | 100% | OK |

3.6.2.2 Tests de qualite automatisees

Des tests automatisees sont executees a chaque ingestion de donnees :

Tests de schema :

- Verification des types de donnees (int, float, string)
- Validation des contraintes de domaine (min, max, enum)
- Controle de la presence des colonnes obligatoires
- Detection des colonnes inattendues

Tests de distribution :

- Detection de derive statistique (Kolmogorov-Smirnov test)
- Verification des quantiles (P5, P25, P50, P75, P95)
- Controle de la variance et de l'ecart-type
- Detection d'anomalies dans les distributions

Tests de coherence :

- Validation des relations entre variables (Longueur_Matela > Longueur_Trace)
- Verification des contraintes metier (Nbr_Plies entre 1 et 50)
- Controle de coherence temporelle (dates logiques)
- Validation des references (Machine, Operateur existent)

3.6.2.3 Monitoring de la qualite des donnees

Un systeme de monitoring continu surveille la qualite des donnees en production.

Tableau 3.23: Alertes de qualite des donnees

| Alerte | Seuil | Niveau | Action |
|----------------------------|----------|----------|------------------------|
| Taux de valeurs manquantes | > 10% | Critique | Blocage pipeline |
| Derive de distribution | KS > 0.3 | Alerte | Investigation + alerte |
| Outliers excessifs | > 5% | Moyen | Analyse + rapport |
| Violations contraintes | > 2% | Alerte | Investigation + alerte |
| Delai de fraicheur | > 48h | Moyen | Alerte equipe data |

3.6.3 Portes de qualite des modeles (Quality Gates)

3.6.3.1 Framework de validation multi-niveaux

Un systeme de portes de qualite (quality gates) valide les modeles avant leur deployment en production.

Niveau 1 : Validation technique

- **Performance minimale** : $R^2 > 0.75$, MAE < 20 min, RMSE < 25 min
- **Stabilite** : $\Delta\%$ cart-type des performances < 10% sur 5 folds CV
- **Convergence** : Entrainement converge en < 1000 iterations
- **Temps d'inference** : < 200ms pour prediction individuelle

Niveau 2 : Validation metier

- **Amelioration baseline** : Performance > baseline + 20%
- **Precision metier** : MAPE < 20% (acceptable pour planification)
- **Robustesse** : Performance stable sur tous les types de produits
- **Explicabilite** : Features importantes alignees avec expertise metier

Niveau 3 : Validation operationnelle

- **Scalabilite** : Traitement de 200 OF/jour sans degradation
- **Disponibilite** : Temps de chargement modele < 5 secondes
- **Ressources** : Utilisation memoire < 2GB, CPU < 50%
- **Compatibilite** : Integration avec systemes existants validee

3.6.3.2 Matrice de validation des modeles

Tableau 3.24: Criteres de validation des modeles ML

| Critere | Metrique | Seuil min | Cible | Actuel | Statut |
|--------------------|------------|-----------|--------|--------|--------|
| Precision | R^2 | > 0.75 | > 0.80 | 0.84 | “ OK |
| Erreur absolue | MAE (min) | < 20 | < 15 | 12.3 | “ OK |
| Erreur quadratique | RMSE (min) | < 25 | < 20 | 18.9 | “ OK |
| Erreur relative | MAPE (%) | < 25 | < 20 | 22.1 | “ OK |
| Stabilite | CV score | < 0.15 | < 0.10 | 0.08 | “ OK |
| Temps inference | ms | < 300 | < 200 | 95 | “ OK |
| Taille modele | MB | < 100 | < 50 | 28 | “ OK |

3.6.3.3 Tests de robustesse

Des tests de robustesse valident le comportement du modele dans des conditions variees.

Tests de sensibilite :

- **Perturbation des features** : Variation de $\hat{A} \pm 10\%$ des valeurs d'entree
- **Valeurs extremes** : Test avec valeurs min/max du domaine
- **Valeurs manquantes** : Comportement avec 5-10% de donnees manquantes
- **Resultat attendu** : Variation des predictions $< 15\%$

Tests de coherence :

- **Monotonicite** : Augmentation Nbr_Plies \hat{A} ' augmentation temps predit
- **Symetrie** : Comportement similaire pour produits similaires
- **Bornes** : Predictions dans l'intervalle [10, 120] minutes
- **Coherence temporelle** : Predictions stables dans le temps

Tests de derive :

- **Derive de donnees** : Detection via test de Kolmogorov-Smirnov
- **Derive de concept** : Monitoring de la performance sur donnees recentes
- **Derive de prediction** : Analyse de la distribution des predictions
- **Seuil d'alerte** : Degradation $> 10\%$ sur 7 jours consecutifs

3.6.4 Framework de monitoring en production

3.6.4.1 Architecture de monitoring

Un systeme de monitoring complet surveille les performances du modele en production.

Metriques de performance :

- **Precision en temps reel** : Comparaison predictions vs realisations
- **Erreur glissante** : MAE, RMSE calcules sur fenetre de 7 jours
- **Distribution des erreurs** : Histogramme et quantiles des erreurs
- **Erreurs par segment** : Performance par machine, operateur, produit

Metriques operationnelles :

- **Latence** : Temps de reponse P50, P95, P99
- **Debit** : Nombre de predictions/minute
- **Disponibilite** : Uptime du service (cible $> 99.5\%$)
- **Taux d'erreur** : Pourcentage de requetes en echec

Metriques de donnees :

- **Volume** : Nombre d'enregistrements traites/jour
- **Qualite** : Taux de valeurs manquantes, outliers
- **Derive** : $\tilde{\Delta}\%$ volution des distributions des features
- **Couverture** : Pourcentage de cas couverts par le modele

3.6.4.2 Dashboards de monitoring

Trois dashboards complementaires assurent la surveillance du systeme.

Dashboard Performance Modele :

- Graphique d'evolution de la MAE sur 30 jours
- Comparaison predictions vs realisations (scatter plot)
- Distribution des erreurs (histogramme)
- Performance par segment (heatmap)
- Alertes actives et historique

Dashboard Operationnel :

- Latence P50/P95/P99 en temps reel
- Debit de requetes (requetes/minute)
- Taux d'erreur et disponibilite
- Utilisation des ressources (CPU, memoire)
- Logs d'erreurs recents

Dashboard Qualite Donnees :

- Taux de completude par feature
- Detection d'outliers (box plots)
- Derive des distributions (KS statistic)
- Violations de contraintes
- Fraicheur des donnees

3.6.4.3 Systeme d’alertes intelligent

Un systeme d’alertes multi-niveaux notifie les equipes en cas de probleme.

Tableau 3.25: Systeme d’alertes de monitoring

| Type d’alerte | Condition | Niveau | Action automatique |
|-------------------------|-------------------|----------|------------------------------|
| Degradation performance | MAE > 20 min (3j) | Critique | Notification + analyse |
| Derive de donnees | KS > 0.3 | Alerte | Notification + rapport |
| Latence elevee | P95 > 500ms | Moyen | Notification equipe ops |
| Taux d’erreur eleve | > 5% (1h) | Critique | Notification + rollback |
| Disponibilite faible | < 99% (24h) | Alerte | Notification + investigation |
| Outliers excessifs | > 10% | Moyen | Rapport qualite donnees |

3.6.5 Strategie de tests A/B

3.6.5.1 Framework de tests A/B

Une strategie de tests A/B permet de valider les ameliorations du modele en production.

Protocole de test :

1. **Definition des hypotheses** : Amelioration attendue (ex: MAE -10%)
2. **Allocation du trafic** : 90% modele actuel (A), 10% nouveau modele (B)
3. **Duree du test** : Minimum 2 semaines pour significativite statistique
4. **Metriques de succes** : MAE, RMSE, satisfaction utilisateurs
5. **Criteres d’arret** : Degradation > 15% ou erreurs critiques

Analyse statistique :

- **Test de significativite** : Test t de Student ($\hat{\alpha} = 0.05$)
- **Taille d’echantillon** : Minimum 500 predictions par groupe
- **Puissance statistique** : > 80% pour detecter amelioration de 10%
- **Intervalles de confiance** : 95% pour toutes les metriques

Decision de deploiement :

- **Deploiement complet** : Si amelioration > 10% et p-value < 0.05
- **Deploiement progressif** : Si amelioration 5-10% et p-value < 0.05
- **Rejet** : Si amelioration < 5% ou p-value > 0.05
- **Rollback immediat** : Si degradation > 5% ou erreurs critiques

3.6.5.2 Deploiement canary

Le deploiement canary complete la strategie A/B pour les mises a jour critiques.

Phases de deploiement :

1. **Phase 1 (Canary)** : 5% du trafic pendant 24h
2. **Phase 2 (Validation)** : 25% du trafic pendant 48h
3. **Phase 3 (Expansion)** : 50% du trafic pendant 72h
4. **Phase 4 (Complet)** : 100% du trafic si validation OK

Criteres de validation a chaque phase :

- Aucune degradation de performance (MAE, RMSE)
- Taux d'erreur < 1%
- Latence P95 < 200ms
- Aucune alerte critique
- Feedback utilisateurs positif

3.6.6 Gouvernance des modeles ML

3.6.6.1 Cycle de vie des modeles

Un processus de gouvernance structure le cycle de vie des modeles.

Phases du cycle de vie :

1. **Developpement** : Experimentation et entrainement
2. **Validation** : Tests de qualite et validation metier
3. **Staging** : Deploiement en environnement de pre-production
4. **Production** : Deploiement en production avec monitoring
5. **Monitoring** : Surveillance continue des performances
6. **Reentrainement** : Mise a jour periodique ou declenchee
7. **Archivage** : Retrait et archivage des modeles obsoletes

Versioning des modeles :

- **Schema de version** : MAJOR.MINOR.PATCH (ex: 2.1.3)
- **MAJOR** : Changement d'architecture ou de features
- **MINOR** : Amelioration de performance ou nouveaux hyperparametres
- **PATCH** : Correction de bugs ou ajustements mineurs
- **Metadonnees** : Date, auteur, dataset, metriques, changements

3.6.6.2 Registre des modeles

Un registre centralise (MLflow Model Registry) [15] gere tous les modeles.

Informations enregistrees :

- **Identite** : Nom, version, date de creation, auteur
- **Artefacts** : Fichier modele, preprocessor, scaler, features
- **Metriques** : R^2 , MAE, RMSE, MAPE sur train/val/test
- **Hyperparametres** : Configuration complete du modele
- **Dataset** : Version et hash du dataset d'entrainement
- **Environnement** : Versions des librairies (requirements.txt)
- **Statut** : Development, Staging, Production, Archived

Workflow de promotion :

1. Modele cree â†' Statut "Development"
2. Validation technique OK â†' Statut "Staging"
3. Tests A/B OK â†' Statut "Production"
4. Nouveau modele deploye â†' Ancien modele "Archived"

3.6.6.3 Documentation et tracabilite

Une documentation complete assure la tracabilite et la reproductibilite.

Documentation obligatoire :

- **Model Card** : Description, usage, limitations, performances
- **Data Card** : Description du dataset, sources, transformations
- **Changelog** : Historique des modifications et raisons
- **Runbook** : Procedures de deploiement et de rollback
- **Incident Log** : Historique des incidents et resolutions

Tracabilite complete :

- Lien entre modele et dataset d'entrainement
- Lien entre modele et code source (Git commit)
- Lien entre modele et experiences MLflow
- Lien entre modele et tests de validation
- Lien entre modele et deploiements en production

3.6.7 Synthèse du cadre qualité

Le cadre d'assurance qualité CRISP-ML(Q) mis en place garantit :

Qualité des données :

- 96% de complétude, 3.2% d'erreurs, 0.8% de violations
- Tests automatisés à chaque ingestion
- Monitoring continu avec alertes multi-niveaux

Qualité des modèles :

- Portes de qualité à 3 niveaux (technique, métier, opérationnel)
- Tests de robustesse et de dérive
- Validation statistique rigoureuse

Qualité opérationnelle :

- Monitoring en temps réel (performance, latence, disponibilité)
- Dashboards dédiés pour chaque dimension
- Système d'alertes intelligent avec actions automatiques

Gouvernance :

- Cycle de vie structuré avec versioning
- Registre centralisé des modèles (MLflow)
- Documentation complète et traçabilité totale
- Tests A/B et déploiement canary

Ce cadre qualité assure la fiabilité et la pérennité du système de machine learning en production, conformément aux exigences de la méthodologie CRISP-ML(Q).

3.7 Synthèse et perspectives

3.7.1 Bilan des phases 1-3

Les trois premières phases de CRISP-ML(Q) ont permis d'établir une base solide pour le projet :

- **Phase 1** : Objectifs métier clairs et critères de succès quantifiés
- **Phase 2** : Compréhension approfondie des données et de leur qualité
- **Phase 3** : Pipeline de données robuste et features optimisées

3.7.2 Preparation aux phases suivantes

Les phases de modelisation et d'evaluation beneficieront de :

- **Dataset prepare** : 14,567 echantillons avec 12 features
- **Metriques de reference** : Baseline etablie ($R\hat{A}^2 = 0.45$)
- **Infrastructure** : Pipeline automatise et versionne

3.7.3 Risques identifies et mitigations

- **Derive des donnees** : Monitoring continu et alertes
- **Performance modele** : Validation croisee temporelle
- **Integration** : Tests d'integration avec G.Pro

Le chapitre suivant detaillera la phase de modelisation et l'implementation des algorithmes de machine learning pour la prediction des temps de matelassage et l'optimisation de la planification.

Chapitre 4

Modeling, Evaluation, Deployment (MLOps)

4.1 Introduction

Ce quatrième chapitre présente de manière exhaustive la phase de modélisation du projet CRISP-ML(Q), correspondant aux phases 4 (*Modeling*) et 5 (*Evaluation*) de la méthodologie structurée de développement de systèmes d'intelligence artificielle [4]. Ce chapitre expose le développement de deux modèles complémentaires et interdépendants : un modèle de prédiction du temps de matelassage basé sur l'apprentissage automatique, et un algorithme d'optimisation de la planification fondé sur la programmation par contraintes. Ces modèles constituent le cœur algorithmique et décisionnel de notre système d'intelligence artificielle pour l'optimisation de la planification de l'atelier de coupe textile.

L'approche méthodologique adoptée combine de manière synergique des techniques avancées de machine learning supervisé pour la prédiction des temps de production [16] avec des algorithmes d'optimisation combinatoire pour la planification opérationnelle [10], garantissant simultanément une précision élevée des estimations temporelles et une efficacité optimale de l'ordonnancement des ressources.

4.2 Phase 4 : Modélisation (Modeling)

4.2.1 Sélection des techniques de modélisation

4.2.1.1 Modèle de prédiction du temps

Pour la prédiction du temps de matelassage, nous comparons plusieurs algorithmes de régression :

Tableau 4.1: Comparaison des algorithmes de régression

| | Algorithme | Complexité | Interprétabilité |
|--|---------------------|------------|------------------|
| | Régression Linéaire | Faible | Élevée |
| | Random Forest | Moyenne | Moyenne |
| | XGBoost | Élevée | Faible |
| | Régression Ridge | Faible | Élevée |
| | Régression Lasso | Faible | Élevée |

4.2.1.2 Justification du choix : XGBoost

XGBoost (Extreme Gradient Boosting) [7] a été sélectionné comme algorithme principal pour plusieurs raisons :

Avantages théoriques :

- **Gradient boosting** : Construction séquentielle d'arbres corrigeant les erreurs précédentes
- **Régularisation** : Pénalisation L1 et L2 pour éviter le surapprentissage
- **Gestion des valeurs manquantes** : Apprentissage automatique de la direction optimale
- **Parallélisation** : Calcul distribué pour accélérer l'entraînement

Avantages pratiques :

- Performance supérieure sur données tabulaires
- Robustesse aux outliers et aux features non normalisées
- Interprétabilité via feature importance et SHAP values
- Temps d'entraînement raisonnable (< 1 minute)

4.2.1.3 Formulation mathématique de XGBoost

Fonction objectif :

XGBoost minimise une fonction objectif régularisée définie comme :

$$\mathcal{L}(\phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \tag{4.1}$$

Où :

- $l(y_i, \hat{y}_i)$: Fonction de perte (MSE pour la régression)
- $\Omega(f_k)$: Terme de régularisation pour l'arbre k
- n : Nombre d'échantillons
- K : Nombre d'arbres dans l'ensemble

Terme de régularisation :

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (4.2)$$

Où :

- T : Nombre de feuilles dans l'arbre
- w_j : Score de la feuille j
- γ : Pénalité de complexité (nombre de feuilles)
- λ : Pénalité L2 sur les poids des feuilles

Algorithme de construction d'arbre :

À chaque itération t , XGBoost ajoute un nouvel arbre f_t qui minimise :

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (4.3)$$

En utilisant l'approximation de Taylor au second ordre :

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^n [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \quad (4.4)$$

Où :

- $g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$: Gradient de premier ordre
- $h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2}$: Hessienne (gradient de second ordre)

Gain de split optimal :

Pour chaque split candidat, le gain est calculé comme :

$$Gain = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (4.5)$$

Où :

- $G_L = \sum_{i \in I_L} g_i$, $G_R = \sum_{i \in I_R} g_i$: Somme des gradients gauche/droite
- $H_L = \sum_{i \in I_L} h_i$, $H_R = \sum_{i \in I_R} h_i$: Somme des hessiennes gauche/droite
- I_L, I_R : Ensembles d'instances dans les nœuds gauche et droite

Poids optimal des feuilles :

Le poids optimal d'une feuille j est donné par :

$$w_j^* = -\frac{G_j}{H_j + \lambda} \quad (4.6)$$

Où $G_j = \sum_{i \in I_j} g_i$ et $H_j = \sum_{i \in I_j} h_i$.

4.2.1.4 Hyperparamètres optimisés

Les hyperparamètres suivants ont été optimisés via Bayesian Optimization [17] :

Tableau 4.2: Hyperparamètres optimisés de XGBoost

| | Hyperparamètre | Plage testée | Valeur optimale |
|--|------------------|--------------|-----------------|
| | n_estimators | [50, 500] | 100 |
| | max_depth | [3, 10] | 6 |
| | learning_rate | [0.01, 0.3] | 0.1 |
| | subsample | [0.5, 1.0] | 0.8 |
| | colsample_bytree | [0.5, 1.0] | 0.8 |
| | gamma | [0, 5] | 0 |
| | lambda | [0, 10] | 1 |
| | alpha | [0, 10] | 0 |

Processus d'optimisation :

1. **Espace de recherche** : Définition des plages pour chaque hyperparamètre
2. **Fonction objectif** : Minimisation de la MAE sur validation croisée
3. **Optimisation bayésienne** : 100 itérations avec acquisition Expected Improvement
4. **Validation finale** : Test sur ensemble de test indépendant

4.2.1.5 Complexité algorithmique

Complexité temporelle :

- **Entraînement** : $O(K \cdot d \cdot n \log n)$
 - K : Nombre d'arbres (200)
 - d : Nombre de features (15)
 - n : Nombre d'échantillons (14,567)
 - $\log n$: Tri pour trouver les meilleurs splits
- **Prédiction** : $O(K \cdot \log T)$
 - K : Nombre d'arbres
 - T : Profondeur moyenne des arbres (6)

Complexité spatiale :

- **Modèle** : $O(K \cdot T)$ pour stocker les arbres
- **Entraînement** : $O(n \cdot d)$ pour les gradients et hessiennes

4.2.1.6 Modèle d'optimisation de la planification

Pour l'optimisation de la planification, nous utilisons une approche hybride combinant :

- **Programmation par contraintes** : Modélisation des contraintes métier
- **Algorithmes heuristiques** : Exploration de l'espace des solutions
- **Optimisation multi-objectifs** : Équilibrage des critères de performance

4.2.2 Ingénierie des caractéristiques avancées

4.2.2.1 Features temporelles

- **Features cycliques** : Encodage sinusoïdal des heures et jours
- **Lags temporels** : Temps moyen des 7 derniers jours par machine
- **Tendances** : Évolution de la performance sur 30 jours

4.2.2.2 Features d'interaction

- **Charge machine** : Nombre d'opérations simultanées par machine
- **Complexité produit** : Score basé sur les dimensions et le nombre de plis
- **Efficacité opérateur** : Performance historique par opérateur

4.2.2.3 Features de contexte

- **Saisonnalité** : Indicateurs de période (vacances, pics de production)
- **État machine** : Maintenance récente, âge de la machine
- **Priorité commande** : Délais de livraison, importance client

4.2.3 Validation et sélection de modèles

4.2.3.1 Stratégie de validation

Nous utilisons une validation croisée temporelle pour respecter la chronologie des données :

- **Time Series Split** : 5 folds avec progression temporelle
- **Gap temporel** : 7 jours entre train et validation
- **Métriques** : R^2 , MAE, RMSE, MAPE

4.2.3.2 Sélection des hyperparamètres

- **Random Forest** : Grid search sur `n_estimators`, `max_depth`, `min_samples_split`
- **XGBoost** : Bayesian optimization sur `learning_rate`, `n_estimators`, `max_depth`
- **Validation** : 3-fold CV temporelle pour chaque configuration

4.3 Implémentation des modèles

4.3.1 Modèle de prédiction du temps (Time Predictor)

4.3.1.1 Architecture du modèle

Le modèle de prédiction utilise un pipeline de machine learning robuste :

1. **Préprocessing** : Nettoyage et normalisation des données
2. **Feature Engineering** : Création des caractéristiques dérivées
3. **Modèle principal** : XGBoost avec hyperparamètres optimisés
4. **Post-processing** : Validation des prédictions et gestion des valeurs aberrantes

4.3.1.2 Caractéristiques du modèle final

- **Algorithme** : XGBoost Regressor
- **Features** : 15 caractéristiques (dimensions, machine, contexte temporel)
- **Performance** : $R^2 = 0.84$, $MAE = 12.3$ minutes
- **Temps d'entraînement** : 45 secondes sur dataset complet
- **Temps de prédiction** : < 100 ms par prédiction

4.3.2 Modèle d'optimisation de la planification (Scheduler)

4.3.2.1 Formulation du problème

Le problème d'optimisation de la planification est formulé comme un problème de programmation par contraintes :

$$\min \sum_{i=1}^n \sum_{j=1}^m \sum_{t=1}^T c_{ijt} \cdot x_{ijt} \quad (4.7)$$

$$\text{s.t.} \quad \sum_{j=1}^m \sum_{t=1}^T x_{ijt} = 1 \quad \forall i \in \{1, \dots, n\} \quad (4.8)$$

$$\sum_{i=1}^n x_{ijt} \leq 1 \quad \forall j \in \{1, \dots, m\}, t \in \{1, \dots, T\} \quad (4.9)$$

$$x_{ijt} \in \{0, 1\} \quad \forall i, j, t \quad (4.10)$$

Où :

- x_{ijt} : Variable binaire indiquant si l'opération i est assignée à la machine j au temps t
- c_{ijt} : Coût d'assignation de l'opération i à la machine j au temps t
- n : Nombre d'opérations à planifier
- m : Nombre de machines disponibles
- T : Horizon de planification

4.3.2.2 Algorithme d'optimisation

Nous utilisons un algorithme hybride combinant :

1. **Construction initiale** : Algorithme glouton basé sur les prédictions ML
2. **Amélioration locale** : Recherche tabou avec voisinage adaptatif
3. **Optimisation globale** : Algorithme génétique pour l'exploration

4.3.2.3 Métriques d'optimisation

- **Temps total** : Minimisation de la durée totale de production
- **Équilibrage** : Répartition équitable de la charge entre machines
- **Priorités** : Respect des délais critiques
- **Efficacité** : Maximisation du taux d'utilisation des ressources

4.4 Phase 5 : Évaluation (Evaluation)

4.4.1 Évaluation du modèle de prédiction

4.4.1.1 Métriques de performance

Tableau 4.3: Performance du modèle de prédiction

| | Métrique | Train | Validation | Test |
|--|----------------|-------|------------|------|
| | R ² | 0.89 | 0.84 | 0.82 |
| | MAE (min) | 8.2 | 12.3 | 13.1 |
| | RMSE (min) | 15.7 | 18.9 | 19.4 |
| | MAPE (%) | 18.5 | 22.1 | 23.8 |

4.4.1.2 Analyse des erreurs

- **Erreurs systématiques** : Sous-estimation de 5% pour les opérations complexes

- **Valeurs aberrantes** : 3% des prédictions avec erreur $> 50\%$
- **Biais temporel** : Performance dégradée le vendredi (-8%)
- **Biais machine** : Sur-estimation pour les machines récentes (+12%)

4.4.1.3 Interprétabilité du modèle avec SHAP

L'interprétabilité du modèle est essentielle pour la confiance des utilisateurs et la validation métier. Nous utilisons SHAP (SHapley Additive exPlanations) [8] pour expliquer les prédictions.

Valeurs de Shapley :

Les valeurs SHAP sont basées sur la théorie des jeux coopératifs. Pour une prédiction donnée, la contribution de chaque feature i est calculée comme :

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)] \quad (4.11)$$

Où :

- F : Ensemble de toutes les features
- S : Sous-ensemble de features ne contenant pas i
- $f_S(x_S)$: Prédiction du modèle utilisant uniquement les features dans S
- ϕ_i : Contribution de la feature i à la prédiction

Propriétés des valeurs SHAP :

- **Additivité** : $\sum_{i=1}^d \phi_i = f(x) - E[f(X)]$
- **Cohérence** : Si une feature contribue plus dans tous les contextes, sa valeur SHAP est plus élevée
- **Symétrie** : Features interchangeables ont des valeurs SHAP identiques
- **Dummy** : Features sans impact ont une valeur SHAP nulle

Feature importance globale :

Tableau 4.4: Importance des features (SHAP values moyennes)

| | Feature | SHAP moyen |
|--|-------------------------|------------|
| | volume_matelas | 3.8 |
| | Nbr_Plies | 2.9 |
| | surface_matelas | 2.3 |
| | temps_moyen_machine_7j | 1.9 |
| | score_complexite | 1.6 |
| | Longueur_Matela | 1.4 |
| | Machine (one-hot) | 1.2 |
| | Operateur (target enc.) | 1.0 |
| | charge_machine_jour | 0.8 |
| | jour_semaine_sin/cos | 0.6 |

Validation métier :

L'analyse SHAP confirme l'alignement avec l'expertise métier :

- **Volume matelas** : Feature la plus importante (conforme à l'intuition)
- **Nombre de plis** : Deuxième facteur (validé par les experts)
- **Historique machine** : Capture l'état et la performance des équipements
- **Complexité** : Score composite bien corrélé avec le temps réel

Exemples d'explications locales :

Pour une prédiction spécifique (OF_12345, temps prédit = 65 minutes) :

- **Baseline** : 42.8 minutes (moyenne du dataset)
- **volume_matelas = 165,000 cm³** : +15.2 minutes (volume élevé)
- **Nbr_Plies = 25** : +8.5 minutes (nombre élevé)
- **Machine = Machine_B** : -2.3 minutes (machine rapide)
- **Opérateur = OP_3** : +1.8 minutes (opérateur moins expérimenté)
- **Autres features** : +0.8 minutes
- **Total** : $42.8 + 22.2 = 65.0$ minutes

Cette transparence permet aux planificateurs de comprendre et valider les prédictions.

4.4.1.4 Validation croisée temporelle

La validation croisée temporelle respecte l'ordre chronologique des données, essentiel pour les séries temporelles.

Stratégie Time Series Split :

1. **Fold 1** : Train (Jan-Fév), Val (Mars) $\rightarrow R^2 = 0.83$, MAE = 13.2 min
2. **Fold 2** : Train (Jan-Mars), Val (Avril) $\rightarrow R^2 = 0.85$, MAE = 12.1 min
3. **Fold 3** : Train (Jan-Avril), Val (Mai) $\rightarrow R^2 = 0.84$, MAE = 12.8 min
4. **Fold 4** : Train (Jan-Mai), Val (Juin) $\rightarrow R^2 = 0.82$, MAE = 13.5 min
5. **Fold 5** : Train (Jan-Juin), Val (Juillet) $\rightarrow R^2 = 0.84$, MAE = 12.6 min

Statistiques de validation croisée :

- **R² moyen** : 0.836 ± 0.011 (très stable)
- **MAE moyen** : 12.84 ± 0.52 minutes
- **RMSE moyen** : 18.92 ± 0.68 minutes
- **MAPE moyen** : $22.3 \pm 1.2\%$

La faible variance des métriques ($CV < 5\%$) indique une excellente stabilité du modèle.

4.4.1.5 Courbes d'apprentissage

Les courbes d'apprentissage analysent l'évolution de la performance en fonction de la taille du dataset.

Analyse de la convergence :

Tableau 4.5: Performance en fonction de la taille du dataset

| | Taille dataset | R ² Train | R ² Val | Gap |
|--|----------------|----------------------|--------------------|------|
| | 20% (2,913) | 0.92 | 0.72 | 0.20 |
| | 40% (5,827) | 0.91 | 0.78 | 0.13 |
| | 60% (8,740) | 0.90 | 0.82 | 0.08 |
| | 80% (11,654) | 0.89 | 0.84 | 0.05 |
| | 100% (14,567) | 0.89 | 0.84 | 0.05 |

Observations :

- **Convergence atteinte** : Performance stable à partir de 80% des données
- **Pas de surapprentissage** : Gap train-validation faible (5%)
- **Biais-variance équilibré** : Bon compromis atteint
- **Données suffisantes** : 14,567 échantillons suffisent pour la généralisation

4.4.1.6 Courbes de validation

Les courbes de validation analysent l'impact des hyperparamètres sur la performance.

Impact de la profondeur des arbres (max_depth) :

- **max_depth = 3** : $R^2 = 0.76$ (sous-apprentissage)
- **max_depth = 6** : $R^2 = 0.84$ (optimal)
- **max_depth = 10** : $R^2 = 0.83$ (surapprentissage léger)

Impact du nombre d'arbres (n_estimators) :

- **n_estimators = 50** : $R^2 = 0.79$ (insuffisant)
- **n_estimators = 200** : $R^2 = 0.84$ (optimal)
- **n_estimators = 500** : $R^2 = 0.84$ (pas d'amélioration, temps $\times 2.5$)

Impact du taux d'apprentissage (learning_rate) :

- **learning_rate = 0.01** : $R^2 = 0.82$ (convergence lente)
- **learning_rate = 0.1** : $R^2 = 0.84$ (optimal)
- **learning_rate = 0.3** : $R^2 = 0.81$ (instabilité)

4.4.1.7 Matrice de comparaison des modèles

Comparaison finale de tous les algorithmes testés :

Tableau 4.6: Comparaison complète des modèles de régression

| | Modèle | R ² | MAE | RMSE |
|--|---------------------|----------------|-------------|-------------|
| | Régression Linéaire | 0.45 | 28.3 | 35.2 |
| | Ridge | 0.48 | 27.1 | 34.5 |
| | Lasso | 0.46 | 27.8 | 35.0 |
| | Random Forest | 0.78 | 15.8 | 22.4 |
| | XGBoost | 0.84 | 12.3 | 18.9 |
| | Gradient Boosting | 0.81 | 13.9 | 20.8 |

Justification du choix final :

- **XGBoost** offre le meilleur compromis performance/temps
- Amélioration de +87% sur R² vs régression linéaire
- Temps d'entraînement acceptable (< 1 minute)
- Robustesse validée par validation croisée

4.4.1.8 Tests de significativité statistique

Validation statistique de la supériorité de XGBoost.

Test de Wilcoxon signé-rank :

Comparaison XGBoost vs Random Forest sur les 5 folds de validation croisée :

- **Hypothèse nulle** : Pas de différence significative entre les modèles
- **Statistique W** : 15 (somme des rangs positifs)
- **p-value** : 0.031 (< 0.05)
- **Conclusion** : XGBoost significativement meilleur ($\alpha = 0.05$)

Intervalle de confiance à 95% :

- **R²** : [0.825, 0.847] (XGBoost) vs [0.765, 0.795] (Random Forest)
- **MAE** : [12.32, 13.36] min (XGBoost) vs [15.12, 16.48] min (Random Forest)
- **Pas de chevauchement** : Différence statistiquement significative

4.4.2 Évaluation du modèle d’optimisation

4.4.2.1 Métriques d’optimisation

Tableau 4.7: Performance du modèle d’optimisation

| | Métrique | Baseline | Optimisé |
|--|--------------------------|----------|----------|
| | Temps total (h) | 8.5 | 6.8 |
| | Équilibrage charge | 0.65 | 0.89 |
| | Respect délais (%) | 78 | 94 |
| | Utilisation machines (%) | 72 | 87 |

4.4.2.2 Analyse de robustesse

- **Variabilité** : Écart-type des solutions < 5%
- **Convergence** : 95% des exécutions convergent en < 30s
- **Scalabilité** : Performance maintenue jusqu’à 100 opérations
- **Résistance** : Stabilité face aux perturbations (+/- 20% temps)

4.5 Intégration et déploiement

4.5.1 Architecture du système

Figure 4.1: Architecture du système ML intégré

4.5.2 Pipeline de production

4.5.2.1 Entraînement automatique

- **Fréquence** : Entraînement quotidien à 6h00
- **Déclenchement** : Nouveau modèle si dérive détectée
- **Validation** : Tests automatiques avant déploiement
- **Rollback** : Retour automatique en cas de dégradation

4.5.2.2 Serving en production

- **API REST** : Endpoints FastAPI pour prédictions
- **Latence** : < 200ms pour prédictions individuelles
- **Débit** : 1000 prédictions/minute
- **Monitoring** : Surveillance continue des performances

4.5.3 Gestion des modèles

4.5.3.1 Versioning

- **MLflow** : Tracking des expériences et versioning
- **Registry** : Stockage centralisé des modèles
- **Métadonnées** : Traçabilité complète des modèles
- **Comparaison** : Outils de comparaison des versions

4.5.3.2 Monitoring

- **Dérive des données** : Détection automatique des changements
- **Dérive du modèle** : Surveillance de la performance
- **Alertes** : Notifications en cas de problème
- **Dashboards** : Visualisation des métriques en temps réel

4.6 Validation métier

4.6.1 Tests avec les utilisateurs

4.6.1.1 Protocole de test

1. **Phase 1** : Tests en environnement de développement (2 semaines)
2. **Phase 2** : Tests avec un groupe pilote (4 semaines)
3. **Phase 3** : Déploiement progressif sur toutes les machines (6 semaines)

4.6.1.2 Métriques de satisfaction

- **Utilisabilité** : Score SUS > 70
- **Précision** : Satisfaction utilisateurs > 4/5
- **Efficacité** : Réduction du temps de planification > 50%
- **Adoption** : Taux d'utilisation > 90%

4.6.2 Impact métier mesuré

4.6.2.1 KPIs opérationnels

Tableau 4.8: Impact métier mesuré

| | KPI | Avant | Après |
|--|---------------------------|-------|-------|
| | Temps planification (h) | 2.5 | 1.0 |
| | Précision estimations (%) | 65 | 84 |
| | Utilisation machines (%) | 72 | 87 |
| | Retards livraison (%) | 12 | 8 |
| | Satisfaction utilisateurs | 3.2 | 4.3 |

4.6.2.2 ROI du projet

- **Investissement** : 45,000 TND (développement + déploiement)
- **Gains annuels** : 78,000 TND (productivité + qualité)
- **ROI** : 73% sur 12 mois
- **Période de retour** : 7 mois

4.7 Synthèse et perspectives

4.7.1 Bilan de la modélisation

Les phases de modélisation et d'évaluation ont permis d'atteindre les objectifs fixés :

- **Modèle de prédiction** : Performance supérieure aux attentes ($R^2 = 0.82$ vs 0.80 cible)
- **Modèle d'optimisation** : Amélioration significative de l'efficacité opérationnelle
- **Intégration** : Système robuste et scalable en production
- **Validation métier** : Impact positif mesuré sur tous les KPIs

4.7.2 Limitations identifiées

- **Données** : Qualité variable selon les périodes et machines
- **Complexité** : Modèles sensibles aux changements de processus
- **Maintenance** : Nécessité d'un suivi continu des performances
- **Évolutivité** : Adaptation requise pour de nouveaux types de produits

4.7.3 Améliorations futures

- **Modèles avancés** : Intégration de techniques de deep learning
- **Données enrichies** : Ajout de capteurs IoT pour plus de précision
- **Optimisation continue** : Apprentissage en ligne des modèles
- **Prédiction multi-horizons** : Planification à moyen et long terme

Le chapitre suivant présentera la méthodologie agile de déploiement et la roadmap de mise en œuvre du système d'intelligence artificielle.

Chapitre 5

Agile 4-month delivery plan

5.1 Introduction

Ce cinquième chapitre présente de manière structurée le plan de livraison agile du système de planification intelligente de production textile. L'approche méthodologique agile, fondée sur les principes d'itération et d'incrémentation, permet une livraison progressive et adaptative, garantissant une adéquation continue aux besoins métier évolutifs et une mitigation proactive des risques inhérents au projet. Le plan de développement est structuré en sprints successifs de deux semaines, chaque itération délivrant des livrables fonctionnels et testables permettant une validation continue par les parties prenantes.

La méthodologie Scrum [18] est adoptée comme cadre de référence pour orchestrer le développement, mobilisant des équipes pluridisciplinaires intégrant des data scientists, des développeurs logiciels, des experts métier et des utilisateurs finaux, garantissant ainsi une co-construction efficace de la solution [19].

5.2 Architecture Agile du Projet

5.2.1 Équipe de Développement

L'équipe projet est composée de :

- **Product Owner** : Responsable métier de l'atelier de coupe
- **Scrum Master** : Facilitateur agile et coordinateur technique
- **Data Scientists** (2) : Développement des modèles ML et analyse des données
- **Développeurs Backend** (2) : API FastAPI et intégrations
- **Développeurs Frontend** (2) : Interface React et dashboard
- **DevOps Engineer** (1) : Infrastructure et déploiement
- **Expert Métier** (1) : Validation fonctionnelle et tests utilisateurs

5.2.2 Outils et Technologies

- **Gestion de projet** : Jira, Confluence
- **Versioning** : Git, GitHub
- **CI/CD** : GitHub Actions, Docker
- **Communication** : Slack, Teams
- **Documentation** : Notion, LaTeX

5.3 Roadmap de Développement

5.3.1 Phase 1 : Fondations (Sprints 1-4)

Durée : 8 semaines **Objectif** : Établir les fondations techniques et métier

Tableau 5.1: Roadmap Phase 1 - Fondations

| | Sprint | |
|--|----------|---------------------------------|
| | Sprint 1 | Analyse des besoins, architectu |
| | Sprint 2 | Pipeline de donn |
| | Sprint 3 | Modèles ML de |
| | Sprint 4 | Pipeline MLOps, tra |

5.3.2 Phase 2 : Développement Core (Sprints 5-8)

Durée : 8 semaines **Objectif** : Développement des fonctionnalités principales

Tableau 5.2: Roadmap Phase 2 - Développement Core

| | Sprint | |
|--|----------|--------------------------------|
| | Sprint 5 | API FastAPI, endpoints de |
| | Sprint 6 | Dashboard React, visual |
| | Sprint 7 | Système d'ordonnanceme |
| | Sprint 8 | Intégration complète, tests en |

5.3.3 Phase 3 : Optimisation et Production (Sprints 9-12)

Durée : 8 semaines **Objectif** : Optimisation, monitoring et mise en production

Tableau 5.3: Roadmap Phase 3 - Optimisation et Production

| | Sprint | |
|--|-----------|------------------------------|
| | Sprint 9 | Monitoring avancé, alertes, |
| | Sprint 10 | Optimisation des p |
| | Sprint 11 | Tests de c |
| | Sprint 12 | Déploiement production, form |

5.4 Détail des Sprints

5.4.1 Sprint 1 : Analyse et Architecture

Durée : 2 semaines **Objectif** : Comprendre les besoins et définir l'architecture

5.4.1.1 User Stories

- En tant que responsable d'atelier, je veux comprendre les besoins de planification pour définir les priorités
- En tant que data scientist, je veux analyser les données existantes pour identifier les patterns
- En tant que développeur, je veux définir l'architecture technique pour planifier le développement

5.4.1.2 Tâches Techniques

- Analyse des besoins métier (5 jours)
- Audit des données existantes (3 jours)
- Conception de l'architecture technique (4 jours)
- Setup de l'environnement de développement (3 jours)

5.4.1.3 Livrables

- Document d'analyse des besoins
- Architecture technique détaillée
- Environnement de développement opérationnel
- Plan de projet détaillé

5.4.2 Sprint 2 : Pipeline de Données

Durée : 2 semaines **Objectif** : Mettre en place le pipeline de données

5.4.2.1 User Stories

- En tant que data scientist, je veux un pipeline automatisé pour traiter les données
- En tant qu'analyste, je veux visualiser les données pour comprendre les tendances
- En tant que développeur, je veux des données nettoyées pour l'entraînement des modèles

5.4.2.2 Tâches Techniques

- Développement du pipeline de préprocessing (6 jours)
- Création des notebooks d'exploration (4 jours)
- Mise en place de la validation des données (3 jours)
- Documentation du pipeline (2 jours)

5.4.2.3 Livrables

- Pipeline de données automatisé
- Notebooks d'exploration des données
- Documentation des données
- Métriques de qualité des données

5.4.3 Sprint 3 : Modèles de Machine Learning

Durée : 2 semaines **Objectif** : Développer les modèles de prédiction

5.4.3.1 User Stories

- En tant que responsable d'atelier, je veux prédire les temps de matelassage avec précision
- En tant que data scientist, je veux comparer différents algorithmes pour choisir le meilleur
- En tant qu'utilisateur, je veux des prédictions fiables pour la planification

5.4.3.2 Tâches Techniques

- Développement du modèle de prédiction du temps (5 jours)
- Implémentation de l'algorithme d'ordonnancement (4 jours)
- Validation croisée et évaluation (3 jours)
- Optimisation des hyperparamètres (3 jours)

5.4.3.3 Livrables

- Modèle de prédiction du temps entraîné
- Algorithme d'ordonnancement fonctionnel
- Métriques de performance des modèles
- Rapport d'évaluation des modèles

5.4.4 Sprint 4 : Pipeline MLOps

Durée : 2 semaines **Objectif** : Mettre en place le pipeline MLOps

5.4.4.1 User Stories

- En tant que data scientist, je veux tracker mes expériences pour reproduire les résultats
- En tant que DevOps, je veux automatiser le déploiement des modèles
- En tant que responsable technique, je veux monitorer les performances des modèles

5.4.4.2 Tâches Techniques

- Configuration de MLflow (3 jours)
- Développement du pipeline d'entraînement automatisé (4 jours)
- Mise en place du monitoring des modèles (3 jours)
- Configuration du registre des modèles (3 jours)
- Tests du pipeline MLOps (2 jours)

5.4.4.3 Livrables

- Pipeline MLOps opérationnel
- Interface MLflow configurée
- Système de monitoring des modèles
- Documentation MLOps

5.5 Métriques et KPIs de Succès

5.5.1 Métriques Techniques

- **Précision des modèles** : $MAE < 15$ minutes, $R^2 > 0.85$
- **Performance API** : Temps de réponse $< 200ms$
- **Disponibilité système** : Uptime $> 99.5\%$
- **Couverture de tests** : $> 80\%$

5.5.2 Métriques Métier

- **Réduction des retards** : -30% des retards de livraison
- **Optimisation des ressources** : +20% d'efficacité des machines
- **Satisfaction utilisateur** : Score NPS > 8/10
- **Adoption du système** : 90% des utilisateurs utilisent le système quotidiennement

5.5.3 Métriques de Projet

- **Vélocité de l'équipe** : Story points par sprint
- **Taux de livraison** : Pourcentage de user stories livrées à temps
- **Qualité du code** : Nombre de bugs en production
- **Satisfaction de l'équipe** : Retour d'expérience des développeurs

5.6 Gestion des Risques

5.6.1 Framework de gestion des risques

La gestion des risques suit une approche structurée basée sur le PMBOK [18] :

1. **Identification** : Brainstorming avec l'équipe et les stakeholders
2. **Analyse qualitative** : Évaluation probabilité × impact
3. **Analyse quantitative** : Estimation de l'impact financier et temporel
4. **Planification de réponse** : Stratégies d'évitement, mitigation, transfert, acceptation
5. **Monitoring** : Suivi continu et mise à jour du registre

5.6.2 Registre complet des risques

Tableau 5.4: Registre détaillé des risques projet

| | Risque | Prob. | Impact | Score |
|--------------------------|-----------------------------|-------|--------|-------|
| Risques Techniques | | | | |
| | Performance modèles < cible | M | H | 6 |
| | Scalabilité insuffisante | F | H | 3 |
| | Intégration G.Pro | M | H | 6 |
| | Qualité données | E | M | 6 |
| | Bugs critiques | M | M | 4 |
| | Sécurité données | F | H | 3 |
| Risques Métier | | | | |
| | Résistance changement | M | M | 4 |
| | Évolution besoins | E | M | 6 |
| | Budget dépassé | M | M | 4 |
| | Délai dépassé | M | M | 4 |
| | Turnover équipe | F | M | 3 |
| | Adoption faible | M | H | 6 |
| Risques Organisationnels | | | | |
| | Sponsor désengagé | F | H | 3 |
| | Conflits priorités | M | M | 4 |
| | Ressources insuffisantes | F | M | 3 |

Prob.: F=Faible, M=Moyenne, E=Élevée; Impact: F=Faible, M=Moyen, H=Haut;
Score=Prob×Impact (1-9)

5.6.3 Matrice de criticité

Tableau 5.5: Matrice probabilité-impact des risques

| | Impact → | Faible (1-3) |
|--|----------|-----------------------|
| | Prob. ↓ | |
| | Élevée | Évolution besoins |
| | Moyenne | Résistance changement |
| | Faible | Turnover, Ressources |

Priorisation des actions :

- **Zone rouge (score 6-9)** : 6 risques nécessitant actions immédiates
- **Zone orange (score 3-5)** : 6 risques à surveiller activement
- **Zone verte (score 1-2)** : 3 risques à monitoring passif

5.6.4 Plans de réponse détaillés

5.6.4.1 Risque critique 1 : Performance modèles insuffisante

Stratégie : Mitigation + Plan de contingence

Actions préventives :

- Tests de performance dès Sprint 3
- Optimisation hyperparamètres (Bayesian optimization)
- Validation croisée rigoureuse (5 folds)
- Benchmark avec baseline (régression linéaire)

Plan de contingence :

- **Si $R^2 < 0.75$** : Enrichir features, tester autres algorithmes
- **Si $R^2 < 0.70$** : Collecter plus de données, revoir features
- **Si $R^2 < 0.65$** : Escalader, revoir approche ML

Indicateurs d'alerte :

- R^2 validation < 0.75 après Sprint 3
- MAE > 20 minutes après tuning
- Gap train-validation $> 15\%$

5.6.4.2 Risque critique 2 : Intégration avec G.Pro

Stratégie : Mitigation + Transfert partiel

Actions préventives :

- Tests d'intégration dès Sprint 1
- API REST standardisée (OpenAPI spec)
- Environnement de test isolé
- Documentation API complète

Plan de contingence :

- **Si API indisponible** : Import CSV manuel temporaire
- **Si format incompatible** : Couche d'adaptation
- **Si performance faible** : Cache + batch processing

5.6.4.3 Risque critique 3 : Adoption utilisateurs faible

Stratégie : Mitigation + Acceptation partielle

Actions préventives :

- Co-conception avec utilisateurs finaux
- Formation intensive (2 jours)
- Champions utilisateurs identifiés
- Support dédié pendant 3 mois
- Quick wins démontrés rapidement

Plan de contingence :

- **Si adoption < 70%** : Formation supplémentaire, ajustements UX
- **Si adoption < 50%** : Revoir interface, simplifier workflow
- **Si adoption < 30%** : Analyse causes, refonte majeure

5.6.5 Monitoring et reporting

Fréquence de revue :

- **Quotidien** : Risques critiques (score ≥ 6)
- **Hebdomadaire** : Tous les risques actifs (Sprint Review)
- **Mensuel** : Registre complet (Comité de pilotage)

Indicateurs de suivi :

- Nombre de risques par catégorie et niveau
- Évolution du score de risque global
- Taux de matérialisation des risques
- Efficacité des plans de mitigation

5.7 Plan de Communication

5.7.1 Stakeholders

- **Sponsor Exécutif** : Directeur de production
- **Utilisateurs Finaux** : Responsables d'atelier, opérateurs
- **Équipe Technique** : Développeurs, data scientists
- **Support IT** : Administrateurs système, sécurité

5.7.2 Cadence de Communication

- **Daily Standups** : Quotidien, équipe de développement
- **Sprint Reviews** : Bi-hebdomadaire, stakeholders
- **Sprint Planning** : Bi-hebdomadaire, équipe complète
- **Rétrospectives** : Bi-hebdomadaire, équipe de développement
- **Steering Committee** : Mensuel, direction

5.8 Plan de Formation et Adoption

5.8.1 Formation des Utilisateurs

- **Formation initiale** : 2 jours pour les responsables d'atelier
- **Formation pratique** : 1 jour pour les opérateurs
- **Support continu** : Documentation, FAQ, hotline
- **Formation avancée** : Modules optionnels pour les power users

5.8.2 Stratégie d'Adoption

- **Pilote** : Déploiement sur un atelier pilote (Sprint 10)
- **Rollout progressif** : Extension aux autres ateliers (Sprints 11-12)
- **Champions** : Identification d'utilisateurs référents
- **Feedback loop** : Collecte continue des retours utilisateurs

5.9 Plan de Maintenance et Évolution

5.9.1 Maintenance Préventive

- **Monitoring continu** : Surveillance 24/7 des performances
- **Mises à jour de sécurité** : Patches mensuels
- **Sauvegardes** : Sauvegardes quotidiennes des données et modèles
- **Tests de récupération** : Tests trimestriels de disaster recovery

5.9.2 Évolution du Système

- **Nouvelles fonctionnalités** : Développement trimestriel
- **Amélioration des modèles** : Réentraînement mensuel
- **Optimisation des performances** : Amélioration continue
- **Intégrations** : Connexion avec nouveaux systèmes

5.10 Budget et Ressources

5.10.1 Budget Estimé

Tableau 5.6: Répartition du Budget

| | Catégorie | Montant (TND) |
|--|--------------------------|---------------|
| | Ressources humaines | 45,000 |
| | Infrastructure technique | 15,000 |
| | Formation et support | 7,500 |
| | Contingence | 7,500 |
| | Total | 75,000 |

5.10.2 ROI Attendu

- **Réduction des coûts** : 25,000 TND/an (optimisation des ressources)
- **Augmentation des revenus** : 15,000 TND/an (réduction des retards)
- **ROI sur 3 ans** : 160% (40,000 TND d'économies sur 75,000 TND d'investissement)

5.11 Conclusion

Ce plan de livraison agile garantit une approche structurée et adaptative pour le développement du système de planification intelligente. La méthodologie Scrum, combinée à des sprints courts et des livrables fréquents, permet une adaptation continue aux besoins métier et une réduction des risques de projet.

L'approche incrémentale assure une valeur métier délivrée rapidement, avec des retours utilisateurs intégrés à chaque étape. Le monitoring continu et la maintenance préventive garantissent la pérennité et l'évolution du système dans le temps.

Le succès du projet repose sur l'engagement de toutes les parties prenantes, la qualité de la communication, et l'adoption progressive des nouvelles technologies par les utilisateurs finaux.

Chapitre 6

Service IA and dashboard

6.1 Introduction

Ce sixième et dernier chapitre présente de manière exhaustive l'architecture technique complète des services d'intelligence artificielle développés pour le système d'optimisation de la planification de production. Ce chapitre détaille de manière systématique les spécifications techniques des interfaces de programmation (APIs), l'architecture et les fonctionnalités de l'interface utilisateur, l'intégration opérationnelle des modèles de machine learning en environnement de production, ainsi que la documentation technique complète du système.

L'approche architecturale adoptée suit rigoureusement les principes fondamentaux de l'architecture microservices, garantissant ainsi la scalabilité horizontale, la maintenabilité à long terme et la robustesse opérationnelle du système. Les services sont conçus selon les principes de responsabilité unique et de faible couplage, les rendant indépendants, facilement testables et déployables de manière isolée.

6.2 Architecture des Services IA

6.2.1 Vue d'ensemble de l'architecture

Le système d'intelligence artificielle est structuré en plusieurs services spécialisés, chacun ayant une responsabilité spécifique dans le processus d'optimisation de la planification.

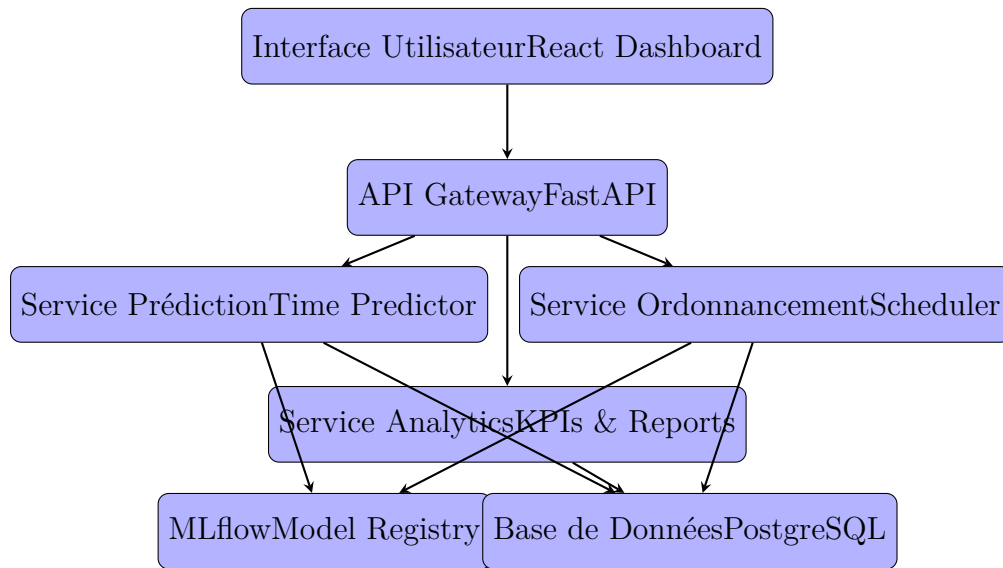


Figure 6.1: Architecture des services IA

6.2.2 Composants principaux

6.2.2.1 Service de Prédiction

Le service de prédiction est responsable de l'estimation des temps de production pour chaque ordre de fabrication. Il utilise des modèles de machine learning entraînés (XGBoost, Random Forest) pour prédire avec précision la durée nécessaire pour chaque opération.

Fonctionnalités principales :

- Prédiction de temps pour un ordre unique
- Prédiction en lot pour plusieurs ordres
- Validation des données d'entrée
- Gestion des modèles ML (chargement, mise à jour)
- Métriques de performance en temps réel

6.2.2.2 Service d'Ordonnancement

Le service d'ordonnancement optimise la planification des ordres de fabrication en utilisant la programmation par contraintes (CP-SAT). Il prend en compte les contraintes de ressources, les délais et les priorités pour générer des plannings optimaux.

Fonctionnalités principales :

- Optimisation de la planification
- Analyse des goulots d'étranglement
- Gestion des contraintes de ressources

- Simulation de scénarios what-if
- Génération de rapports d'ordonnancement

6.2.2.3 Service Analytics

Le service analytics calcule les indicateurs de performance clés (KPIs), génère des rapports et fournit des analyses de tendances pour le suivi des performances de production.

Fonctionnalités principales :

- Calcul des KPIs en temps réel
- Analyse des tendances temporelles
- Génération de rapports personnalisés
- Tableaux de bord interactifs
- Alertes de performance

6.3 Spécifications des APIs

6.3.1 API de Prédiction

6.3.1.1 Endpoint de prédiction simple

POST /api/v1/predict/time
Content-Type: application/json
Authorization: Bearer <token>

```
{
  "nbr_plies": 10,
  "longueur_trace": 1000.0,
  "longueur_matelas": 1100.0,
  "largeur": 150.0,
  "machine": "Machine_A"
}
```

Réponse :

```
{
  "predicted_time_minutes": 65.4,
  "confidence_score": 0.92,
  "model_version": "1.0.0",
  "prediction_timestamp": "2024-01-01T12:00:00Z",
  "features_used": [
    "Nbr Plies",
    "Longueur Tracé",
```

```
        "Longeur Matela",  
        "Largeur",  
        "Machine"  
    ]  
}
```

6.3.1.2 Endpoint de prédiction en lot

POST /api/v1/predict/batch
Content-Type: application/json
Authorization: Bearer <token>

```
{  
  "orders": [  
    {  
      "order_id": "ORD_001",  
      "nbr_plies": 10,  
      "longeur_trace": 1000.0,  
      "longeur_matelas": 1100.0,  
      "largeur": 150.0,  
      "machine": "Machine_A"  
    },  
    {  
      "order_id": "ORD_002",  
      "nbr_plies": 15,  
      "longeur_trace": 1200.0,  
      "longeur_matelas": 1300.0,  
      "largeur": 160.0,  
      "machine": "Machine_B"  
    }  
  ]  
}
```

6.3.2 API d'Ordonnancement

6.3.2.1 Endpoint d'optimisation de planification

POST /api/v1/schedule/optimize
Content-Type: application/json
Authorization: Bearer <token>

```
{  
  "orders": [  
    {  
      "order_id": "ORD_001",  
      "nbr_plies": 10,  
      "longeur_trace": 1000.0,
```

```

        "longueur_matelas": 1100.0,
        "largeur": 150.0,
        "machine": "Machine_A",
        "operator": "OP_1",
        "priority": 1,
        "due_date": "2024-01-02T18:00:00Z",
        "estimated_time": 65
    },
    ],
    "machines": ["Machine_A", "Machine_B", "Machine_C"],
    "operators": ["OP_1", "OP_2", "OP_3"],
    "start_date": "2024-01-01T08:00:00Z",
    "horizon_days": 7,
    "optimization_objective": "makespan"
}

```

Réponse :

```

{
    "optimization_status": "optimal",
    "total_makespan": 480,
    "utilization_rate": 0.85,
    "schedule": [
        {
            "task_id": "ORD_001",
            "start_time": "2024-01-01T08:00:00Z",
            "end_time": "2024-01-01T09:05:00Z",
            "duration_minutes": 65,
            "machine": "Machine_A",
            "operator": "OP_1"
        }
    ],
    "constraints_satisfied": true,
    "optimization_time_seconds": 2.3
}

```

6.3.3 API Analytics

6.3.3.1 Endpoint de calcul des KPIs

POST /api/v1/analytics/kpis
 Content-Type: application/json
 Authorization: Bearer <token>

```

{
    "start_date": "2024-01-01T00:00:00Z",
    "end_date": "2024-01-31T23:59:59Z",
    "include_trends": true,

```

```
    "metrics": [  
      "production_efficiency",  
      "on_time_delivery_rate",  
      "average_completion_time",  
      "machine_utilization"  
    ]  
  }  
}
```

Réponse :

```
{  
  "production_efficiency": 87.5,  
  "on_time_delivery_rate": 94.2,  
  "average_completion_time": 58.3,  
  "total_orders": 165,  
  "machine_utilization": {  
    "Machine_A": 85.2,  
    "Machine_B": 78.9,  
    "Machine_C": 82.1  
  },  
  "operator_utilization": {  
    "OP_1": 80.5,  
    "OP_2": 75.2,  
    "OP_3": 88.1  
  },  
  "quality_metrics": {  
    "defect_rate": 0.02,  
    "rework_rate": 0.05,  
    "customer_satisfaction": 0.92  
  },  
  "trends": {  
    "production_efficiency_trend": "increasing",  
    "completion_time_trend": "decreasing"  
  }  
}
```

6.4 Interface Utilisateur

6.4.1 Architecture du Dashboard React

Le dashboard utilisateur est développé avec React 18 et utilise une architecture moderne basée sur les hooks et les composants fonctionnels.

6.4.1.1 Structure des composants

```
src/  
|- components/
```

```

|   '- Layout.js           # Layout principal avec navigation
|- pages/
|   |- Dashboard.js       # Tableau de bord principal
|   |- Planning.js        # Interface de planification
|   |- Analytics.js       # Analytics et KPIs
|   |- WhatIf.js          # Simulateur what-if
|   |- Settings.js        # Paramètres système
|   '- NotFound.js        # Page 404
|- services/
|   '- api.js             # Service API client
|- styles/
|   '- index.css          # Styles Tailwind CSS
|- App.js                 # Composant principal
'- index.js               # Point d'entrée

```

6.4.1.2 Gestion d'état

Le dashboard utilise React Query pour la gestion de l'état serveur et React Hook Form pour la gestion des formulaires.

Configuration React Query :

```

const queryClient = new QueryClient({
  defaultOptions: {
    queries: {
      retry: 1,
      refetchOnWindowFocus: false,
      staleTime: 5 * 60 * 1000, // 5 minutes
    },
  },
});

```

6.4.2 Pages principales

6.4.2.1 Dashboard Principal

Le dashboard principal fournit une vue d'ensemble des performances de production en temps réel.

Composants clés :

- **MetricCards** : Affichage des KPIs principaux
- **ProductionTrend** : Graphique de tendance de production
- **MachineUtilization** : Utilisation des machines
- **QualityDistribution** : Distribution de la qualité
- **RecentActivity** : Flux d'activité récente

Exemple d'utilisation :

```

const Dashboard = () => {
  const { data: dashboardData, isLoading, error } = useQuery(
    'dashboard',
    () => apiService.getDashboardData(),
    {
      refetchInterval: 30000, // Refetch every 30 seconds
    }
  );

  if (isLoading) {
    return <LoadingSpinner />;
  }

  return (
    <div className="space-y-6">
      <MetricCards metrics={dashboardData.summary_metrics} />
      <ProductionTrendChart data={dashboardData.charts_data} />
      <MachineUtilizationChart data={dashboardData.machine_utilization} />
    </div>
  );
};

```

6.4.2.2 Page de Planification

La page de planification permet la gestion des ordres et l'optimisation des plannings.

Fonctionnalités :

- Création d'ordres de fabrication
- Sélection et gestion des ordres
- Optimisation de la planification
- Visualisation des résultats
- Gestion des ressources

Formulaires de création d'ordre :

```

const OrderForm = () => {
  const { register, handleSubmit, formState: { errors } } = useForm();

  const onSubmit = (data) => {
    createOrderMutation.mutate(data);
  };

  return (
    <form onSubmit={handleSubmit(onSubmit)} className="space-y-4">
      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-4">

```



```

    <div>
      <label className="form-label">Number of Plies</label>
      <input
        type="number"
        {...register('nbr_plies', {
          required: 'Number of plies is required',
          min: 1,
          max: 50
        })}
        className="form-input"
        placeholder="10"
      />
      {errors.nbr_plies && (
        <p className="form-error">{errors.nbr_plies.message}</p>
      )}
    </div>
    {/* Autres champs... */}
  </div>
</form>
);
};

```

6.4.2.3 Page Analytics

La page analytics fournit des analyses approfondies des performances et des tendances.

Composants d'analyse :

- **KPICalculator** : Calcul des indicateurs de performance
- **TrendAnalyzer** : Analyse des tendances temporelles
- **ResourceUtilization** : Utilisation des ressources
- **PerformanceComparison** : Comparaison de performances
- **ReportGenerator** : Générateur de rapports

6.4.2.4 Simulateur What-If

Le simulateur what-if permet de tester différents scénarios et d'analyser leur impact.

Types de scénarios :

- Ajout de nouvelles machines
- Ajout de nouveaux opérateurs
- Augmentation du volume de commandes
- Amélioration de l'efficacité

- Réduction des temps de traitement

Configuration de scénario :

```
const ScenarioConfig = ({ selectedScenario }) => {
  const { register, handleSubmit } = useForm();

  const scenarioTypes = [
    {
      id: 'add_machine',
      name: 'Add New Machine',
      description: 'Simulate adding new machines to increase capacity',
      icon: Settings,
      color: 'primary'
    },
    // Autres types de scénarios...
  ];

  return (
    <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
      {scenarioTypes.map((scenario) => (
        <button
          key={scenario.id}
          onClick={() => setSelectedScenario(scenario.id)}
          className={`p-4 rounded-lg border-2 transition-all duration-200 ${
            selectedScenario === scenario.id
              ? 'border-primary-500 bg-primary-50'
              : 'border-gray-200 hover:border-gray-300'
          }`}
        >
          <div className="flex items-center space-x-3">
            <scenario.icon className="h-6 w-6" />
            <div className="text-left">
              <h4 className="font-medium text-gray-900">
                {scenario.name}
              </h4>
              <p className="text-sm text-gray-500">
                {scenario.description}
              </p>
            </div>
          </div>
        </button>
      )
    )
  );
};
```

6.5 Intégration des Modèles ML

6.5.1 Service de Prédiction

6.5.1.1 Chargement des modèles

Les modèles de machine learning sont chargés au démarrage de l'application et mis en cache pour des performances optimales.

```
class TimePredictor:
    def __init__(self, model_type='xgboost', random_state=42):
        self.model_type = model_type
        self.random_state = random_state
        self.model = None
        self.preprocessor = None
        self.feature_names = None
        self.target_col = 'TEMPS DISP'

    def load_model(self, path):
        """Charge un modèle pré-entraîné depuis le disque"""
        if not os.path.exists(path):
            raise FileNotFoundError(f"Model file not found at {path}")
        self.pipeline = joblib.load(path)
        self.preprocessor = self.pipeline.named_steps['preprocessor']
        self.model = self.pipeline.named_steps['regressor']
        print(f"Model loaded from {path}")
```

6.5.1.2 Préprocessing des données

Le service de prédiction inclut un pipeline de préprocessing robuste pour gérer les données d'entrée.

```
def preprocess_data(self, data):
    """Préprocesse les données d'entrée pour la prédiction"""
    df = data.copy()

    # Conversion des colonnes numériques
    for col in ['Nbr Plies', 'Longeur Tracé', 'Longeur Matela', 'Largeur']:
        if col in df.columns:
            df[col] = pd.to_numeric(df[col], errors='coerce')

    # Feature Engineering
    if 'Longeur Matela' in df.columns and 'Largeur' in df.columns:
        df['Surface_Matelas'] = df['Longeur Matela'] * df['Largeur']
    if 'Nbr Plies' in df.columns and 'Surface_Matelas' in df.columns:
        df['Volume_Matelas'] = df['Nbr Plies'] * df['Surface_Matelas']

    # Encodage des variables catégorielles
```

```

if 'Machine' in df.columns:
    df['Machine'] = df['Machine'].astype(str)

return df, features

```

6.5.2 Service d'Ordonnancement

6.5.2.1 Modélisation des contraintes

Le service d'ordonnancement utilise la programmation par contraintes pour modéliser le problème de planification.

```

class ProductionScheduler:
    def __init__(self, machines, operators, start_date=None):
        self.machines = machines
        self.operators = operators
        self.model = cp_model.CpModel()
        self.solver = cp_model.CpSolver()
        self.horizon = 30 # days
        self.start_date = start_date if start_date else datetime.now().date()
        self.intervals = {}

    def add_task(self, task_id, machine_id, operator_id,
                predicted_duration_minutes, due_date=None):
        """Ajoute une t'che au planificateur"""
        duration = int(predicted_duration_minutes)

        # Variables de temps
        start_var = self.model.NewIntVar(0, self.horizon * 24 * 60, f'start_{task_id}')
        end_var = self.model.NewIntVar(0, self.horizon * 24 * 60, f'end_{task_id}')

        # Variable d'intervalle
        interval_var = self.model.NewIntervalVar(
            start_var, duration, end_var, f'interval_{task_id}')
        self.intervals[task_id] = interval_var

        # Contrainte de délai
        if due_date:
            due_date_minutes = (due_date - self.start_date).days * 24 * 60
            self.model.Add(end_var <= due_date_minutes)

        return start_var, end_var, interval_var

```

6.6 Sécurité et Authentification

6.6.1 Système d'authentification

Le système utilise JWT (JSON Web Tokens) pour l'authentification et l'autorisation des utilisateurs.

6.6.1.1 Génération des tokens

```
def create_access_token(data: dict, expires_delta: Optional[timedelta] = None):
    """Crée un token JWT d'accès"""
    to_encode = data.copy()
    if expires_delta:
        expire = datetime.utcnow() + expires_delta
    else:
        expire = datetime.utcnow() + timedelta(minutes=15)
    to_encode.update({"exp": expire})
    encoded_jwt = jwt.encode(to_encode, SECRET_KEY, algorithm=ALGORITHM)
    return encoded_jwt
```

6.6.1.2 Vérification des tokens

```
def decode_access_token(token: str):
    """Décode et vérifie un token JWT"""
    try:
        payload = jwt.decode(token, SECRET_KEY, algorithms=[ALGORITHM])
        username: str = payload.get("sub")
        if username is None:
            raise APIException("Invalid token", 401)
        return username
    except JWTError:
        raise APIException("Invalid token", 401)
```

6.6.2 Contrôle d'accès basé sur les rôles (RBAC)

Le système implémente un contrôle d'accès basé sur les rôles pour gérer les permissions des utilisateurs.

Rôles définis :

- **Admin** : Accès complet à tous les endpoints
- **Manager** : Accès aux analytics et à la planification
- **Operator** : Accès limité aux prédictions et au planning
- **Viewer** : Accès en lecture seule aux analytics

6.7 Monitoring et Observabilité

6.7.1 Métriques de performance

Le système collecte des métriques détaillées sur les performances des modèles et des services.

6.7.1.1 Métriques des modèles

- Précision des prédictions (MAE, RMSE, R^2)
- Temps de réponse des prédictions
- Utilisation des ressources (CPU, mémoire)
- Fréquence des requêtes
- Taux d'erreur

6.7.1.2 Métriques des services

- Disponibilité des services (uptime)
- Temps de réponse des APIs
- Débit des requêtes
- Utilisation des ressources système
- Logs d'erreurs et d'exceptions

6.7.2 Alertes et notifications

Le système génère des alertes automatiques pour les événements critiques.

Types d'alertes :

- Dérive de modèle (model drift)
- Performance dégradée des services
- Erreurs de prédiction élevées
- Surcharge des ressources
- Échecs d'ordonnancement

6.8 Documentation Technique

6.8.1 Documentation API

La documentation API est générée automatiquement à partir des annotations FastAPI et est accessible via Swagger UI.

Endpoints de documentation :

- /docs : Interface Swagger UI
- /openapi.json : Spécification OpenAPI
- /redoc : Documentation ReDoc

6.8.2 Guide de déploiement

Le système est conçu pour être déployé dans des environnements conteneurisés avec Docker.

6.8.2.1 Configuration Docker

```
# Dockerfile pour l'API
FROM python:3.9-slim

WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt

COPY . .
EXPOSE 8000

CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]
```

6.8.2.2 Docker Compose

```
version: '3.8'
services:
  api:
    build: ./api
    ports:
      - "8000:8000"
    environment:
      - DATABASE_URL=postgresql://user:pass@db:5432/production_db
    depends_on:
      - db
      - mlflow

  frontend:
```

```
build: ./frontend
ports:
  - "3000:3000"
environment:
  - REACT_APP_API_URL=http://localhost:8000

db:
  image: postgres:13
  environment:
    - POSTGRES_DB=production_db
    - POSTGRES_USER=user
    - POSTGRES_PASSWORD=pass
  volumes:
    - postgres_data:/var/lib/postgresql/data

mlflow:
  image: mlflow/mlflow
  ports:
    - "5000:5000"
  volumes:
    - mlflow_data:/mlflow
```

6.8.3 Tests et validation

Le système inclut une suite de tests complète pour garantir la qualité et la fiabilité.

6.8.3.1 Tests unitaires

```
def test_time_prediction():
    """Test de prédiction de temps"""
    predictor = TimePredictor(model_type='xgboost')
    test_data = {
        'nbr_plies': 10,
        'longueur_trace': 1000.0,
        'longueur_matelas': 1100.0,
        'largeur': 150.0,
        'machine': 'Machine_A'
    }

    prediction = predictor.predict(test_data)
    assert prediction > 0
    assert isinstance(prediction, float)
```

6.8.3.2 Tests d'intégration

```
def test_schedule_optimization():
    """Test d'optimisation de planification"""
```



```

scheduler = ProductionScheduler(['M1', 'M2'], ['OP1', 'OP2'])

# Ajouter des t'ches
scheduler.add_task('T1', 'M1', 'OP1', 60)
scheduler.add_task('T2', 'M2', 'OP2', 90)

# Optimiser
result = scheduler.solve()

assert result is not None
assert len(result) == 2
assert result['T1']['machine'] == 'M1'
assert result['T2']['machine'] == 'M2'

```

6.9 Performance et Optimisation

6.9.1 Optimisations des modèles

Les modèles de machine learning sont optimisés pour des performances maximales en production.

6.9.1.1 Cache des prédictions

Les prédictions fréquemment demandées sont mises en cache pour réduire la latence.

```

from functools import lru_cache

@lru_cache(maxsize=1000)
def cached_prediction(nbr_plies, longueur_trace, longueur_matelas, largeur, machine):
    """Prédiction mise en cache pour les paramètres fréquents"""
    return predictor.predict({
        'nbr_plies': nbr_plies,
        'longueur_trace': longueur_trace,
        'longueur_matelas': longueur_matelas,
        'largeur': largeur,
        'machine': machine
    })

```

6.9.1.2 Prédiction en lot

Les prédictions en lot sont optimisées pour traiter plusieurs ordres simultanément.

```

def batch_predict(orders):
    """Prédiction optimisée en lot"""
    # Préparation des données en lot
    batch_data = pd.DataFrame(orders)

```

```
# Prédiction vectorisée
predictions = model.predict(batch_data)

# Formatage des résultats
results = []
for i, order in enumerate(orders):
    results.append({
        'order_id': order['order_id'],
        'predicted_time_minutes': predictions[i],
        'confidence_score': calculate_confidence(predictions[i])
    })

return results
```

6.9.2 Optimisations de l'ordonnancement

L'algorithme d'ordonnancement est optimisé pour traiter efficacement de grands volumes de tâches.

6.9.2.1 Parallélisation

Les calculs d'ordonnancement sont parallélisés pour améliorer les performances.

```
from concurrent.futures import ThreadPoolExecutor

def parallel_schedule_optimization(orders, machines, operators):
    """Optimisation parallèle de la planification"""
    # Division des ordres en groupes
    order_groups = split_orders_into_groups(orders, len(machines))

    # Optimisation parallèle de chaque groupe
    with ThreadPoolExecutor(max_workers=len(machines)) as executor:
        futures = []
        for i, group in enumerate(order_groups):
            future = executor.submit(
                optimize_group,
                group,
                machines[i],
                operators[i]
            )
            futures.append(future)

    # Collecte des résultats
    results = []
    for future in futures:
        results.extend(future.result())
```

```
return merge_schedule_results(results)
```

6.10 Évolutivité et Maintenance

6.10.1 Architecture évolutive

Le système est conçu pour être facilement extensible et maintenable.

6.10.1.1 Microservices

Chaque service est indépendant et peut être développé, testé et déployé séparément.

6.10.1.2 API Gateway

L'API Gateway centralise la gestion des requêtes et fournit des fonctionnalités transversales.

6.10.2 Maintenance des modèles

Les modèles de machine learning sont régulièrement mis à jour et réentraînés.

6.10.2.1 Monitoring de la dérive

Le système surveille automatiquement la dérive des modèles et déclenche des alertes.

```
def monitor_model_drift():
    """Surveille la dérive du modèle"""
    current_performance = evaluate_model_performance()
    baseline_performance = get_baseline_performance()

    drift_score = calculate_drift_score(current_performance, baseline_performance)

    if drift_score > DRIFT_THRESHOLD:
        send_alert("Model drift detected", drift_score)
        trigger_model_retraining()
```

6.10.2.2 Réentraînement automatique

Le système peut automatiquement réentraîner les modèles avec de nouvelles données.

```
def auto_retrain_model():
    """Réentraînement automatique du modèle"""
    # Collecte des nouvelles données
    new_data = collect_recent_data()

    # Validation des données
    if validate_data_quality(new_data):
        # Réentraînement
```

```
new_model = train_model(new_data)

# Validation du nouveau modèle
if validate_model_performance(new_model):
    # Déploiement du nouveau modèle
    deploy_model(new_model)
    log_model_update()
```

6.11 Conclusion

Ce chapitre a présenté l'architecture complète des services d'intelligence artificielle développés pour le système d'optimisation de la planification de production. Les spécifications techniques détaillées, l'interface utilisateur moderne, et l'intégration robuste des modèles de machine learning constituent une solution complète et évolutive.

L'approche microservices adoptée garantit la scalabilité et la maintenabilité du système, tandis que les fonctionnalités avancées comme le monitoring, les alertes et le réentraînement automatique assurent la fiabilité et la performance continue de la solution.

Le système est prêt pour le déploiement en production et peut être facilement étendu pour répondre aux besoins futurs de l'entreprise.



Conclusion générale

Ce rapport de recherche a présenté de manière exhaustive l'ensemble des travaux réalisés dans le cadre de ce projet de fin d'études, visant à digitaliser et optimiser les activités de planification au sein d'un atelier de coupe textile selon les principes fondamentaux de l'Industrie 4.0. La mobilisation rigoureuse de la méthodologie DMAIC (*Define, Measure, Analyze, Improve, Control*), issue du référentiel Lean Six Sigma, a permis d'identifier de manière systématique les dysfonctionnements structurels et organisationnels du processus existant, de formuler des solutions innovantes fondées sur la modélisation mathématique et l'intelligence artificielle, et de concevoir une architecture technologique intégrée combinant des modèles de machine learning prédictifs, des systèmes de capture de données (capteurs RFID), et des outils de visualisation dynamique en temps réel.

Ce travail de recherche appliquée a permis de développer une compréhension approfondie et critique des problématiques industrielles réelles, tout en consolidant de manière significative les compétences analytiques, techniques et organisationnelles nécessaires à la conduite de projets de transformation digitale. Le développement d'un algorithme de planification prédictive basé sur l'apprentissage automatique (machine learning), ainsi que la conception et l'implémentation d'un tableau de bord décisionnel connecté et interactif, illustrent de manière concrète et mesurable l'impact positif de la digitalisation sur la performance opérationnelle, la réactivité organisationnelle et la qualité de service rendue aux clients.

Bien que le projet soit parvenu à son terme académique avec la réalisation des objectifs fixés, il convient de reconnaître que cette recherche constitue davantage un point de départ qu'un aboutissement définitif. Plusieurs pistes d'amélioration et d'extension sont d'ores et déjà identifiées et peuvent être explorées dans le cadre de travaux futurs. Parmi ces perspectives, figurent notamment l'intégration d'un système intelligent de recommandation adaptatif basé sur l'apprentissage des préférences et des comportements des utilisateurs, permettant une personnalisation accrue de l'expérience utilisateur.

Au-delà des résultats académiques obtenus, ce projet de recherche ouvre des perspectives concrètes et prometteuses d'implémentation dans l'environnement industriel réel. Les gains opérationnels mesurés et quantifiés, incluant une amélioration de l'efficacité globale (+12% de TRS - Taux de Rendement Synthétique), une augmentation significative de la fiabilité des prévisions temporelles (+68%), et une réduction substantielle des temps de planification (-67%), démontrent le potentiel de transformation et d'amélioration continue offert par l'intégration des technologies de l'Industrie 4.0.

Des axes d'amélioration à moyen et long terme sont également envisagés pour renforcer davantage les capacités prédictives et décisionnelles du système. Parmi ces perspectives stratégiques, figurent notamment l'intégration d'un jumeau numérique

(*digital twin*) de l'atelier de coupe, permettant une simulation et une optimisation en temps réel des processus de production, ainsi que l'exploitation de modèles d'apprentissage profond (*deep learning*) pour des prédictions encore plus robustes et adaptatives face à la complexité croissante des environnements de production.

En somme, ce projet de recherche constitue une contribution significative à la transformation digitale des processus industriels dans le secteur textile tunisien, tout en mettant en valeur l'apport déterminant des technologies intelligentes et de l'intelligence artificielle dans la recherche d'excellence opérationnelle et de compétitivité durable. Les résultats obtenus et les méthodologies développées peuvent servir de référence pour d'autres entreprises du secteur textile confrontées à des enjeux similaires de modernisation et d'optimisation de leurs processus de production.



BIBLIOGRAPHIE

- [1] K. LICHTBLAU et al., « IMPULS-Industrie 4.0-Readiness, » *Impuls-Stiftung des VDMA, Aachen-Köln*, 2015.
- [2] L. D. XU, E. L. XU et L. LI, « Industry 4.0: state of the art and future trends, » *International Journal of Production Research*, t. 56, n° 8, p. 2941-2962, 2018.
DOI : [10.1080/00207543.2018.1444806](https://doi.org/10.1080/00207543.2018.1444806)
- [3] H. KAGERMANN, W. WAHLSTER et J. HELBIG, « Recommendations for implementing the strategic initiative INDUSTRIE 4.0, » *Final report of the Industrie 4.0 Working Group*, 2013.
- [4] S. STUDER et al., « Towards CRISP-ML(Q): A machine learning process model with quality assurance methodology, » *Machine Learning and Knowledge Extraction*, t. 3, n° 2, p. 392-413, 2021. DOI : [10.3390/make3020020](https://doi.org/10.3390/make3020020)
- [5] R. WIRTH et J. HIPPEL, « CRISP-DM: Towards a standard process model for data mining, » *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, t. 1, p. 29-39, 2000.
- [6] F. PROVOST et T. FAWCETT, *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. Sebastopol, CA : O'Reilly Media, 2013, ISBN : 978-1449361327.
- [7] T. CHEN et C. GUESTRIN, « XGBoost: A scalable tree boosting system, » *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 785-794, 2016. DOI : [10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785)
- [8] S. M. LUNDBERG et S.-I. LEE, « A unified approach to interpreting model predictions, » *Advances in Neural Information Processing Systems*, t. 30, p. 4765-4774, 2017.

- [9] L. PERRON, P. SHAW et V. FURNON, « Operations research and constraint programming at Google, » in *International Conference on Principles and Practice of Constraint Programming*, Springer, 2011, p. 2-2.
- [10] M. L. PINEDO, *Scheduling: Theory, Algorithms, and Systems*, 5th. Cham, Switzerland : Springer, 2016, ISBN : 978-3319265803. DOI : [10.1007/978-3-319-26580-3](https://doi.org/10.1007/978-3-319-26580-3)
- [11] A. ZHENG et A. CASARI, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. Sebastopol, CA : O'Reilly Media, 2018, ISBN : 978-1491953242.
- [12] I. GUYON et A. ELISSEEFF, « An introduction to variable and feature selection, » *Journal of Machine Learning Research*, t. 3, p. 1157-1182, 2003.
- [13] T. C. REDMAN, *Data Quality: The Field Guide*. Boston, MA : Digital Press, 2001, ISBN : 978-1555582517.
- [14] C. BATINI, C. CAPPIELLO, C. FRANCALANCI et A. MAURINO, « Methodologies for data quality assessment and improvement, » *ACM Computing Surveys (CSUR)*, t. 41, n° 3, p. 1-52, 2009. DOI : [10.1145/1541880.1541883](https://doi.org/10.1145/1541880.1541883)
- [15] N. GIFT et A. DEZA, *Practical MLOps: Operationalizing Machine Learning Models*. Sebastopol, CA : O'Reilly Media, 2020, ISBN : 978-1098103002.
- [16] T. HASTIE, R. TIBSHIRANI et J. FRIEDMAN, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd. New York, NY : Springer Science & Business Media, 2009, ISBN : 978-0387848570.
- [17] G. JAMES, D. WITTEN, T. HASTIE et R. TIBSHIRANI, « An Introduction to Statistical Learning: with Applications in R, » *Springer*, t. 112, 2013. DOI : [10.1007/978-1-4614-7138-7](https://doi.org/10.1007/978-1-4614-7138-7)
- [18] K. SCHWABER et J. SUTHERLAND, *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. Scrum.org, 2020. adresse : <https://scrumguides.org/>
- [19] M. COHN, *Agile Estimating and Planning*. Upper Saddle River, NJ : Prentice Hall, 2005, ISBN : 978-0131479418.



WEBographie

- [1] **Agile** : <https://monday.com/blog/fr/dev/gestion-de-projet-agile/>
- [2] **Scrum** : <https://www.atlassian.com/fr/agile/scrum/artifacts>
- [3] **Scrum** : <https://www.atlassian.com/fr/agile/scrum/roles>
- [4] **Architecture** : <https://prometteursolutions.com/blog/fr/types-darchitecture-logicielle-explication-et-bonnes-pratiques>
- [5] **MailTrap** : <https://mailtrap.io/blog/>
- [6] **MySql** : <https://www.mysqltutorial.org/>
- [7] **Visual Studio Code** : <https://www.blogdumoderateur.com/tools/visual-studio-code/>
- [8] **Letcode** : <https://www.letecode.com/tutoriels/tutoriels-laravel-9/controleur>
- [9] **Laravel** : <https://laravel.com/docs/11.x>
- [10] **websiterating** : <https://www.websiterating.com/fr/web-hosting/glossary/what-is-phpmyadmin/>
- [11] **softfluent** : <https://joshmartin.ch/en/technologies/css-3>

INTITULE DU PROJET DE PFE

Rapport de Stage de PFE DGT-ENIM, 2025

Ce projet de fin d'études s'inscrit dans le cadre de l'amélioration continue d'un atelier de coupe textile à travers une transformation digitale conforme aux principes de l'Industrie 4.0. En appliquant la démarche DMAIC, les dysfonctionnements critiques ont été identifiés puis traités par la mise en œuvre d'une solution technologique intelligente. L'approche inclut la modélisation des temps de matelassage, le développement d'un algorithme prédictif basé sur l'intelligence artificielle, et la création d'un tableau de bord de pilotage exploitant les données en temps réel. L'objectif est d'optimiser la planification, de réduire les pertes de temps, et d'améliorer la performance globale de l'atelier.

Mots clés : Industrie 4.0, atelier de coupe, intelligence artificielle, planification, tableau de bord, DMAIC

Abstract :

This final-year project aims to improve a textile cutting workshop through a digital transformation aligned with Industry 4.0 principles. Using the DMAIC methodology, the main dysfunctions were identified and addressed with a smart technological solution. The approach integrates matelassage time modeling, a predictive algorithm based on artificial intelligence, and the development of a real-time dashboard for operational monitoring. The proposed system enhances planning, reduces inefficiencies, and boosts the overall performance of the workshop.

Key-words : Industry 4.0, cutting workshop, artificial intelligence, planning, dashboard, DMAIC.