

Titre: Estimation des coûts de constructions de routes en utilisant
l'apprentissage automatique explicable

Auteur: Rosanne Larocque

Date: 2022

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Larocque, R. (2022). Estimation des coûts de constructions de routes en utilisant
l'apprentissage automatique explicable [Master's thesis, Polytechnique Montréal].
Citation: PolyPublie. <https://publications.polymtl.ca/10502/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/10502/>
PolyPublie URL:

**Directeurs de
recherche:** Quentin Cappart
Advisors:

Programme: Génie informatique
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Estimation des coûts de constructions de routes en utilisant l'apprentissage
automatique explicable**

ROSANNE LAROCQUE

Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Août 2022

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

**Estimation des coûts de constructions de routes en utilisant l'apprentissage
automatique explicable**

présenté par **Rosanne LAROCQUE**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Michel DESMARAIS, président

Quentin CAPPART, membre et directeur de recherche

Camélia DADOUCHI, membre

DÉDICACE

*À Benoit et Suzanne, mes chers parents,
pour votre support et votre amour inconditionnels.
Je vous aime. . .*

REMERCIEMENTS

Merci à toute l'équipe du ministère des Transports du Québec (MTQ) d'avoir permis la réalisation de ce projet. Merci de m'avoir fait confiance et de m'avoir supportée tout au long du projet. Merci pour les belles discussions ayant donné lieu à de nouvelles pistes et de nouvelles idées. Merci de permettre la publication des résultats obtenus. Un grand merci à Anne-Marie pour son travail initial d'exploration des données, d'ajout de variables et d'implémentation d'un modèle prédictif original. Merci à Jérôme pour ses innombrables conseils et idées. Merci à Alain pour sa supervision exemplaire et son oeil critique face à l'avancement du projet.

Merci à l'équipe technique du MTQ qui supporte la base de données BDP. Sans cette belle source d'informations, rien de tout cela n'aurait été possible.

Merci à mes parents, à mon frère et à ma soeur, de m'avoir toujours soutenue et encouragée au travers de mon parcours académique. Vos messages d'amour et d'encouragements m'ont poussée à me dépasser et m'ont toujours grandement touchée.

Surtout, un énorme merci à mon merveilleux directeur de recherche, Quentin Cappare, qui a su me transmettre sa passion et sa curiosité intellectuelle dès le premier cours qu'il m'a enseigné. Je suis extrêmement reconnaissante de tant de choses ; les opportunités qu'il m'a présentées, la confiance qu'il m'a témoignée, les multiples relectures, les conseils de rédaction, son sens de la rigueur hors pair. Merci de m'avoir permis de me surpasser et de m'avoir soutenue tout au long de cette aventure.

Sincèrement, merci à tous !

RÉSUMÉ

L'estimation préalable des coûts est une étape cruciale lors de la réalisation d'un projet de construction d'infrastructures de transports. Une estimation de qualité assure une meilleure planification de projet, une gestion des fonds adéquate et une prise de décision éclairée. En effet, il est commun de sélectionner une solution à un projet routier via la publication d'appels d'offres. Généralement, un projet regroupe différentes tâches et un appel d'offres sera émis pour chacune d'entre elles. Une compétition émerge entre les différents soumissionnaires. Cette compétition étant publique, elle peut potentiellement influencer les coûts suggérés. Ainsi, avant toute prise de décision, les chargés de projet doivent anticiper le coût raisonnable attendu pour un certain appel d'offres. Si le coût suggéré par les soumissionnaires est estimé à la baisse, le service offert risque d'être d'une qualité médiocre. À l'inverse, un coût surestimé se traduirait par une mauvaise gestion des fonds publics. Bref, on cherche à accepter un prix qui serait des plus justes. Les chargés de projet doivent également justifier leur choix de soumissionnaire de manière objective et compréhensible. Offrir une estimation préalable des coûts claire et justifiée représente un défi pour ces derniers. Au ministère des Transports du Québec, l'approche utilisée combine une analyse statistique des coûts historiques et l'avis d'experts en estimation. Or, le Vérificateur Général du Québec a soulevé des doutes quant à l'exactitude de ces estimations. De plus, ce dernier soulève son manque de transparence. Des approches plus modernes utilisent l'intelligence artificielle pour accomplir cette même tâche. Plus particulièrement, l'apprentissage automatique supervisé est fort utile pour faire de telles prédictions en présence de données numériques historiques. Toutefois, la conception d'un modèle d'apprentissage automatique étant à la fois performant et explicable est un défi dans l'état de l'art. Un autre défi lié à l'utilisation de l'apprentissage automatique est qu'un modèle entraîné à prédire les coûts pour une tâche en particulier ne sera pas forcément approprié pour une tâche tierce. La solution proposée affronte ces différents défis.

D'abord, ce mémoire propose un pipeline d'optimisation de la performance des modèles d'apprentissage automatique. Les premières étapes de ce pipeline visent à améliorer la qualité des données en vue de son utilisation avec le modèle étudié. Pour ce faire, on propose différentes techniques de mise à l'échelle des données, on procède à une sélection de variables rigoureuse via un algorithme d'élimination récursive des variables et on effectue une détection automatique des valeurs aberrantes. L'étape finale du pipeline d'optimisation permet de sélectionner les meilleurs hyperparamètres du modèle. Au travers de ce processus, chaque étape valide sa configuration à l'aide d'une validation croisée adaptée à notre problème. Six modèles d'apprentissage automatique seront optimisés pour chacune des tâches étudiées. Les

modèles étudiés sont l'arbre de décision, les forêts aléatoires, XGBoost, CatBoost, les réseaux de neurones, ainsi qu'un modèle hybride comprenant les cinq premiers.

Ensuite, on utilise une approche basée sur les Shapley Additives Values afin de rendre les prédictions des différents modèles plus transparentes. Deux types d'explications visuelles sont présentés. En premier lieu, l'explication détaille numériquement l'apport que chaque variable explicative a eu pour arriver à une estimation spécifique. Ce premier type d'explication visuelle illustre de manière élégante la logique de décision du modèle pour une estimation en particulier. En deuxième lieu, on offre une explication globale pour chaque tâche étudiée quant aux facteurs les plus influents au niveau du coût. Ce deuxième type d'explication permet de comparer l'importance des diverses variables au sein de multiples projets.

Les expériences ont été effectuées sur des ensembles de données réelles fournis par le ministère. Nous avons obtenu huit ensembles de données, correspondant à huit tâches différentes. Les données totalisent 11,646 contrats issus des années 2014 à 2021. Une instance de chacun des six modèles fut entraînée et optimisée de manière indépendante pour chaque tâche. Les résultats ont démontré que le modèle artificiel arrive à surpasser les performances de l'estimation humaine dans des proportions variant entre 3.42% et 31.56% sur des contrats non utilisés durant l'entraînement. De plus, les prédictions artificielles sont explicables et justifiées. Autant l'exactitude que les explications ont été validées par des experts du ministère. L'application concrète de cette solution permet à la fois d'améliorer l'exactitude de l'estimation préliminaire, ainsi que sa transparence. La solution complète pourrait également être utilisée dans divers domaines au sein du gouvernement du Québec. Par exemple, on pourrait l'appliquer pour estimer les coûts de construction de bâtiments ou encore de conception de logiciels.

ABSTRACT

Preliminary cost estimation is a predominant factor for the success of a road construction project. It supports the project planning, proper funds management and the decision-making process. The decision-making process usually goes by independent calls for tenders for each sub-task associated with a project. From this call emerges a competition between the various subcontractors. This competition being public, it impacts the price suggested by the several submissions. Prior to making a decision, project planners need to anticipate the fair price expected for a project. If the price given is too low, we risk having poor quality of service. Conversely, we risk paying an unnecessary amount if prices are higher than expected. Moreover, project planners need to justify their acceptance of a certain bidder to stakeholders in an objective and clear manner. This transparency is paramount to avoid illegal collusion. It also helps to identify the most influential factors in a project's cost. Having an estimation that is both of quality and transparent is a challenge for a project executive. In Quebec's Ministry of Transportation (MTQ), the task is carried out by a combination of an analysis of historical cost and expert knowledge. The latter method was however the object of doubts and questions regarding both its transparency and its correctness by the Auditor General of Quebec. A modern approach is to use a model based on artificial intelligence, and more precisely supervised learning. Having a supervised learning model that is both explainable and precise is nonetheless a challenge in literature. Another challenge is that a model trained to forecast cost for a specific sub-task may not be suitable for another sub-task. Our approach tackles those challenges.

First, we propose a full pipeline for optimizing performance of six machine learning models (decision tree, random forest, XGBoost, CatBoost, neural networks, and ensemble voters). The pipeline presents methods for enhancing data quality. In particular, we use scaling techniques, rigorous feature selection through the algorithm of recursive feature elimination and automatic outliers' detection. Besides, the final step of the pipeline ensures an optimal configuration of the model by tuning hyperparameters. Throughout this process, the quality of the model is ensured using a customized cross-validation based technique. Each one of the six models is independently trained and optimized for each sub-task of the experiment. Second, we use a method based on Shapley Additive Explanations to justify any of the prediction made by our models. Two types of visual explanations are provided. The first one describes the impact that each feature had for a specific estimation. The second one compares the impact of the features with one another among all the predictions.

Experiments have been done using real data provided by the Ministry covering eight different sub-tasks and totalling 11,646 contracts from 2014 to 2021. Results show that models built through our pipeline were able to outperform the precision of human estimation by a proportion varying between 3.42% to 31.56% on out of sample contracts, while providing a visual explanation for each estimation made. Both the estimations and their associated explanations were validated by specialists of the Ministry. By leveraging our solution, one can expect an enhancement of the quality and the transparency of the preliminary cost estimation in road construction projects. There is also great potential for using the same approach in other domains within the government of Quebec. For instance, it could be used for estimating cost of projects in domains such as building construction, or even software engineering.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES TABLEAUX	xii
LISTE DES FIGURES	xiv
LISTE DES SIGLES ET ABRÉVIATIONS	xvi
LISTE DES ANNEXES	xvii
CHAPITRE 1 INTRODUCTION	1
1.1 Contexte	1
1.1.1 Division par tâche	1
1.1.2 Appel d’offres et soumissions	2
1.1.3 Estimation préliminaire des coûts au sein du ministère	2
1.1.4 Coût unitaire	2
1.1.5 Transparence des estimations et Vérificateur général du Québec (VGQ)	3
1.2 Éléments de la problématique	4
1.2.1 Questionnements au VGQ	4
1.2.2 Solutions actuelles au MTQ	5
1.2.3 Intégration de l’intelligence artificielle (IA) à BDP	5
1.3 Contributions	6
1.4 Plan du mémoire	7
CHAPITRE 2 REVUE DE LITTÉRATURE	8
2.1 Intelligence artificielle en estimation de coût de projets de construction	8
2.2 Explicabilité en apprentissage automatique	11

CHAPITRE 3	MÉTHODOLOGIE	13
3.1	Ensembles de données	13
3.2	Variables externes ajoutées	17
3.2.1	Distance avec l’usine d’enrobage la plus proche	17
3.2.2	Prix historique du bitume	17
3.2.3	Salaires des employés de la construction	18
3.2.4	Indice économique (IQ-30)	18
3.3	Modèles d’apprentissage automatique	18
3.3.1	Arbre de décision (AD) et forêt aléatoire (FA)	19
3.3.2	XGBoost (XB) et CatBoost (CB)	19
3.3.3	Réseau de neurones	21
3.3.4	Modèle ensembliste hybride : EnsembleVoters (EV)	22
3.4	Méthodes de validation et d’évaluation de la performance	23
3.4.1	Métrique	23
3.4.2	Méthode de validation croisée	23
3.4.3	Méthode d’évaluation finale	24
3.5	Pipeline d’optimisation de la performance	25
3.5.1	Préparation des données (étape 1)	25
3.5.2	Sélection de la mise à l’échelle (étape 2)	26
3.5.3	Sélection des variables (étape 3)	27
3.5.4	Traitement des valeurs aberrantes (étape 4)	28
3.5.5	Configuration du modèle (étape 5)	29
3.6	SHAP : Explication des prédictions	29
3.6.1	Origine des valeurs de Shapley	30
3.6.2	Expliquer une prédiction selon la méthode SHAP	31
3.7	Implémentation : outils en lignes de commandes simples et réutilisables	32
3.7.1	Description de l’outil	32
3.7.2	Librairies utilisées	32
3.7.3	Utilisation du pipeline	33
3.7.4	Génération d’explications	36
CHAPITRE 4	ANALYSE DES RÉSULTATS	37
4.1	Comparaison des modèles d’IA étudiés	37
4.2	Sélection de variables avec la méthode RFE	40
4.3	Détection automatique des valeurs aberrantes	43
4.4	Analyse comparative des techniques de validation croisée	44

4.5 Comparaison de la performance de l'IA avec celle de l'estimation humaine . .	45
CHAPITRE 5 INTERPRÉTATION DES RÉSULTATS	47
CHAPITRE 6 CONCLUSION	50
6.1 Synthèse des travaux	50
6.2 Limitations de la solution proposée	51
6.3 Améliorations et travaux futurs	52
RÉFÉRENCES	54
ANNEXES	59

LISTE DES TABLEAUX

Tableau 3.1	Les huit tâches étudiées et leur nombre de données disponibles	13
Tableau 3.2	Les différentes variables explicatives utilisées	15
Tableau 3.3	Variables utilisées par tâche	16
Tableau 4.1	Comparaison du MAPE (%) sur l'ensemble de test (année 2021) des modèles d'IA par tâche	38
Tableau 4.2	Temps (sec.) requis pour l'entraînement des modèles sur les années 2014-2020 (Matériel : processeur 2,6 GHz Intel Core i5, 8GB de mémoire et aucun GPU)	39
Tableau 4.3	MAPE de validation moyen (%) des meilleurs modèles avant et après l'application de la méthode RFE	41
Tableau 4.4	Rang d'importance des variables externes ajoutées à l'expérience pour chaque tâche	42
Tableau 4.5	MAPE de validation moyen (%) des meilleurs modèles suite au retrait (ou non) des valeurs identifiées aberrantes	44
Tableau 4.6	MAPE (%) des meilleurs modèles selon les validations croisées aléatoires et temporelles	45
Tableau 4.7	Résumé des meilleures performances des modèles d'IA comparées à celles de l'estimation humaine	46
Tableau A.1	Valeurs de hyperparamètres du modèle EnsembleVoter entraîné sur la tâche couche de surface, enrobé. Les paramètres non mentionnés portent les valeurs par défaut.	59
Tableau A.2	Valeurs de hyperparamètres du modèle CatBoost entraîné sur la tâche couche de base, enrobé. Les paramètres non mentionnés portent les valeurs par défaut.	60
Tableau A.3	Valeurs de hyperparamètres du modèle forêt aléatoire entraîné sur la tâche armature. Les paramètres non mentionnés portent les valeurs par défaut.	60
Tableau A.4	Valeurs de hyperparamètres du modèle XGBoost entraîné sur la tâche déblais de 1 ^{re} classe. Les paramètres non mentionnés portent les valeurs par défaut.	61
Tableau A.5	Valeurs de hyperparamètres du modèle XGBoost entraîné sur la tâche déblais de 2 ^e classe. Les paramètres non mentionnés portent les valeurs par défaut.	61

Tableau A.6	Valeurs de hyperparamètres du modèle EnsembleVoters entraîné sur la tâche sous-fondation de chaussée. Les paramètres non mentionnés portent les valeurs par défaut.	62
Tableau A.7	Valeurs de hyperparamètres du modèle EnsembleVoters entraîné sur la tâche fondation de chaussée. Les paramètres non mentionnés portent les valeurs par défaut.	63
Tableau A.8	Valeurs de hyperparamètres du modèle réseau de neurones entraîné sur la tâche organisation de chantier. Les paramètres non mentionnés portent les valeurs par défaut.	63

LISTE DES FIGURES

Figure 3.1	Une représentation simplifiée d'un arbre de décision dans le contexte du problème d'estimation des coûts.	19
Figure 3.2	Une représentation simplifiée d'un modèle basé sur le <i>gradient tree boosting</i> . Il reçoit en entrée (Quantité = 20, Type de travail = Art, Année = 2021) et il estime le coût unitaire à 3\$, en fonction de la somme des prédictions des deux arbres.	20
Figure 3.3	Une représentation simplifiée d'un petit réseau de neurones recevant trois variables en entrée. Il contient une seule couche cachée et six neurones, en incluant les entrées et la sortie.	22
Figure 3.4	Séparation de l'ensemble de données dans les quatre phases de validation.	24
Figure 3.5	Schéma des six étapes du pipeline d'optimisation pour un ensemble de données.	25
Figure 3.6	Contenu du dossier de sortie généré à la suite de l'exécution du pipeline. Le nom des dossiers ou des fichiers se trouve au bas de la figure.	36
Figure 5.1	Analyse de la prédiction du prix unitaire d'une seule soumission du modèle FA pour la tâche Armature.	47
Figure 5.2	Analyse de l'impact SHAP du modèle FA pour la tâche Armature.	48
Figure B.1	Analyse de la prédiction du prix unitaire d'une seule soumission du modèle EnsembleVoters pour la tâche couche de surface, enrobé.	64
Figure B.2	Analyse de l'impact des variables du modèle EnsembleVoters pour la tâche couche de surface, enrobé.	64
Figure B.3	Analyse de la prédiction du prix unitaire d'une seule soumission du modèle CatBoost pour la tâche couche de base, enrobé.	65
Figure B.4	Analyse de l'impact des variables du modèle CatBoost pour la tâche couche de base, enrobé.	65
Figure B.5	Analyse de la prédiction du prix unitaire d'une seule soumission du modèle XGBoost pour la tâche déblai de 1 ^{re} classe.	66
Figure B.6	Analyse de l'impact des variables du modèle XGBoost pour la tâche déblai de 1 ^{re} classe	66
Figure B.7	Analyse de la prédiction du prix unitaire d'une seule soumission du modèle XGBoost pour la tâche déblai de 2 ^e classe.	67

Figure B.8	Analyse de l'impact des variables du modèle XGBoost pour la tâche déblai de 2 ^e classe.	67
Figure B.9	Analyse de la prédiction du prix unitaire d'une seule soumission du modèle EnsembleVoters pour la tâche sous-fondation de chaussée.	68
Figure B.10	Analyse de l'impact des variables du modèle EnsembleVoters pour la tâche sous-fondation de chaussée.	68
Figure B.11	Analyse de la prédiction du prix unitaire d'une seule soumission du modèle EnsembleVoters pour la tâche fondation de chaussée.	69
Figure B.12	Analyse de l'impact des variables du modèle EnsembleVoters pour la tâche fondation de chaussée.	69
Figure B.13	Analyse de la prédiction du prix unitaire d'une seule soumission du modèle réseau de neurones pour la tâche organisation de chantier.	70
Figure B.14	Analyse de l'impact des variables du modèle réseau de neurones pour la tâche organisation de chantier.	70

LISTE DES SIGLES ET ABRÉVIATIONS

AD	arbre de décision
BDP	Bordereau et Demande de paiement
CB	CatBoost
CMH	coût moyen historique
EV	EnsembleVoters
FA	forêt aléatoire
IA	intelligence artificielle
IF	IsolationForest
LOF	LocalOutlierFactor
MAPE	erreur absolue moyenne en pourcentage
MTQ	ministère des Transports du Québec
OCSVM	OneClassSVM
RFE	Recursive Feature Elimination
SHAP	SHapley Additive exPlanations
VGQ	Vérificateur Général du Québec
XB	XGBoost

LISTE DES ANNEXES

Annexe A	Hyperparamètres des meilleurs modèles	59
Annexe B	Explications SHAP locales et globales par tâches	64

CHAPITRE 1 INTRODUCTION

L'estimation préliminaire des coûts joue un rôle crucial quant au succès d'un projet de construction publique. Elle a notamment un impact important au niveau de la prise de décision des chargés de projet [1,2]. Un projet de construction regroupe généralement diverses tâches devant être prises en charge par différents type d'experts (architectes, ingénieurs civils, ingénieurs mécaniques, charpentiers, etc.). Un seul projet propose donc plusieurs contrats à différents fournisseurs. Ces derniers se retrouvent donc dans une compétition dans laquelle le coût est un critère prépondérant. Pour favoriser leur sélection, il y a un risque que les fournisseurs manipulent les prix soumis de sorte à les diminuer, ce qui peut mener à un éventuel dépassement des coûts ou à un service de qualité médiocre. Une manière d'encadrer les prix soumis est justement d'établir une estimation maison faite par ceux qui initie ledit projet de construction. Ainsi, une estimation préalable du juste prix permet une prise de décision éclairée et d'éviter des solutions inutilement dispendieuses. Cela dit, afin de garantir que cette estimation soit juste et objective, une justification claire doit l'accompagner. Une bonne justification permet également de solidifier le lien de confiance qui unit les gouvernements et les citoyens quant à la bonne gestion des fonds publics. Dans le cadre de cette recherche, nous étudierons le cas des projets routiers pilotés par le gouvernement du Québec. Plus particulièrement, ces projets sont de la responsabilité du ministère des Transports du Québec (MTQ).

1.1 Contexte

Cette section introduit les concepts nécessaires à la compréhension du processus de prise de décision pour un projet d'infrastructure routière au sein du MTQ.

1.1.1 Division par tâche

Un projet de construction d'infrastructures de transport regroupe généralement diverses tâches. Par exemple, la construction d'une nouvelle route requiert d'établir les fondations de la chaussée, qui sera ensuite recouverte d'asphalte, puis on pourrait y ajouter des panneaux de signalisation, des câbles pour l'électrification et un garde fou [3]. Ces tâches impliquent des experts de différents domaines. Chaque tâche peut donc être la responsabilité de diverses équipes, ce qui implique qu'une solution indépendante est donnée pour chacune d'entre elles.

1.1.2 Appel d'offres et soumissions

Au MTQ, chaque tâche sera ultimement attribuée à divers entrepreneurs privés. Pour dénicher d'éventuels entrepreneurs, il est commun au sein du MTQ de passer par le biais d'appels d'offres. Il s'agit d'une invitation officielle décrivant une tâche à accomplir. L'invitation comprend entre autres des informations sur le chantier de construction, la nature des travaux à effectuer, la date envisagée et les matériaux utilisés. Les informations peuvent toutefois varier selon la nature de la tâche sous-jacente à l'appel d'offres. Selon ces informations, tout entrepreneur a la possibilité de soumettre une solution détaillée et d'y apposer un prix. Par cette action, il propose une soumission. Les chargés de projet du MTQ étudieront ensuite les solutions proposées et le coût sera un facteur prépondérant dans la prise de décision.

1.1.3 Estimation préliminaire des coûts au sein du ministère

Les soumissionnaires privés ne seront pas les seuls devant estimer un coût pour une solution. Les experts du MTQ devront également se mettre à la tâche afin qu'ils puissent prendre une décision éclairée quant aux différentes propositions obtenues. En effet, étant donné la nature publique de l'appel d'offres, une concurrence émerge entre les soumissionnaires ce qui peut influencer les prix suggérés. Un prix anormalement bas par rapport à ce qui est attendu peut résulter en une qualité de service médiocre ou encore à un éventuel dépassement des coûts. À l'inverse, il est également indésirable d'accepter un prix trop élevé. On cherche donc à se rapprocher d'un prix des plus justes.

1.1.4 Coût unitaire

Il existe de nombreuses manières de calculer le coût d'un service. En l'occurrence, on pourrait directement estimer son coût total. Une autre méthode est d'estimer son coût unitaire et de le multiplier avec la quantité requise. Au sein du MTQ, c'est plutôt cette dernière qui est utilisée. Avant de publier un appel d'offres, le ministère a pour responsabilité d'estimer la quantité requise pour un service, qui prend souvent la forme d'un volume. Par exemple, cette quantité pourrait correspondre à la taille d'une nouvelle route en mètres cubes. Cette quantité est mise à la disposition des entrepreneurs, qui devront se servir de cette information afin d'établir le coût unitaire qu'ils proposent. Ainsi, autant pour les soumissionnaires que pour les experts du MTQ, les prix estimés sont en réalité des coûts unitaires.

1.1.5 **Transparence des estimations et Vérificateur général du Québec (VGQ)**

Au sein du gouvernement du Québec, il existe un office indépendant qui est responsable de valider l'utilisation adéquate des fonds publics. Pour ce faire, elle effectue des audits. Ces audits sont suivis par la rédaction de rapports contenant des constatations et des recommandations qui permettraient d'assurer une gestion des fonds des plus saines et juste. Cet office assure agir de manière intègre, et ce, pour le pur bénéfice des citoyens et des citoyennes. Il s'agit du Vérificateur général du Québec [4]. Il est d'ailleurs de sa responsabilité de s'assurer qu'une bonne gestion des fonds est faite au niveau du MTQ. Ainsi, l'exactitude de l'estimation du MTQ n'est pas la seule caractéristique souhaitée. On s'attend également à ce que l'estimation du juste prix soit objective et bien justifiée, afin d'éviter tout risque de corruption, encore bien présent à ce jour [5].

1.2 Éléments de la problématique

Dans cette gestion contractuelle, les défis abordés par ce mémoire concernent en particulier les estimations préalables faites par le MTQ. Ces défis ont été mis en évidence suite à un audit du VQG, qui sera d'ailleurs détaillé dans cette section. Ensuite, on y décrira les solutions actuellement employées par le MTQ pour y répondre, leurs limites, ainsi que les difficultés entourant l'amélioration des solutions actuelles.

1.2.1 Questionnements au VGQ

Lors de son rapport de 2017, le VGQ a mis en doute le processus de gestion contractuelle du MTQ [6]. En particulier, le rapport soulève des doutes quant à la rigueur des estimations émises par le MTQ. De plus, on y mentionne le manque de profondeur dans l'analyse de marché préalable ayant pour but de permettre une plus grande concurrence. En effet, plusieurs contrats parmi ceux vérifiés ont été attribués à un fournisseur alors qu'il était le seul concurrent. D'ailleurs, la justification relative à avoir accepté de tels contrats n'était pas suffisante. En résumé, on s'interroge autant la transparence que l'exactitude des estimations préalables faites au MTQ.

Concernant ces constatations quant à l'estimation des coûts, le VGQ présente les recommandations suivantes :

1. « Produire une estimation rigoureuse des coûts de chaque contrat afin de s'assurer que le prix soumis est juste et raisonnable et que, ultimement, les fonds publics sont utilisés adéquatement » [6] ;
2. « Encadrer adéquatement les travaux confiés à des firmes externes, notamment à l'égard de la conception des plans et devis (estimation des coûts des contrats) et de la surveillance des chantiers (gestion des dépenses supplémentaires), afin d'assurer l'intégrité du processus de gestion contractuelle » [6] ;
3. « Approfondir ses analyses pour connaître les marchés afin d'adapter ses stratégies d'intervention, et ce, dans le but de permettre une plus grande concurrence et une diminution des situations où il y a une seule offre conforme » [6]

En addition à ces recommandations, il va de soi qu'une justification adéquate devra accompagner les estimations. Ces recommandations sont intéressantes d'un point de vue plus global puisqu'elles visent des problématiques communes à tout projet gouvernemental ayant un processus de gestion contractuelle similaire.

1.2.2 Solutions actuelles au MTQ

Une technique mise en place pour répondre à la première et à la deuxième recommandation fut l'utilisation du coût moyen historique (CMH). Pour obtenir cette information, le MTQ possède une banque de données nommée Bordereau et Demande de paiement (BDP). Chaque ligne de cette base de données traduit les informations contenues dans un appel d'offres, le prix estimé par le ministère et celui soumis par un fournisseur. Il y a une ligne par soumissionnaire pour un certain appel d'offres, donc plusieurs lignes sont associées à un même appel. C'est grâce aux informations recueillies dans BDP que l'on peut calculer le CMH. L'avis d'un expert est finalement requis avant de produire l'estimation du juste prix unitaire.

Cette méthode d'estimation n'est pas suffisante pour répondre aux trois recommandations du VGQ. D'abord, on ne fait qu'une moyenne, ce qui laisse quelques doutes quant à la rigueur demandée par la première recommandation. En effet, c'est une estimation plutôt simple, car la distribution des coûts ne suit probablement pas la moyenne historique de manière uniforme et cet outil ne tient pas compte de l'ensemble des éléments spécifiques qui peuvent influencer les prix des soumissions. La deuxième recommandation est somme toute respectée en ce qui concerne l'estimation, puisqu'elle est dorénavant faite par le MTQ, encadrant ainsi les éventuels fournisseurs. Cela dit, cette méthode ne constitue pas une analyse de marché approfondie, écartant la troisième recommandation. De plus, aucune justification n'est présentée quant à la logique d'une prédiction, ce qui rend son interprétation ardue.

1.2.3 Intégration de l'intelligence artificielle (IA) à BDP

Bien que des méthodes telles que celle basée sur le CMH soient encore communément utilisées, de plus en plus d'approches se basent sur l'utilisation de l'IA. Plus spécifiquement, on utilise l'apprentissage automatique supervisé [7, 8]. L'avantage espéré en utilisant l'IA est d'obtenir une estimation des plus précises.

Obtenir une estimation plus précise ne veut toutefois pas dire qu'elle sera plus transparente. En IA, on dit d'un modèle qu'il est explicable s'il est en mesure de décrire sa logique et sa prise de décision de manière intelligible à une personne. On réfère à ce concept en le nommant *apprentissage machine explicable* [9] (traduit de l'anglais). Le niveau d'explicabilité de l'IA varie grandement d'un modèle à l'autre. En général, il varie de manière inversement proportionnelle à son niveau de performance. De cette manière, plus la qualité des prédictions est bonne, moins le modèle sous-jacent est explicable. Par exemple, l'arbre de décision [10] est une approche simple et facile à expliquer. À l'inverse, les réseaux de neurones profonds [11] sont des modèles très complexes. Concevoir un modèle à la fois performant et explicable est

donc un défi dans l'état de l'art.

Un autre défi est qu'un modèle entraîné à estimer le coût unitaire de travaux d'une certaine tâche ne sera pas nécessairement aussi précis pour estimer les coûts d'une tâche tierce. En effet, il est fort probable que la distribution des prix soumissionnés soit différente d'une tâche à l'autre, notamment puisqu'elles ne nécessitent pas les mêmes spécialistes. De plus, certaines variables ne sont utilisées qu'avec une certaine tâche et sont sans importance pour les autres.

1.3 Contributions

Le lien de confiance du public envers le gouvernement concernant la bonne gestion des fonds a toujours été fragile. En effet, il est arrivé des incidents de corruption déplorables à travers le temps. Une estimation objective et juste peut donc d'être faite afin d'encadrer les prix soumis par des particuliers lors des appels d'offres. Ce n'est toutefois pas une tâche simple. L'intelligence artificielle pourrait être un outil intéressant pour effectuer cette tâche, mais l'estimation doit être bien justifiée. Cette étude permet de démontrer comment l'apprentissage automatique peut proposer une estimation à la fois précise et justifiée. Cette solution peut d'ailleurs s'étendre à tout domaine public qui effectue une gestion contractuelle via un processus d'appels d'offres. Elle vise permettre une meilleure gestion des fonds publics, tout en solidifiant le lien de confiance unissant les citoyens à leur gouvernement. Cette recherche propose donc une alternative intéressante aux méthodes d'estimation actuelles.

Au sein du MTQ, notre cas d'étude, ces contributions se traduiront par les apports suivants :

1. Une bonification de l'information contenue dans BDP en ajoutant des variables externes aux soumissions ;
2. La conception d'un pipeline d'optimisation de la performance des modèles, qui est réutilisable, puisqu'il est agnostique au modèle et aux données ;
3. L'application du pipeline d'optimisation à six modèles d'IA et à huit tâches, de manière indépendante. Les modèles étudiés sont l'arbre de décision [10], la forêt aléatoire [12], XGBoost [13], CatBoost [14], les réseaux de neurones [11], et un modèle hybride basé sur une méthode ensembliste [15] ;
4. Une première explication basée sur les Shapley Additives Explanations (SHAP) [16] justifiant visuellement chaque estimation faite par un modèle artificiel ;
5. Une deuxième explication basée sur les SHAP permettant d'analyser globalement les divers facteurs qui influencent le prix unitaire pour une tâche ;

1.4 Plan du mémoire

Ce manuscrit présentera d’abord une revue de littérature (chapitre 2) sur les algorithmes d’apprentissage automatique prometteurs pouvant s’appliquer aux données de BDP. Selon ces algorithmes, diverses techniques pour améliorer l’explicabilité des modèles seront également revues. Le chapitre 3 présentera les ensembles de données, les modèles d’IA qui seront étudiés, le pipeline d’optimisation de la performance des modèles et les métriques utilisées. La méthodologie complète d’expérimentation y sera donc détaillée. Le chapitre 4 vise d’abord à présenter les différents résultats obtenus avec les six modèles étudiés. On y observe ensuite l’apport des différentes bonifications proposées dans notre pipeline d’optimisation. Ce quatrième chapitre se termine en comparant la performance du meilleur modèle pour chaque tâche avec celle obtenue par l’estimation humaine. Le chapitre 5 présentera les deux types d’explications proposées, basées sur les méthodes SHAP. Le rapport se terminera au chapitre 6, dans lequel on met en lumière les faits saillants, les limites associées aux méthodes utilisées et les pistes possibles pour la suite du projet.

CHAPITRE 2 REVUE DE LITTÉRATURE

Le présent mémoire s’articule autour de deux grands concepts : l’utilisation de l’IA en estimation de coût et l’explicabilité en IA. En effet, puisqu’il s’agit d’un problème d’estimation de coût en construction, il est pertinent de porter un premier regard sur l’utilisation de l’intelligence artificielle dans ce domaine spécifique, qui sera présenté dans une première partie de cette revue. Cette dernière permettra de déceler les modèles susceptibles d’être les plus performants pour ce problème. Cela dit, il existe souvent une tension entre performance et explicabilité en apprentissage automatique. Il est donc probable que les méthodes les plus performantes dans l’état de l’art soient également les plus complexes. Il importe donc de porter un regard particulier sur l’explicabilité en IA. La littérature à ce sujet contient notamment des manières de rendre les méthodes complexes plus transparentes. Ce sera présenté dans la deuxième partie de la revue.

2.1 Intelligence artificielle en estimation de coût de projets de construction

L’estimation des coûts d’un projet de construction est un facteur important pour sa réussite. L’impact est encore plus grand lorsqu’il s’agit d’un projet public, puisque la gestion des fonds et des budgets se complexifie davantage. Il n’est donc pas surprenant que ce sujet en ait intéressé plus d’un. Cette tâche n’est toutefois pas si simple malgré sa grande importance.

Depuis les années 1970, beaucoup de projets ont utilisé des approches basées sur la régression linéaire [1, 2, 17]. C’est vers les années 1990 que des méthodes plus proches de celles d’IA modernes ont commencé à être utilisées à cette fin. En l’occurrence, plusieurs auteurs découvrent l’avantage d’effectuer des estimations en utilisant les réseaux de neurones (RN). Les résultats de [7] démontrent que les RN avaient le potentiel de dépasser les performances des précédentes méthodes d’estimation de coût de construction de pompes. Similairement, [18] décrit une expérience dans laquelle les RN sont utilisés pour prédire les coûts de construction de bâtiments. Leurs résultats ont montré que les performances du RN se rapprochent de celles d’un système expert. Or, alors qu’un système expert requiert l’intervention de spécialistes, l’algorithme d’apprentissage derrière les RN est autonome. Le potentiel de ces réseaux a également été comparé avec celui de la régression linéaire multiple pour prédire le coût de tuyau en acier. Encore une fois, cette expérience menée dans [19] montre que les RN ont surpassé les performances de la régression linéaire multiple.

En 2004, l’engouement se poursuit alors que [20] compare les performances des RN, de la

régression linéaire, et d'une méthode basée sur le «Case-based reasoning» (CBR) afin d'estimer les coûts de projets de constructions. Les méthodes basées sur le CBR visent à estimer un nouveau coût en fonction des projets similaires précédents. Ce principe imite en quelque sorte le raisonnement d'un humain se basant sur ses expériences antérieures. Les conclusions sont similaires à celles des travaux antérieurs. Ils ont constaté que les RN ont donné la meilleure performance, suivi de près par le CBR. Les auteurs ont toutefois conclu que la méthode CBR était la plus appropriée puisqu'elle était plus transparente et plus facile à appliquer. En effet, l'article de 2004 observe déjà que les prédictions d'un RN sont peu transparentes.

Malgré ces observations quant au manque de transparence des RN, une revue de littérature systématique de 2017 sur les méthodes d'estimation de coût en infrastructures de transport recense encore une fois les RN comme étant la deuxième technique la plus utilisée [21]. Il faut croire que l'explicabilité n'est pas une caractéristique recherchée de manière universelle, car plusieurs ne semblent que considérer la précision des estimations. En première position, ce sont les techniques dites paramétriques, ce qui inclut entre autres la régression linéaire. Depuis cette revue, la tendance à vouloir explorer les RN semble se maintenir. En 2018 et en 2019, un grand nombre d'expérimentations les choisissent pour estimer le coût des infrastructures de transport. C'est notamment le cas de l'Inde [22], de la Grèce [23], des Philippines [24] et même de la Croatie [25]. Ces auteurs basent leur choix d'algorithmes sur la littérature existante et sur le fait qu'ils estiment économiser du temps en prétraitement de données avec les RN. Ils ne semblent toutefois pas accorder une attention particulière à l'explicabilité.

Plus récemment, plusieurs auteurs sont revenus sur la tendance que peuvent avoir les RN à faire du surapprentissage [26]. Il s'agit d'un phénomène observable lorsqu'un modèle d'IA apprend par cœur l'ensemble de données qui lui est fourni lors de son entraînement. Il n'apprend à ce moment pas la réelle relation entre les variables explicatives et le coût. Ainsi, la précision de l'estimation sera pratiquement parfaite si évaluée sur une donnée contenue dans l'ensemble d'entraînement original, mais fort probablement erronée sur une nouvelle donnée. C'est donc un comportement à éviter. Le surapprentissage a particulièrement tendance à survenir lorsqu'un modèle très complexe est entraîné sur un ensemble de données relativement petit (donc facile à apprendre par cœur), ce qui a tendance à être le cas dans le domaine de la construction. Pour pallier cela, certains ont trouvé succès dans des approches basées sur des modèles hybrides [26–28]. En l'occurrence, [26] combine un modèle linéaire et un RN non linéaire afin d'effectuer une prédiction finale. Cette technique permet de profiter des avantages respectifs de chacun. En l'occurrence, le modèle linéaire tend plus rarement à faire du surapprentissage, mais les RN sont plus précis. Ce modèle hybride a su surpasser les performances des deux autres pris de manière individuelle. [27,28] ont mené des expériences similaires avec différents modèles linéaires et ont tiré les mêmes conclusions.

Outre les RN, une revue de l'état de l'art de 2020 recense une vingtaine de modèles d'intelligence artificielle ayant été utilisés afin d'estimer les coûts de projets de constructions [29]. Cette étude comparative identifie l'algorithme XGBoost [13] comme étant le plus performant. Introduit pour la première fois en 2016, ce modèle se base d'abord sur le concept d'arbres de décision, mais y ajoute un concept d'ensemble. Au lieu de n'utiliser qu'un seul arbre, l'idée est d'en utiliser plusieurs qui se complètent mutuellement. C'est l'idée du *Gradient Tree Boosting* [30]. Cette idée, combinée avec divers ajouts permettant une régularisation, en fait un estimateur robuste qui s'adapte bien à différents types de données. En effet, bien que les RN connaissent un succès sans précédent dans plusieurs applications, l'algorithme XGBoost est mieux adapté pour des données mixtes contenant des variables continues et des variables discrètes. De plus, ce modèle est plus facile à optimiser et plus rapide à entraîner [31]. Un autre algorithme un peu moins connu, mais également basé sur le *Gradient Tree Boosting*, est encore mieux adapté lorsque l'on rencontre beaucoup de données qualitatives. Il s'agit de la technique CatBoost [14]. Elle a notamment surpassé la performance des RN et de l'algorithme XGBoost dans le domaine de la finance [32].

Les expériences mentionnées jusqu'à présent partagent toutes plus ou moins la même manière d'évaluer la performance. Il s'agit de l'utilisation d'ensembles d'entraînement et de validation. Le principe général, peu importe la métrique qui sera comparée (erreur absolue, pourcentage d'erreur, etc.), est de séparer l'ensemble de données initial en deux. La première partie, la plus grande, sera utilisée pour entraîner le modèle. La deuxième partie est alors utilisée pour mesurer sa performance. Ainsi, on utilise le modèle pour faire des prédictions sur cette deuxième partie et on en calcule l'erreur. Cette manière de faire vise à observer la capacité de généralisation du modèle. Cependant, les précédentes expériences séparaient les données de manière aléatoire. Or, dans l'optique où l'on voudrait utiliser un modèle artificiel pour des projets futurs, il est plus adéquat de séparer les données en fonction de l'année des projets. Ce sont d'ailleurs des méthodes d'évaluation suivant ce principe qui sont utilisées par les deux plus récents projets consultés dans le cadre de cette revue [26, 33]. L'objectif de [26] est légèrement différent du nôtre dans la mesure où ils tentent d'estimer les indices de coût de construction pour différentes villes plutôt que le coût associé à un projet en soi. Toutefois, leur méthodologie rigoureuse est originale. Leur méthode d'évaluation de la performance est innovatrice. Ils évaluent les modèles selon trois angles : la performance court terme (12 mois), moyen terme (36 mois) et long terme (60 mois). Ainsi, l'ensemble sur lequel est calculée la performance correspond respectivement aux données de la dernière année, des trois dernières années et des cinq dernières années de l'ensemble initial. Le modèle est entraîné sur les données restantes dans les trois cas. Bien que cette méthode soit intéressante d'un point de vue analytique, la performance court terme est sans doute la plus appropriée. En pratique, on

mettrait sans doute le modèle à jour fréquemment, selon les nouvelles données cumulées. Il est donc peu utile de connaître les performances d'un modèle entraîné avec des données vieilles de trois ans. L'approche proposée dans [33] suit davantage l'optique de l'évaluation à court terme. Elle est cependant légèrement plus complexe et s'inspire de la technique de validation croisée. C'est d'ailleurs celle-ci qui sera utilisée dans la présente expérience. Elle sera donc expliquée et détaillée au chapitre 3. Outre leur méthode de validation, l'expérience de [33] est fortement similaire à la présente. Il s'agit d'estimer les coûts d'items en construction d'auto-routes en utilisant l'IA. Comme plusieurs autres, ils ont comparé divers modèles linéaires et non linéaires (incluant les RN). L'originalité de cet article, en plus d'utiliser une manière intéressante d'évaluer les modèles, est d'avoir porté une attention particulière à l'amélioration de la qualité des données. En l'occurrence, trois méthodes de sélection de variables explicatives sont implémentées et comparées. De plus, les résultats obtenus indiquent que les modèles linéaires ont mieux performé que les RN. Il s'agit d'un résultat s'opposant à la littérature antérieure. Il est possible que l'attention particulière accordée à l'amélioration de la qualité des données ait permis d'obtenir un tel résultat. Cela implique qu'il ne faut pas écarter de sitôt des modèles simples comme l'arbre de décision [10].

En résumé, il n'existe actuellement pas de consensus clair dans la littérature pointant vers un modèle d'IA en particulier comme étant le meilleur dans tous les cas. D'autant plus que les méthodes d'évaluation et de prétraitement de données varient d'une expérience à l'autre, ce qui rend la comparaison difficile. Le RN reste toutefois un incontournable. Les modèles basés sur le *gradient tree boosting* semblent également prometteurs, tout comme les modèles hybrides. Finalement, étant donné l'importance accordée à l'explicabilité, des modèles d'apprentissage automatique plus simples semblables à l'arbre de décision [10] seront à étudier.

2.2 Explicabilité en apprentissage automatique

Les modèles prometteurs dont nous avons discuté jusqu'à présent n'ont pas été conçus dans le but d'atteindre un haut niveau de transparence. Il faut effectivement utiliser des stratégies additionnelles si l'on veut rendre ces derniers compréhensibles et explicables. En IA, l'explicabilité caractérise la capacité d'un algorithme à décrire sa logique et sa prise de décision de manière intelligible à une personne [9].

Il existe des méthodes qui ne fonctionnent qu'avec certains modèles. Par exemple, on peut utiliser le principe de distillation avec les RN [34]. Il s'agit de tenter de copier le comportement d'un réseau avec un modèle plus simple, de sorte à améliorer les performances de ce dernier. Ainsi, le modèle utilisé pour les prédictions sera le modèle plus simple. L'équipe de Google

Brain a tenté ce principe pour traduire un RN en arbre de décision [35]. Cette méthode connaît toutefois des limites. En effet, l'arbre de décision issu de la distillation a permis une amélioration de 2% par rapport à celui entraîné de manière standard sur un ensemble de données plutôt simple. Le RN original surpasse également la performance de l'arbre distillé. On ne réduit donc pas beaucoup la tension entre performance et explicabilité en utilisant la distillation. D'ailleurs, cette technique ne fonctionne que pour les RN. Similairement, des algorithmes permettent de transformer un groupe d'arbres de décision (forêts aléatoires) en un seul arbre. De plus, l'algorithme garantit que l'arbre unique performera aussi bien que la forêt aléatoire pour l'ensemble de données utilisé [36]. Toutefois, ce dernier est également limité aux modèles basés sur un ensemble d'arbres de décision.

D'autres méthodes permettent d'expliquer la décision d'un modèle peu importe la nature de ce dernier. On nommera ces dernières les techniques agnostiques au modèle. D'ailleurs, un article unifie plusieurs de ces méthodes sous une même famille. Les méthodes sélectionnées se basent sur ce que l'on appelle les valeurs de Shapley [37]. Il s'agit d'une forme de pondération accordée pour chaque variable pour une prédiction et un modèle donné. Pour calculer cette pondération, ces algorithmes retirent différentes variables successivement et observent comment la prédiction du modèle se modifie selon la présence ou l'absence de différentes variables. L'article nomme cette famille SHAP (SHapley Additive exPlanations) et y identifie six différentes méthodes y appartenant [16]. Son grand avantage est de pouvoir s'adapter à différents modèles, ce qui est bien utile si on veut comparer différents modèles entre eux. Bien qu'elle soit intéressante pour expliquer l'apport de chaque variable explicative pour une prédiction donnée, elle n'explique pas réellement le processus de décision du modèle de manière globale. L'explication de ce processus est d'ailleurs plutôt ardue lorsqu'on utilise un modèle n'ayant pas été conçu pour être transparent. C'est pourquoi certains auteurs sont réticents à utiliser des modèles complexes et de les rendre explicables par la suite [38]. Selon ces auteurs, il serait préférable d'utiliser des modèles qui sont plus simples et naturellement explicables. Il s'agirait ensuite d'améliorer la qualité ou d'augmenter la quantité des données plutôt que la complexité de notre modèle pour avoir un modèle performant et explicable. Les arbres de décision classiques sont de bons exemples de modèles simples. Le modèle étant fixe, peu importe l'entrée, il est facile de suivre le cheminement de cet algorithme. Ainsi, si tous les problèmes pouvaient s'adapter à un tel modèle pour le rendre aussi performant que ses semblables qui sont plus complexes, l'explicabilité ne serait pas un problème. Cela dit, il est plutôt rare en pratique que ce modèle surpasse les modèles plus modernes.

En somme, vu la variété des modèles qui seront étudiés dans le cadre de ce mémoire, il est plus approprié d'utiliser des techniques agnostiques au modèle. Ainsi, les méthodes issues de la famille SHAP seront utilisées.

CHAPITRE 3 MÉTHODOLOGIE

Ce troisième chapitre présente chacune des étapes du processus d'expérimentation. On y présente d'abord l'ensemble de données et les variables externes qui y ont été ajoutées. Ensuite, on définit les six modèles d'IA qui seront étudiés, ainsi que les différentes méthodes permettant d'en évaluer la performance. La section se poursuit en présentant le pipeline créé dans le but d'améliorer la qualité des données et la performance des modèles. On décrit également les méthodes d'explication SHAP. Finalement, on présente l'implémentation d'un outil en lignes de commandes permettant de reproduire ces expériences dans différents problèmes d'apprentissage automatique.

3.1 Ensembles de données

Un des défis liés à l'utilisation de l'apprentissage supervisé est qu'un modèle entraîné pour estimer les coûts d'une tâche en particulier ne sera pas nécessairement adapté à prédire les coûts d'une tâche tierce. De plus, comme indiqué au tableau 3.3, les variables explicatives varient d'une tâche à l'autre. La même instance d'un modèle ne pourrait pas être utilisée pour estimer les coûts de deux tâches n'ayant pas les mêmes variables. Pour pallier cela, une solution simple est de diviser les soumissions par tâche, de sorte à obtenir un ensemble de données pour chacune d'entre elles. Cela implique toutefois qu'un modèle devra être optimisé pour chaque tâche, ce qui décuple le travail nécessaire. Dans le cadre de cette expérience, huit tâches seront étudiées. Elles ont été sélectionnées pour leur abondance de données. Le tableau 3.1 énumère ces dernières, ainsi que le nombre de données disponibles pour chacune.

TABLEAU 3.1 – Les huit tâches étudiées et leur nombre de données disponibles

Tâche	Nombre de données
Couche de surface, enrobé	2596
Couche de base, enrobé	1819
Armature	936
Déblai de 1 ^{re} classe	675
Déblai de 2 ^e classe	1733
Sous-fondation de chaussée	1029
Fondation de chaussée	1191
Organisation de chantier	1667

Les ensembles utilisés sont directement extraits de la base de données BDP. Les données

ont été recueillies entre les années 2014 et 2021. Les données de 2014 à 2020 sont réservées pour tout le processus d'entraînement et de configuration des modèles. Celles de 2021 sont conservées pour évaluer un modèle de manière finale.

Chaque ligne de BDP contient l'information des soumissions émises par divers entrepreneurs à des appels d'offres. Elle contient donc des informations, pour chaque soumission, sur la quantité, la localisation, la date, le type de matériau, le type de travaux, etc. Une liste exhaustive des variables disponibles pour chaque ligne se trouve dans le tableau 3.2. Certaines des variables sont d'ailleurs un ajout à celles initialement fournies dans BDP. Le tableau 3.2 identifie ces dernières par la mention « (ajout) ». Les variables « Prix historique du bitume », « Salaires construction » et « IQ-30 » sont ajoutées dans l'espoir d'offrir un indicateur de santé économique ou d'inflation aux modèles. La variable « Distance usine » a quant à elle été ajoutée dans l'objectif de préciser la localisation du chantier par rapport aux usines d'enrobage. Le tableau 3.3 associe les variables aux différentes tâches. On peut y observer que toutes les tâches utilisent les informations spatio-temporelles, ainsi que les informations de style d'ouvrage.

TABLEAU 3.2 – Les différentes variables explicatives utilisées

Variables	Description
Quantité estimée	Quantité estimée (selon les devis)
Type travaux	Type de travaux (Plantation, engazonnement, écran antibruit, etc.)
Mois publication ao	Mois de publication de l'appel d'offre
Code dt	Identifiant de la direction générale en charge
Sud du fleuve	Si la région est au sud ou au nord du fleuve
Latitude et longitude	Latitude et longitude des travaux
Code nature	Nature des travaux (constructions, signalisation, terrassements, etc.)
Année	Année de publication de l'appel d'offre
Formule enrobé	Formule d'enrobé, d'asphalte (ESG-10, ESG-14, etc.)
Ponceau	Si la description des travaux indique « ponceau »
Réparation	Si la description des travaux indique « réparation », « réhabilitation », « réfection », « renforcement », « consolidation » ou « imperméabilisation »
Construction	Si la description des travaux indique « construction », « prolongement », « élargissement » ou « aménagement »
Galvanisé	Indique si l'armature est en acier galvanisé
Rivière	Si la description des travaux indique « rivière »
Revêtement	Si la description des travaux indique « revêtement »
Sécurité	Si la description des travaux indique « glis » ou « sécur »
Talus	Si la description des travaux indique « talus »
Matériau granulaire	Matériau granulaire utilisé
Prix du bitume (ajout)	Prix du bitume sur le marché au moment de l'appel d'offres
Salaires construction (ajout)	Salaire moyen des employés de la construction, fixé selon une convention
IQ-30 (ajout)	Indice économique de la CASIQ pour projet à grandes liquidités [39]
Distance usine (ajout)	Distance (en min.) entre la localisation d'un chantier et l'usine d'enrobage la plus proche

TABLEAU 3.3 – Variables utilisées par tâche

Variables	Tâches
Quantité estimée	Toutes, sauf organisation de chantier
Type travaux	Toutes
Mois publication ao	Toutes
Code dt	Toutes
Sud du fleuve	Toutes
Latitude et longitude	Toutes
Nature des travaux	Toutes
Année	Toutes
Formule enrobé	Couches de base/surface, enrobé
Ponceau	Toutes, sauf organisation de chantier
Réparation	Couches de base/surface, enrobé et Armature
Construction	Couches de base/surface, enrobé et Armature
Galvanisé	Armature
Rivière	Déblais de 1 ^{re} et de 2 ^e classe
Revêtement	Déblais de 1 ^{re} et de 2 ^e classe
Sécurité	Déblais de 1 ^{re} et de 2 ^e classe
Talus	Déblai de 2 ^e classe
Matériau granulaire	Sous-fondation/fondation de chaussée
Prix du bitume (ajout)	Couches de base/surface, enrobé
Salaires construction (ajout)	Toutes
IQ-30 (ajout)	Toutes
Distance usine (ajout)	Couches de base/surface, enrobé

Une autre incertitude est de savoir ce que l'on considérera comme l'estimation parfaite, la valeur cible à prédire. C'est un défi particulier au cas du MTQ. Selon les experts du MTQ, il est raisonnable de dire qu'il s'agit des meilleures soumissions, soient celles offrant le meilleur rapport qualité-prix. Il s'agit toutefois d'une hypothèse et il est impossible de savoir dans quelle proportion elle est exacte. Il serait donc ardu de l'appliquer dans d'autres contextes. Ainsi, l'objectif de nos modèles sera de prédire le prix moyen soumis par les trois meilleurs fournisseurs. Mathématiquement, la cible a été ajoutée aux données selon le calcul suivant :

$$cible = \frac{(rang_1 + rang_2 + rang_3)}{3} \quad (3.1)$$

où *cible* représente notre estimation cible et $rang_x$ représente le prix unitaire suggéré par le fournisseur positionné au rang x . Le rang 1 correspond au soumissionnaire offrant le meilleur rapport qualité-prix.

3.2 Variables externes ajoutées

Dans l'optique d'améliorer la précision des estimations, plusieurs variables ont été ajoutées à celles initialement contenues dans BDP. Ces différentes variables sont décrites dans les sous-sections suivantes.

3.2.1 Distance avec l'usine d'enrobage la plus proche

L'équipe de MTQ a pour intuition que la distance entre le chantier et l'usine d'enrobage la plus proche pouvait avoir un impact sur le prix de la soumission.

La position des usines (longitude et latitude) fut tirée de données internes du MTQ. Les usines démantelées ou en dormance ont été retirées du calcul. La position des chantiers provient des variables longitude et latitude des appels d'offres.

Pour calculer la distance avec une usine, l'API de Google [40] avait initialement été proposée. Cela dit, il est dorénavant payant. Une alternative gratuite a pu être utilisée [41]. L'API permet de calculer le temps d'un trajet entre deux coordonnées, selon les routes disponibles.

Ainsi, pour chaque appel d'offres, il faut calculer la distance séparant le lieu de son chantier avec chacune des usines disponibles (150 usines). Le temps de calcul total pour l'ajout de cette variable fut de 20 heures. Toutefois, une fois les résultats calculés, ils ont été enregistrés dans un fichier afin d'en faire une lecture instantanée pour les traitements subséquents.

Cette variable fut ajoutée aux tâches couches de base et de surface d'enrobé.

3.2.2 Prix historique du bitume

À la recherche d'une manière d'obtenir un indice sur les prix du marché du bitume, il a été proposé par l'équipe du MTQ d'utiliser les prix historiques.

Les données utilisées proviennent de l'équipe interne du MTQ. Elles contiennent les prix du bitume de 2015 à 2021. Puisque nous avons des soumissions de 2014, le prix historique de toutes les données de 2014 fut la première donnée de l'année 2015.

Le prix du bitume varie parfois aux jours, et d'autres aux semaines, il a été choisi de convertir les données pour avoir un seul prix par mois afin d'uniformiser l'approche. Ainsi, la donnée sur le prix du bitume la plus centrale (autour du 15^e jour de chaque mois) a été utilisée pour déterminer la donnée de ce mois en particulier. Finalement, pour sélectionner le mois et l'année d'un appel d'offres, on utilise sa date de publication.

Cette variable fut ajoutée aux tâches couches de base et de surface d'enrobé.

3.2.3 Salaires des employés de la construction

Une autre information qui pourrait être utile afin de donner au modèle une intuition quant à l'inflation est le salaire moyen des employés de la construction, qui est fixé selon une convention.

Les données disponibles à ce sujet étaient classées par type de métier. L'information utilisée pour ce projet est la moyenne de tous les types de métier.

Cette donnée a été ajoutée à toutes les tâches.

3.2.4 Indice économique (IQ-30)

Une dernière information permettant de donner une indication relative à la santé économique est l'indice économique. L'équipe du MTQ a proposé l'utilisation de l'indice IQ-30 [39], qui est spécifique au marché québécois. Il s'agit d'un indice adapté aux grands projets (secteurs à grandes liquidités). Encore une fois, c'est la date de publication d'appel d'offres qui est utilisée pour lier une soumission à une donnée IQ-30.

Cette donnée a été ajoutée à toutes les tâches.

3.3 Modèles d'apprentissage automatique

Lors de cette expérience, deux approches sont envisageables. On pourrait d'abord partir d'un modèle simple et tenter d'en améliorer la performance au maximum en effectuant un meilleur traitement préalable des données. En effet, il est possible qu'un modèle linéaire plus simple surpasse un modèle plus complexe en améliorant la qualité des données [33]. Si c'est le cas, il sera approprié d'utiliser le modèle en soi comme méthode d'explication. À l'inverse, on pourrait maximiser d'abord la performance en utilisant un modèle plus complexe, puis tenter diverses méthodes pour le rendre plus compréhensible. Elles seront toutes les deux tentées et comparées. Ainsi, un des objectifs de cette expérimentation sera de travailler les données en vue d'améliorer la performance des modèles. Ces techniques seront détaillées un peu plus loin. Cette section décrira les six modèles sélectionnés pour cette expérience, soient l'arbre de décision, la forêt aléatoire, les réseaux de neurones, XGBoost, CatBoost et *EnsembleVoters*. Cette variété de modèles permettra de mettre en évidence les différents équilibres possibles entre facilité et temps d'entraînement, performance et transparence.

3.3.1 Arbre de décision (AD) et forêt aléatoire (FA)

L'AD [10] est étudié en tant que modèle simple et explicable. Une prédiction est émise en posant une série de tests binaires à propos des valeurs que portent les variables explicatives. À titre d'exemple, la figure 3.1 reçoit d'abord les variables explicatives en entrée (quantité, lieu, type de travaux, année, etc.). Ensuite, un premier test est fait par rapport à la quantité. Si elle est plus petite que 50, on suit l'embranchement de gauche et inversement. Supposons que nous suivons l'embranchement de gauche, alors un test similaire sera effectué concernant le type de travaux. L'estimation finale est obtenue lorsque l'on rencontre une feuille de l'arbre.

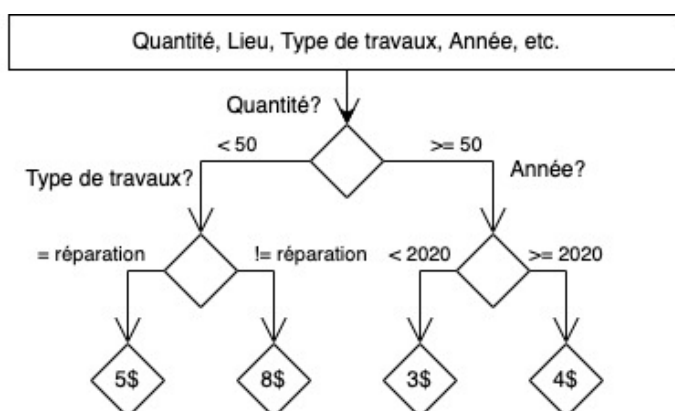


FIGURE 3.1 – Une représentation simplifiée d'un arbre de décision dans le contexte du problème d'estimation des coûts.

La FA [30] sera également étudiée. C'est un modèle avec lequel il est plus difficile d'expliquer une prédiction, quoiqu'on puisse aisément expliquer la logique derrière son algorithme. En effet, il s'agit d'une combinaison d'arbres de décision indépendamment entraînés sur des sous-ensembles aléatoires de l'ensemble de données initial. La sortie du modèle correspond à la prédiction moyenne des prix prédits par tous les arbres. Elle peut être exprimée par la formule suivante :

$$prix_i = \frac{1}{K} \sum_k^K f_k(x_i) \quad (3.2)$$

où K est le nombre d'arbres dans la forêt, f_k est la fonction associée à l'arbre k , x_i est le vecteur d'entrée contenant les variables explicatives pour la soumission i et $price_i$ est le coût unitaire estimé pour la soumission i .

3.3.2 XGBoost (XB) et CatBoost (CB)

Les algorithmes XG et CB sont tous les deux une implémentation de la technique de *gradient tree boosting* [30]. La prédiction du modèle s'appuie sur un groupe de petits arbres de décision

complémentaires. Un premier arbre émet une prédiction, puis les arbres suivants augmentent ou diminuent cette dernière de sorte à améliorer sa précision. Pour illustrer cette procédure, la figure 3.2 présente un exemple d'une prédiction du juste prix selon la logique du *gradient tree boosting*. Pour simplifier, l'illustration ne contient que deux arbres et n'implique que 3 des variables explicatives disponibles. Dans cet exemple, le premier arbre estime une valeur de -2\$ et le deuxième estime une valeur de 5\$. La sortie finale du modèle correspond à la somme des valeurs prédites par tous les arbres. En l'occurrence, il s'agit ici de 3\$.

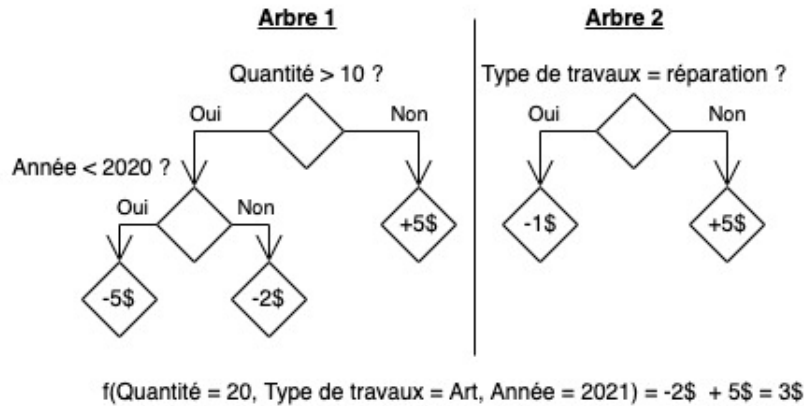


FIGURE 3.2 – Une représentation simplifiée d'un modèle basé sur le *gradient tree boosting*. Il reçoit en entrée (Quantité = 20, Type de travail = Art, Année = 2021) et il estime le coût unitaire à 3\$, en fonction de la somme des prédictions des deux arbres.

Ce processus peut être formalisé selon l'équation suivante :

$$prix_i = \sum_k^K f_k(x_i) \quad (3.3)$$

où $price_i$ est le coût unitaire estimé en fonction de l'entrée x_i , K est le nombre d'arbres dans l'ensemble et f_k est la prédiction de l'arbre k pour l'entrée x_i .

Il n'y a pas de différence évidente dans la logique d'inférence des deux modèles. Cependant, la méthode d'entraînement de CB ne permet pas l'obtention d'un arbre asymétrique, tel que l'arbre 1 dans la figure 3.2. Les différences sont principalement au niveau de la méthode d'apprentissage. CB a l'avantage de supporter les variables catégoriques sous leur forme textuelle. Cela dit, dans cette expérience, les variables catégoriques sont transformées en un format numérique, peu importe le modèle. L'avantage ne s'applique donc pas dans le cadre de cette expérimentation. Un autre avantage de CB est que l'entraînement des arbres peut se faire en parallèle [14], plutôt que de manière séquentielle tel que proposé dans l'algorithme XB. Grâce à cela, l'évaluation d'un arbre peut s'effectuer sur un ensemble différent que celui

avec lequel il a été entraîné, ce qui ajoute de la robustesse au modèle. D'un autre côté, le modèle XB est beaucoup plus rapide à entraîner. En résumé, il existe quelques différences mineures entre les deux implémentations du *gradient tree boosting*, mais ce sont tous les deux des modèles plutôt précis. Il s'agit cependant de modèles assez complexes, donc peu transparents.

3.3.3 Réseau de neurones

Le RN est sans doute le modèle étudié qui est le plus complexe. Il a à la fois un algorithme d'entraînement ardu à expliquer et il est d'autant plus difficile d'en interpréter les prédictions. Il s'agit toutefois d'un modèle offrant des prédictions des plus précises dans l'état de l'art. Ils seront donc étudiés dans cette expérience.

Un RN est composé de plusieurs couches de neurones séparant l'entrée et la sortie du modèle. Chaque neurone transmet ses perceptions aux neurones des couches suivantes, jusqu'au neurone de la dernière couche, qui portera la sortie finale résultante de la combinaison de ses prédécesseurs. Pour illustrer ce processus, la figure 3.3 présente un petit réseau de neurones tel que conçu dans le domaine de l'apprentissage automatique. On y observe qu'il est constitué de plusieurs couches de neurones. La couche d'entrée reçoit naturellement la valeur brute des différentes variables. La couche de sortie, composée d'un seul neurone contient l'estimation finale du réseau. On nomme toutes les couches intermédiaires les couches cachées. Pour effectuer une prédiction, la couche d'entrée reçoit d'abord la valeur des différentes variables de notre problème. Dans ce cas simplifié, on reçoit, respectivement dans les neurones 1, 2 et 3, l'année, le mois et la quantité. Chaque neurone j de la première couche applique un poids appris w_{ij} à cette valeur avant de l'envoyer au neurone i de la couche suivante. Ensuite, un neurone d'une couche cachée aura comme valeur la somme des valeurs portées par les neurones de la couche précédente multipliées par leur poids. Par exemple, le neurone 4 de la figure 3.3 aura une valeur équivalente à $année \times w_{14} + mois \times w_{24} + quantité \times w_{34}$. Avant de transmettre lui aussi sa valeur aux couches subséquentes, un biais appris β est ajouté à la somme, puis on applique une fonction non linéaire au résultat (activation). La logique se poursuit jusqu'au neurone de sortie qui calculera la prédiction finale de la même manière, mais en omettant l'activation.

Bien que la figure 3.3 présente un exemple simplifié, les modèles de cette étude contiennent entre une et deux couches cachées, puis quelques dizaines de neurones. La fonction d'activation choisie est la *Relu* :

$$ReLU = \max(0, x) \tag{3.4}$$

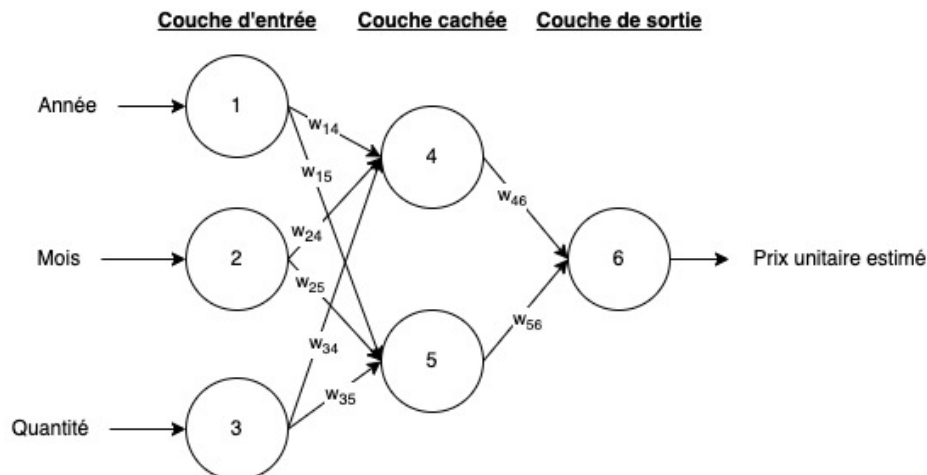


FIGURE 3.3 – Une représentation simplifiée d’un petit réseau de neurones recevant trois variables en entrée. Il contient une seule couche cachée et six neurones, en incluant les entrées et la sortie.

où x est la sortie non activée d’un neurone.

En somme, la procédure pour estimer le prix unitaire de la figure 3.3 peut se formuler mathématiquement de la manière suivante :

$$\begin{aligned}
 \text{neurone}_4 &= \text{ReLu}(\text{année}_i \times w_{14} + \text{mois}_i \times w_{24} + \text{quantité}_i \times w_{34} + b_4) \\
 \text{neurone}_5 &= \text{ReLu}(\text{année}_i \times w_{15} + \text{mois}_i \times w_{25} + \text{quantité}_i \times w_{35} + b_5) \\
 \text{prix}_i &= w_{46} \times \text{neurone}_4 + w_{56} \times \text{neurone}_5 + b_6
 \end{aligned} \tag{3.5}$$

où neurone_4 et neurone_5 correspondent aux sorties activées des neurones quatre et cinq, respectivement. prix_i est le coût unitaire prédit pour la soumission i . La même logique peut être généralisée à des réseaux composés d’un nombre différent de couches et de neurones.

L’avantage d’utiliser des RN est que chaque neurone applique une fonction non linéaire à sa sortie, ce qui le rend capable de représenter des relations complexes entre les variables. Cela dit, le processus d’entraînement est long et il est difficile de choisir la configuration optimale pour un certain problème. De plus, les prédictions sont difficilement explicables.

3.3.4 Modèle ensembliste hybride : EnsembleVoters (EV)

Une dernière famille de modèles qui a démontré d’excellents résultats en expériences similaires est celle des modèles hybrides [26–28]. Ces derniers se basent sur le concept d’ensemble [15]. C’est-à-dire qu’on cherche à combiner plusieurs modèles afin d’effectuer une prédiction finale.

Cette stratégie semble en effet avoir permis de limiter le surapprentissage, tout en améliorant la performance. Ainsi, le dernier modèle ajouté à cette expérience sera basé sur ces concepts. Nous l'appellerons *EnsembleVoters*.

Le modèle EV regroupe les cinq autres modèles étudiés. D'abord, chacun d'entre eux est indépendamment entraîné et optimisé. Ensuite, pour effectuer une prédiction, chaque modèle effectue d'abord la sienne de manière indépendante. C'est le prix médian parmi les cinq qui sera conservé comme prédiction finale. Ce processus est mathématiquement représenté par la formule suivante :

$$prix_i = \text{médiane}\left(\sum_k^5 f_k(x_i)\right) \quad (3.6)$$

où $preis_i$ est le prix unitaire estimé pour l'entrée x_i , f_k est la prédiction faite par le modèle k (XB, CB, AD, FA ou RN) pour l'entrée x_i .

3.4 Méthodes de validation et d'évaluation de la performance

Cette section présentera les méthodes d'évaluation de la performance.

3.4.1 Métrique

Tout au long de l'expérience, la métrique utilisée sera le pourcentage d'erreur absolue moyen (MAPE). Il s'agit d'une métrique communément utilisée dans des expériences similaires et elle est facile à interpréter [22, 25, 27, 28]. Il se calcule de la manière suivante :

$$MAPE = \frac{1}{n} \sum_i^n \frac{abs(\hat{y}_i - y_i)}{y_i} \quad (3.7)$$

où n est le nombre de données dans l'ensemble sur lequel on évalue la performance, y_i est l'estimation cible correspondant à l'appel d'offres i et \hat{y}_i est la prédiction du modèle pour l'appel d'offres i .

On utilisera cette même métrique pour évaluer l'estimation humaine, qui servira de comparatif pour bien interpréter la performance des modèles.

3.4.2 Méthode de validation croisée

Dans le cadre de ce mémoire, on nommera méthode de validation ce qui servira à sélectionner la meilleure configuration parmi d'autres lors du processus d'optimisation de la performance des modèles. Ainsi, on évite de biaiser nos résultats en utilisant l'ensemble de test pour effectuer cette sélection. La méthode qu'on utilisera se base sur la méthode utilisée par [33].

Il s'agit de la méthode classique de validation croisée à quatre plis. En général, les quatre sous-ensembles sont choisis aléatoirement. Or, puisque nos données sont dotées d'un élément temporel, il est plus adéquat de valider la performance de notre modèle sur des données plus récentes que celles sur lesquelles il fut entraîné. En effet, c'est ce qui ressemblera le plus à un cas pratique concret. La méthode choisie pour sélectionner les sous-ensembles tient compte de cette propriété. Comme qu'illustré à la figure 3.4, pour chaque configuration, une première validation sera faite avec les années 2014 à 2016 comme ensemble d'entraînement et l'année 2017 comme ensemble de validation. Les années 2018 à 2020 seront laissées de côté. Pour la deuxième, on laissera de côté les années 2019 et 2020. Les années 2014 à 2017 sont pour l'entraînement et l'année 2018 est pour valider. Ce même processus sera répété pour la troisième et la quatrième validation. Le choix de configuration sera fait selon le MAPE moyen des quatre validations. Cette méthode vise à permettre d'avoir un choix de configuration plus robuste, en diminuant le risque qu'une configuration soit meilleure que les autres par hasard. Pour avoir un comparatif, l'expérience est également tentée en utilisant une séparation aléatoire, afin d'observer l'apport de cette technique de séparation.

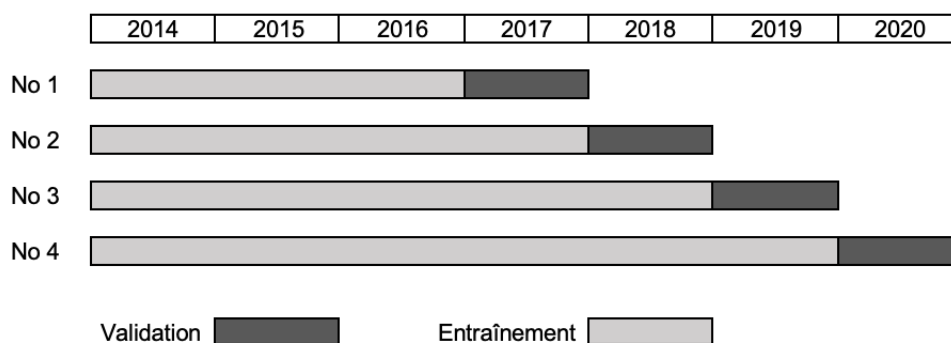


FIGURE 3.4 – Séparation de l'ensemble de données dans les quatre phases de validation.

3.4.3 Méthode d'évaluation finale

Dans la figure 3.4, on peut remarquer que l'année 2021 ne figure dans aucune des phases de validation. Cette dernière année sera conservée pour faire une dernière évaluation de la version finale de chaque modèle d'IA. Autrement dit, il s'agit de l'ensemble de test. Il est donc important que cette année ne soit impliquée d'aucune manière dans le processus de configuration du modèle. Ainsi, les données de 2014 à 2020 seront utilisées pour l'entraînement final et celles l'année 2021 servira à calculer la performance finale du modèle.

3.5 Pipeline d'optimisation de la performance

Un des objectifs de ce projet est de concevoir une manière standardisée d'évaluer et d'entraîner divers modèles d'IA. On propose donc un pipeline d'exécution, supporté par un outil en lignes de commandes permettant de le reproduire avec divers modèles et divers ensembles de données. Il comporte six grandes étapes, qui apparaissent à la figure 3.5.

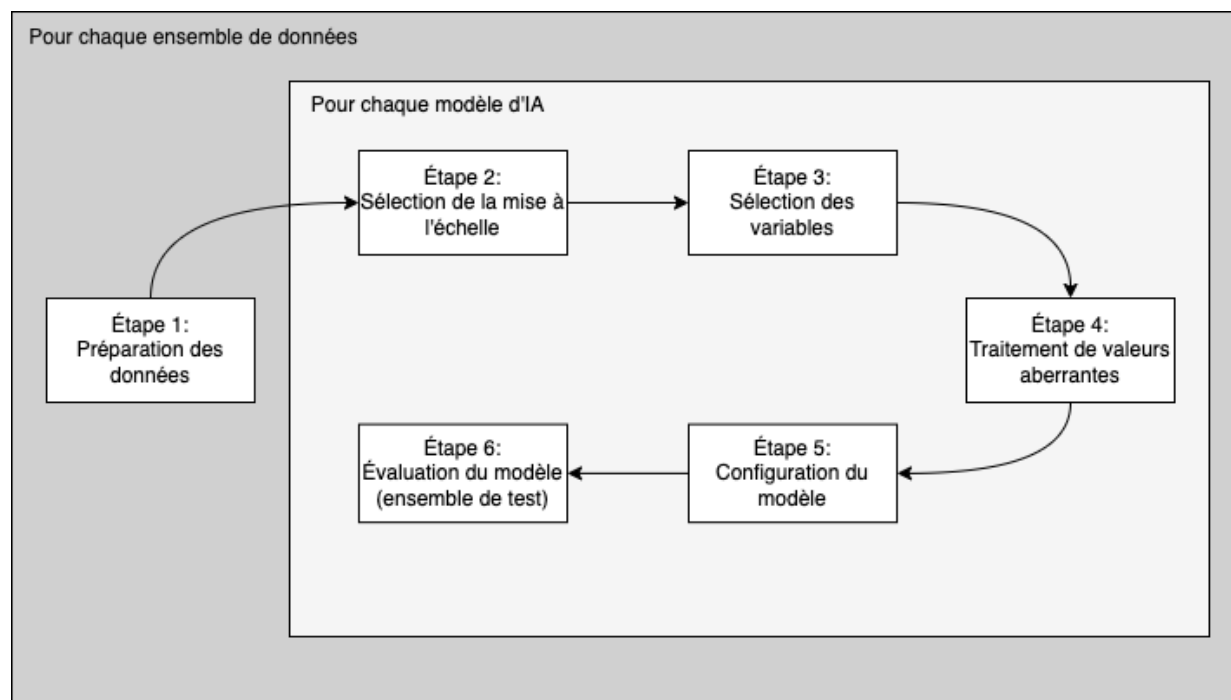


FIGURE 3.5 – Schéma des six étapes du pipeline d'optimisation pour un ensemble de données.

Les différentes étapes seront détaillées dans les sous-sections du même nom, à l'exception de celle d'évaluation finale (étape 6), qui a été expliquée à la section 3.4.3.

3.5.1 Préparation des données (étape 1)

Certaines manipulations sont nécessaires pour que les données brutes soient dans un format interprétable par l'un ou l'autre des modèles. En l'occurrence, les lignes de l'ensemble de données qui contenaient des valeurs manquantes ont été supprimées. D'autres techniques proposent d'imputer les valeurs manquantes par une valeur moyenne, mais il est préférable de les supprimer si peu de lignes sont touchées. Dans notre cas, entre 3% et 5% des données contenaient des valeurs nulles, selon les tâches. Notez que pour de nouvelles prédictions, il faudrait utiliser une forme d'imputation des données en cas de valeur manquante, bien que ce soit à éviter le plus possible.

Aussi, les variables discrètes sont transformées en utilisant l’encodage *one hot*. Il s’agit d’une technique d’encodage permettant de rendre toutes les variables numériques, donc interprétables par nos modèles. Soit un exemple simple où nous avons une variable « Couleur », pouvant porter les valeurs « Rose », « Bleu » ou « Jaune ». Alors, on transformerait cette variable en trois nouvelles variables nommées « Couleur_Rose », « Couleur_Bleu » et « Couleur_Jaune », chacune d’entre elles pouvant avoir la valeur 0 (faux) ou 1 (vrai). Ainsi, pour indiquer une donnée rose, on aurait Couleur_Rose=1, Couleur_Bleu=0 et Couleur_Jaune=0. Ces deux manipulations assurent la compilation des différents modèles.

3.5.2 Sélection de la mise à l’échelle (étape 2)

De nombreux algorithmes d’IA sont sensibles à la mise à l’échelle préalable des données numériques. Il s’agit d’altérer les données de sorte à réduire leur variance ou leur valeur absolue. Pour ce faire, on peut borner les valeurs entre 0 et 1 (normalisation) ou encore modifier les données pour que sa distribution porte une moyenne égale à zéro et un écart-type égal à 1 (standardisation). Cette procédure peut être utile pour certains modèles lorsque l’ensemble de données contient des valeurs ayant des ordres de grandeur différents. C’est notamment le cas de tous ceux qui se basent sur un apprentissage par gradient. Il s’agit d’un système de correction progressive de l’erreur du modèle. Parmi les modèles utilisant ce système, on retrouve les RN. Par extension, le modèle hybride sera également touché. À l’opposé, certains modèles sont connus pour leur invariance face à la mise à l’échelle. C’est le cas des arbres de régressions, donc des forêts aléatoires. Pour les modèles influencés par cette mise à l’échelle, il n’est pas garanti que la mise à l’échelle entre 0 et 1 soit la plus appropriée. En effet, selon l’ensemble de données, il est probable qu’une standardisation soit préférée.

Pour cette étape, chaque modèle passerait par le processus de validation complet (4 entraînements par validation) trois fois. Une première fois sans aucune mise à l’échelle, une deuxième fois avec une mise à l’échelle standard et une troisième fois avec une normalisation. Ce processus totalise 12 entraînements. Toutefois, un seul entraînement d’un RN peut prendre jusqu’à 30 minutes dans le cadre de cette expérience. Il a donc été préféré d’utiliser un modèle alternatif aux RN pour sélectionner la mise à l’échelle. XB fut choisi pour ce faire. Le modèle EV contenant également un RN, la même stratégie a été employée.

La meilleure mise à l’échelle sélectionnée sera conservée pour la suite du pipeline. En cas d’égalité entre les trois méthodes, aucune mise à l’échelle ne sera faite.

3.5.3 Sélection des variables (étape 3)

La présence de certaines variables peut grandement influencer la performance de certains modèles. Les modèles plus complexes tels les RN ou XB y sont généralement moins sensibles, mais les arbres de régression peuvent grandement bénéficier d'une présélection de variables rigoureuse. Afin d'uniformiser le processus, et puisque des modèles complexes pourraient tout de même en bénéficier légèrement, tous les modèles passeront par cette étape.

Deux phénomènes peuvent contribuer au fait qu'une variable nuise à la précision d'un modèle. Il pourrait d'abord y avoir présence de colinéarité entre des variables. À l'inverse, il se pourrait qu'une variable n'ait tout simplement pas de relation claire avec la cible. Dans les deux cas, filtrer la variable problématique peut s'avérer bénéfique.

Ainsi, on utilisera la méthode *Recursive Feature Elimination* (RFE) [42]. Il s'agit d'un processus itératif permettant de réduire le nombre de variables à une valeur cible n . Pour ce faire, à chaque itération, on entraîne le modèle et on calcule l'importance des variables. On retire ensuite la variable la moins significative avant de répéter une itération. Le processus se poursuit jusqu'à ce que le problème contienne n variables. Soit N , le nombre de variables initialement dans le problème, l'algorithme impliquera $N - n$ entraînements.

Un défi de cet algorithme est la sélection de la valeur cible n . Pour ce faire, la méthode de validation croisée définie dans la sous-section 3.4.2 sera utilisée. L'algorithme RFE sera exécuté de manière indépendante avec toutes les valeurs comprises entre 5 et N , en supposant qu'il est peu probable qu'un ensemble de variables de taille plus petite que 5 soit approprié. Cela permet d'éviter quelques entraînements. Pour chaque valeur n testée, on appliquera la technique de validation croisée afin d'en évaluer la performance. On conservera le nombre de variables ayant permis d'obtenir le plus petit MAPE moyen. Cette technique implique que l'algorithme RFE sera exécuté quatre fois pour chaque valeur testée.

Selon les tâches, chaque ensemble contient environ 60 variables explicatives à la suite de la conversion des variables catégoriques en format numérique. Dans ce cas, 55 valeurs différentes seraient testées. RFE serait donc exécuté 220 fois ($55 * 4$) pour chaque modèle et chaque tâche,

ce qui totalise 6160 entraînements. Ce résultat peut être obtenu selon le calcul suivant :

$$\begin{aligned}
 \sum_{n=5}^N 4 * (N - n) &= 4 * (55 + 54 + 53 + \dots + 3 + 2 + 1) \\
 &= 4 * \sum_{n=1}^{55} n \\
 &= 4 * \frac{55 * (55 + 1)}{2} \\
 &= 6160
 \end{aligned} \tag{3.8}$$

Étant donné ce grand nombre d'entraînements, c'est le modèle XB qui sera utilisé pour la sélection des variables des RN et du modèle EV pour éviter une explosion des coûts de calculs.

3.5.4 Traitement des valeurs aberrantes (étape 4)

Une autre manipulation qui peut s'avérer efficace pour améliorer la performance est le traitement des valeurs aberrantes. En effet, cela peut permettre de contrôler l'uniformité des données. Cela dit, le défi est que de savoir si une valeur extrême l'est par erreur ou si elle permet réellement d'illustrer une relation qui survient plus rarement. Il peut être ardu d'effectuer cette tâche manuellement. Pour cette raison, on fait appel à des algorithmes permettant de cibler les valeurs aberrantes de manière automatique. Pour cette expérience, on testera et comparera l'effet de quatre techniques, présentées brièvement ci-dessous. Les différentes implémentations ont été fournies par la librairie python *scikit-learn* [43].

IsolationForest (IF)

Ici, on vise à isoler les données les unes des autres en enlevant l'effet de certaines variables explicatives. L'idée générale est ensuite d'évaluer quelles données ont été les plus faciles à isoler dans l'ensemble. On considère ainsi ces dernières comme des anomalies [44].

LocalOutlierFactor (LOF)

Cette méthode attribue un certain voisinage à chaque donnée et tente d'évaluer statistiquement le degré auquel cette donnée est susceptible de provenir de la même distribution que ses voisins. On choisirait ensuite une valeur frontière à laquelle on considère la donnée comme étant aberrante [45].

OneClassSVM (OCSVM)

Sans entrer dans les détails mathématiques de cette dernière méthode, l'idée générale est de transformer notre problème en un problème de classification de données dans lequel il n'y aurait qu'une seule classe. On tentera de réduire la dimension des données pour unifier au maximum toutes les données. Une fois ce rapprochement maximal, il est attendu de remarquer que les données les plus loin du lot sont des données aberrantes [46].

Le processus comprenant encore une fois 16 entraînements, le modèle XB sera utilisé pour choisir la meilleure technique pour les RN et le modèle EV.

La méthode permettant de rapporter le meilleur score de validation sera conservée.

3.5.5 Configuration du modèle (étape 5)

À l'étape 5, les variables explicatives sont finales et les données ne changeront plus. Cette dernière étape d'optimisation concerne le modèle en soi et non les données. Il est important que cette étape soit effectuée en dernier puisque le choix des hyperparamètres peut dépendre des données. Selon le modèle, une variété raisonnable de combinaisons d'hyperparamètres est proposée. Ces combinaisons sont issues du fait que pour chaque paramètre, on offre une liste contenant diverses valeurs à tester. Ainsi, chaque combinaison possible de ces paramètres formera une configuration. Ensuite, chacune des configurations passera au travers du processus de validation, puis on conservera la meilleure pour l'évaluation finale.

La sélection d'hyperparamètres fut toutefois faite de manière différente pour les réseaux de neurones. La quantité de combinaisons possibles étant pratiquement innombrable, ainsi que le temps d'entraînement plutôt long, il n'était pas possible de faire une recherche de paramètres aussi exhaustive que celle faite avec les autres modèles. La sélection a donc été faite au préalable de manière indépendante pour chaque tâche. L'optimisation fut faite à tâtons en augmentant graduellement la complexité des réseaux jusqu'à l'observation d'un surapprentissage. Une fois ce seuil atteint, diverses techniques de régulation ont été employées. En l'occurrence, des couches *Dropout* de proportions variantes en 30% et 60% ont été utilisées. Davantage de précisions quant à la configuration des réseaux de neurones se trouvent en annexe A.

3.6 SHAP : Explication des prédictions

La précision des prédictions n'est pas la seule caractéristique recherchée dans le cadre de cette expérience. On aspire également à ce que les prédictions puissent être justifiées. Alors

que peu de modèles parmi ceux étudiés sont explicables, il existe des algorithmes permettant d'établir la relation entre les différentes variables et la sortie du modèle, peu importe la nature de ce dernier. On dit de ces algorithmes qu'ils sont agnostiques au modèle. C'est d'ailleurs le cas des méthodes de la famille SHAP [16]. L'algorithme SHAP repose sur le calcul de ce qui est appelé les valeurs de Shapley. La présente section définira ces valeurs et leur implication au sein des méthodes SHAP.

3.6.1 Origine des valeurs de Shapley

Ces valeurs ont d'abord été introduites par Lloyd Shapley dans le contexte de la théorie des jeux de coalition [37]. Le scénario associé aux jeux de coalition suppose un environnement dans lequel plusieurs joueurs collaborent dans l'optique d'obtenir un certain bénéfice. Une fois ce bénéfice obtenu, on cherche à rémunérer chaque joueur de manière proportionnelle à leur contribution, en admettant que les joueurs n'aient pas tous contribué de manière égale. Pour répondre à cette question, on calcule la valeur de Shapley de chaque joueur. Une valeur de Shapley représente la contribution marginale du joueur. Ces valeurs permettent donc de diviser le bénéfice de manière juste. Un parallèle peut être fait avec un problème d'apprentissage automatique. En l'occurrence, on chercherait à connaître la contribution de chaque variable pour une estimation. C'est de ce parallèle que proviennent les méthodes d'explicabilité basées sur les valeurs de Shapley. La valeur associée à chaque variable explicative repose sur le calcul mathématique suivant :

$$\Phi_i(x) = \sum_{S \subset F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|} \left[\underbrace{f_{S \cup \{i\}}(X_{S \cup \{i\}})}_{\text{estimation avec la variable } i} - \underbrace{f_S(x_S)}_{\text{estimation sans la variable } i} \right] \quad (3.9)$$

où :

- x : vecteur des variables données en entrée
- $\Phi_i(x)$: valeur de Shapley pour la variable explicative i et l'entrée x
- F : ensemble de toutes les variables
- f_S : modèle d'IA entraîné avec les variables du sous-ensemble S (S exclut la variable i)
- $f_{S \cup \{i\}}$: modèle d'IA entraîné avec les variables du sous-ensemble S , en incluant cette fois-ci i
- x_S : donnée d'entrée x incluant seulement les variables contenues dans S
- $x_{S \cup \{i\}}$: donnée d'entrée x incluant les variables contenues dans S et la variable i

Globalement, cette formule implique de calculer la différence entre l'estimation faite par le modèle entraîné sans et avec la variable i (contribution marginale). Cette différence sera

toutefois calculée selon tous sous-ensembles de variables concevables. La valeur de Shapley résultante est la moyenne pondérée de toutes les contributions marginales calculées. On peut mathématiquement prouver certaines caractéristiques désirables de ces valeurs [16] :

- La somme des valeurs de Shapley de toutes les variables est égale à l'estimation faite par le modèle entraîné avec l'ensemble complet de variables (modèle original)
- Une variable qui ne modifie pas l'estimation aura une valeur Shapley nulle
- Si deux variables ont la même valeur dans l'entrée x , les valeurs Shapley seront identiques si leur contribution est égale dans tous les sous-ensemble concevables

Ces caractéristiques assurent une certaine cohérence à l'importance attribuée à chaque variable selon cette méthode.

3.6.2 Expliquer une prédiction selon la méthode SHAP

Les valeurs de Shapley sont bien intéressantes étant donné la qualité de cohérence qu'elles portent. Cela dit, le calcul 3.9 représente un effort de calcul considérable. En effet, il existe $2^{|F|}$ sous-ensembles de variables, ce qui implique autant d'entraînements du modèle. Les algorithmes issus de la famille SHAP reposent sur des approximations des valeurs de Shapley. L'importance attribuée aux variables selon ces approximations a toutefois été considérée assez proche que celle qu'un expert aurait pu attribuer [16]. De plus, vous aurez vu que l'équation 3.9 ne comporte pas d'information relative au modèle d'IA spécifique, nous pourrions donc l'utiliser peu importe le modèle choisi. Ainsi, cette méthode est appropriée pour le présent projet.

Une fois les valeurs de Shapley calculées pour toutes les variables impliquées dans une estimation, il sera possible de l'exprimer selon la formule suivante :

$$\begin{aligned}
 f(x) &= E[f(x)] + \sum_i^F \Phi_i(x) \times x_i \\
 &= E[f(x)] + \Phi_{année}(x) \times année + \Phi_{quantité}(x) \times quantité + \dots + \Phi_{mois}(x) \times mois
 \end{aligned}
 \tag{3.10}$$

où :

- x : vecteur des variables données en entrée, correspondant à l'appel d'offres
- f : modèle d'IA entraîné
- $f(x)$: estimation du prix prédit selon l'appel d'offres x
- $E[f(x)]$: espérance de $f(x)$
- F : ensemble de toutes les variables
- $\Phi_i(x)$: valeur de Shapley pour la variable explicative i et l'entrée x
- x_i : valeur de la variable i dans l'appel d'offres x

Cette équation permet de calculer d'établir la contribution marginale de chaque variable, donc de justifier l'estimation du prix en fonction des différentes variables explicatives.

3.7 Implémentation : outils en lignes de commandes simples et réutilisables

Le code conçu pour lancer les expériences a été écrit de sorte qu'il soit réutilisable pour divers problèmes d'apprentissage automatique. En effet, chaque tâche d'un projet de construction représente un problème différent, on peut donc lancer l'expérience en boucle pour chaque tâche étudiée. Cela dit, l'outil créé pourrait aussi s'adapter à un problème de régression totalement différent. Il s'agit d'un outil comprenant deux commandes à lancer dans un terminal. La première commande recherche le modèle le plus performant parmi ceux étudiés et la deuxième permet de générer deux explications SHAP des modèles générés par la première commande. Ainsi, la présente section décrit cet outil et son fonctionnement.

3.7.1 Description de l'outil

L'outil fut principalement développé dans le langage python (version 3.9). Il contient plus de 2000 lignes de code réparties dans 26 fichiers. Il comporte cinq modules :

- **models** : contient le code relatif à l'implémentation des différents modèles d'IA
- **preprocessing** : contient la logique derrière les étapes de préparation des données, principalement lié à l'étape 1 du pipeline (voir section 3.5.1)
- **selectors** : concerne les étapes 2 à 5 du pipeline, soient le code concernant la mise à l'échelle, la sélection de variables, la détection des valeurs aberrantes et la recherche d'hyperparamètres
- **pipeline** : module liant les modules *models*, *preprocessing* et *selectors*, afin de lancer les boucles principales du pipeline
- **explainability** : module permettant de générer les explications SHAP

En somme, ces modules permettent le lancement de deux outils en lignes de commandes *tunai* et *shap-explain*, qui permettent respectivement de lancer le pipeline et de générer les explications SHAP.

3.7.2 Bibliothèques utilisées

Les principales bibliothèques utilisées sont les suivantes :

- **scikit-learn** [43] : modèles FA et AD, ainsi que des techniques de mise à l'échelle, de sélection de variables et de détection de valeurs aberrantes
- **catboost** [47] : modèle CB

- **xgboost** [48] : modèle XB
- **pytorch** [49] : modèle RN
- **shap** [50] : explications SHAP
- **Click** [51] : mise en place des outils en lignes de commandes

3.7.3 Utilisation du pipeline

La commande *tunai* offre un pipeline standardisé configurable offrant des résultats reproductibles pour un même ensemble de données et une même configuration. Il supporte six modèles d'IA, quatre algorithmes de détection de valeurs aberrantes, deux méthodes de mise à l'échelle et deux méthodes de génération des ensembles de validation croisée (aléatoire ou selon le facteur temporel). Il est possible d'exclure des modèles de l'expérience. On peut aussi l'utiliser avec n'importe quel ensemble de données, en spécifiant certaines caractéristiques de ce dernier dans un fichier de configuration. À la suite de son exécution, divers rapports sont donnés en sortie énumérant les performances des différents modèles à divers moments durant le processus d'exécution du pipeline. La sortie comporte également la meilleure version de chaque modèle d'IA dans un format compressé.

Pour lancer le pipeline, il suffit de lancer la commande suivante :

```
tunai chemin/vers/configuration_experience.yml -output_path chemin/pour/la/sortie
```

La commande comporte un argument de position obligatoire, soit le fichier de configuration *yml*, et un argument optionnel *output_path*. Ces derniers sont décrits ci-dessous.

Fichier de configuration. Le fichier de configuration doit être sous format *yml*. Il spécifie des informations relatives aux ensembles de données. En l'occurrence, on doit recevoir le chemin vers les ensembles des données, la colonne à prédire, les colonnes catégoriques et les colonnes numériques. Il permet aussi de personnaliser le pipeline au besoin. On peut modifier la méthode de validation croisée utilisée et spécifier les modèles d'IA utilisés dans l'expérience. D'ailleurs, bien qu'une liste d'hyperparamètres soit déjà fournie par défaut, on pourrait l'écraser pour l'un ou l'autre des modèles. L'exemple de fichier de configuration suivant décrit son contenu plus en détail à l'aide de commentaires.

```
1 # configuration_experience.yml
2
3 # nom de l'expérience
```

```
4 name: Expérience avec tâche démo
5
6 # chemin vers les ensembles de test et d'entraînement sous format .csv
7 input:
8     train: chemin/vers/train.csv
9     test: chemin/vers/test.csv
10
11 # nom et type de la cible à prédire
12 # correspond à une colonne dans les ensembles .csv
13 target:
14     name: prix_unite_moyen_3_soums
15     type: float
16
17 # noms des variables catégoriques
18 categorical_features:
19     - type_travaux
20     - mois_publication_ao
21
22 # noms des variables numériques qui seront mises à l'échelle
23 features_to_scale:
24     - Quantité estimée
25     - latitude
26     - longitude
27     - annee
28
29 # méthode temporelle ou aléatoire pour la validation croisée
30 # soit time_series ou random
31 validation_split_strategy: time_series
32
33 # utilisée lorsque validation_split_strategy=time_series
34 # c'est la colonne utilisée pour séparer ensembles de validation
35 date_feature: annee
36
37 # nombre d'ensembles de validation différents
38 nb_folds: 4
39
```

```

40 # liste des modèles à tester, tous ceux disponibles sont dans cet exemple
41 models:
42   -
43     name: DeterministicDecisionTree
44   -
45     name: RandomForestRegressor
46   -
47     name: XGBRegressor
48   -
49     name: CatBoostRegressor
50   -
51     name: NeuralNetwork
52     # pour préciser une liste d'hyperparamètres à tester
53     # sinon c'est la liste par défaut
54     configuration: NeuralNetwork.yml
55     # modèle alternatif utilisé pour
56     # sélectionner le prétraitement de données
57     # Utile pour accélérer le processus
58     alternate_model: XGBRegressor
59   -
60     name: EnsembleVoters
61     alternate_model: XGBRegressor

```

Fichiers de sortie. L'outil permet d'imprimer un rapport de performance pour chaque modèle, ainsi qu'un rapport comparatif des meilleures versions de chaque modèle. Ces rapports se retrouvent dans un dossier *report* par défaut, mais on peut spécifier une sortie alternative via l'argument *output_path*.

Un résumé du contenu du dossier de sortie se trouve à la figure 3.6. Le premier sous-dossier de gauche correspond au résumé de performance pour un modèle. Ainsi, il contient un fichier *results.csv*, qui énumère les scores de validation obtenus à différents moments du pipeline. La première ligne indique celui avec la meilleure mise à l'échelle, la deuxième celui avec le meilleur ensemble de variables, la troisième celui suite au traitement des valeurs aberrantes et la quatrième celui suite à la recherche des hyperparamètres. Chaque ligne indique également le nom du traitement sélectionné (ou les hyperparamètres sélectionnés). Le deuxième sous-dossier en débutant à gauche contient les ensembles de données suite à l'étape de préparation

des données (étape 1). Le troisième sous-dossier contient une copie des ensembles de données traités avec les meilleures configurations du pipeline et le meilleur modèle (parmi tous ceux étudiés) entraîné sur ces données dans un format compressé. Le fichier *models.csv*, se trouvant à la racine du dossier de sortie contient une ligne pour chaque modèle étudié. Chaque ligne indique le score obtenu par la meilleure version dudit modèle.

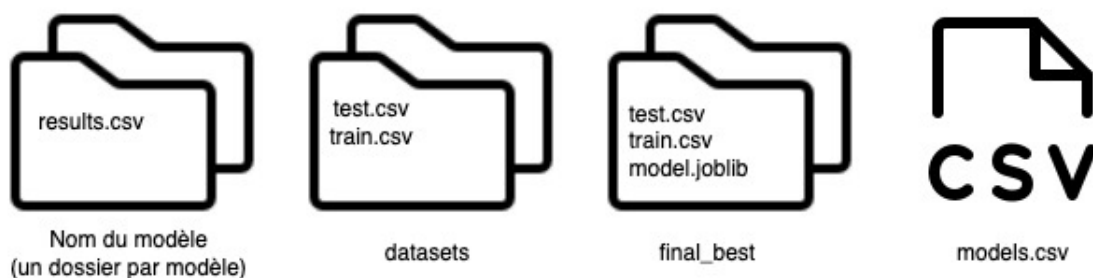


FIGURE 3.6 – Contenu du dossier de sortie généré à la suite de l’exécution du pipeline. Le nom des dossiers ou des fichiers se trouve au bas de la figure.

3.7.4 Génération d’explications

La commande *shap-explain* permet de générer deux explications visuelles basées sur les méthodes SHAP. La première est une explication locale, concernant l’impact de chaque variable pour une seule prédiction. La deuxième résume l’impact moyen de chaque variable sur l’ensemble des prédictions. Cette dernière permet donc une analyse des facteurs les plus influents quant aux estimations faites par un modèle. Les figures sont générées par la librairie python *shap* [50]. Pour générer les explications, on peut lancer la commande suivante :

```
shap-explain -model_path model.joblib -data_path train.csv -output_path explications
```

- **model_path** : lien vers le modèle entraîné pour lequel on veut générer des explications, compatible avec les modèles compressés *joblib* générés par *tunai* (voir section 3.7.3)
- **data_path** : chemin vers l’ensemble de données d’entraînement sous format *csv*
- **output_path** (optionnel) : chemin pour sauvegarder les explications

Des exemples visuels des figures données en sortie sont présentés en annexes et à la section 5.

CHAPITRE 4 ANALYSE DES RÉSULTATS

Le pipeline a été lancé pour chacune des huit tâches, permettant ainsi de générer 48 instances de modèles optimisés (six instances par tâche). L'exécution via *tunai* permet de générer les résultats présentés dans le présent chapitre. Ils seront présentés selon les différents objectifs de ce mémoire. L'analyse débutera avec un comparatif de la performance des différents modèles. On observa ensuite l'apport de l'algorithme RFE et de l'ajout de variables. Une analyse comparative des algorithmes de détection de valeurs aberrantes suivra. On vérifiera également l'apport de la technique de séparation temporelle concernant la validation croisée. L'analyse terminera en identifiant les meilleurs modèles pour chaque tâche, pour ultimement comparer leurs performances avec celles de l'estimation humaine.

4.1 Comparaison des modèles d'IA étudiés

Résultant du lancement du pipeline avec six modèles d'IA et huit tâches, 48 instances de modèles entraînés ont été produites. Pour chacune de ces instances, le MAPE (%) fut calculé. Les résultats sont présentés au tableau 4.1. Chaque cellule de ce tableau correspond à la meilleure version du modèle indiqué par la colonne. Ainsi, il s'agit du résultat obtenu avec les meilleures configurations sélectionnées par le pipeline. Suite à ces sélections, l'entraînement du modèle fut fait sur les années 2014 à 2020 et le MAPE indiqué a été calculé sur l'ensemble de 2021. Les rangées au bas du tableau 4.1 contiennent les performances moyennes et médianes obtenues par un modèle. Autrement dit, il s'agit du MAPE moyen (médián) par colonne. Pour chaque tâche (chaque ligne), le meilleur résultat apparaît en caractère gras. Cela indique le modèle ayant commis la plus petite erreur pour une certaine tâche.

Pour trois tâches parmi les huit, c'est le modèle EV qui a su se tailler une place au premier rang. Il s'agit du modèle ayant surpassé les autres le plus de fois, suivi du modèle XB. Ce résultat concorde avec ceux des expériences similaires ayant comparé des modèles hybrides avec d'autres modèles individuels [26–28]. Il diffère toutefois avec ceux de [29, 31], qui identifiaient plutôt le modèle XB comme étant le plus performant. Or, aucune de ces expériences présentant des modèles hybrides n'étudiaient le modèle XB. À l'inverse, les expériences étudiant XB n'étudiaient pas les modèles hybrides tels que conçus dans la présente étude. Les conclusions découlant de nos résultats peuvent donc être un peu plus larges. Cela dit, bien qu'en nombre de victoires le modèle EV surpasse XB, le modèle XB surpasse le modèle EV en termes de performance moyenne. On remarque toutefois que la médiane du modèle EV est meilleure. Cela indique que la performance du modèle XB est plus stable. En effet, le modèle

EV dépend des autres modèles, qu'ils aient été performants ou non. Ainsi, il peut autant bénéficier des avantages des meilleurs modèles, que de leurs pires désavantages. Il serait donc sans doute moins robuste aux données extrêmes. Outre les cinq tâches pour lesquelles ces deux modèles ont été au premier rang, les résultats sont différents pour les trois autres. Chaque modèle restant, soient FA, CB et NN ont surpassé les autres modèles à leur tour, à l'exception du modèle AD. Le modèle AD est toutefois le modèle étudié ayant le meilleur niveau d'explicabilité. L'expérience démontre que malgré un prétraitement des données rigoureux, ce modèle peine à rivaliser avec les modèles plus complexes. Ce résultat diffère avec celui obtenu dans [33]. En effet, cet article utilisant un traitement de données rigoureux montrait que leur modèle plus simple surpassait les plus complexes dans ces circonstances. Il faut néanmoins souligner que leur modèle plus simple n'était pas un arbre décision. Il est également possible que les relations entre les variables de notre problème soient plus complexes que dans cette dernière expérience. On ne peut donc toujours pas écarter qu'un modèle plus simple pourrait être meilleur qu'un modèle plus complexe, mais ce n'est pas le cas pour nos ensembles de données. Contrairement aux résultats de [32], CB n'a pas su surpasser XB pour presque tous les ensembles de données étudiés. Cela dit, nous n'avons pas profité du traitement natif des variables catégoriques, ce qui aurait pu avoir un impact positif sur la performance des modèles.

TABLEAU 4.1 – Comparaison du MAPE (%) sur l'ensemble de test (année 2021) des modèles d'IA par tâche

Tâche	AD	FA	XB	RN	CB	EV
Couche de surface, enrobé	22,37	22,64	22,60	30,91	29,69	20, 63
Couche de base, enrobé	25,68	19,62	20,87	40,59	19, 57	20,83
Armature	32,10	26, 69	30,76	40,98	27,81	31,22
Déblai de 1 ^{re} classe	77,10	60,53	52, 20	339,58	72,64	55,13
Déblai de 2 ^e classe	44,63	39,38	37, 04	39,64	37,33	38,89
Sous-fondation de chaussée	26,00	21,02	20,37	55,07	20,21	19, 96
Fondation de chaussée	20,36	17,76	17,15	47,89	17,14	16, 96
Organisation de chantier	103,23	113,82	83,34	82, 15	132,06	109,07
Moyenne	43,93	40,18	35, 54	84,60	44,55	39,09
Médiane	26,00	22,64	22,60	40,98	27,81	20, 83

Un autre critère important en apprentissage automatique appliqué est le temps et l'énergie requis pour optimiser et entraîner une instance d'un modèle. Ainsi, le tableau 4.2 présente le temps qui fut nécessaire pour l'entraînement des modèles sur les années 2014 à 2020. On observe que les modèles DT, RF et XB offrent un entraînement quasi instantané pour chaque tâche. Le modèle CB est également rapide, mais il peut arriver qu'il prenne jusqu'à 230 se-

condes pour l'une des tâches. Ce modèle utilise un mécanisme d'arrêt précoce (*early stopping*) lui permettant de raccourcir l'entraînement dans de nombreux cas, mais ce mécanisme a des limites. Les modèles RN et EV sont ceux nécessitant le plus de temps d'entraînement, prenant jusqu'à 30 minutes. Le modèle EV contient toutefois un RN, ce qui allonge considérablement son temps d'entraînement. Le temps d'entraînement est d'ailleurs directement proportionnel à la difficulté d'optimisation de la performance. Ainsi, plus l'entraînement d'un modèle est long, plus il sera difficile de cibler la meilleure configuration pour ce dernier, donc de profiter de son plein potentiel. Cela pourrait expliquer pourquoi nos résultats diffèrent avec ceux de la littérature concernant le modèle RN, souvent acclamé pour sa performance [22, 23, 25]. Il faut aussi souligner que ces expériences n'étudiaient pas de modèles hybrides ni les modèles XB ou CB.

TABLEAU 4.2 – Temps (sec.) requis pour l'entraînement des modèles sur les années 2014-2020 (Matériel : processeur 2,6 GHz Intel Core i5, 8GB de mémoire et aucun GPU)

Tâche	AD	FA	XB	RN	CB	EV
Couche de surface, enrobé	< 1	1	< 1	2	230	513
Couche de base, enrobé	< 1	1	1	549	2	394
Armature	< 1	< 1	< 1	1840	2	570
Déblai de 1 ^{re} classe	< 1	< 1	< 1	87	4	507
Déblai de 2 ^e classe	< 1	< 1	< 1	370	10	430
Sous-fondation de chaussée	< 1	< 1	< 1	762	1	297
Fondation de chaussée	< 1	< 1	< 1	563	2	834
Organisation de chantier	< 1	< 1	< 1	290	1	220

En somme, le modèle hybride surpasse les autres en utilisant le critère de médiane et celui du nombre absolu de tâches pour lesquelles il arrive au premier rang. Le modèle XB arrive au premier rang selon le critère de performance moyenne. Il semble également plus stable et est considérablement plus rapide à entraîner, donc plus facile à optimiser. Pour ses raisons, le modèle XB est le plus adapté à notre problème selon les résultats actuels. Une piste d'amélioration pouvant mitiger ce résultat pourrait être d'enlever le modèle RN du modèle EV. On pourrait ainsi espérer une accélération considérable du temps d'entraînement et une performance plus stable. Une autre piste possible serait de profiter du traitement de variables catégoriques natif du modèle CB, en espérant que cela puisse lui permettre de surpasser le modèle XB.

4.2 Sélection de variables avec la méthode RFE

Le tableau 4.3 vise à présenter l'apport qu'a eu la sélection de variables via l'algorithme RFE. Bien que la sélection fût faite pour les 48 instances présentées au tableau 4.1, on s'intéresse ici seulement au meilleur modèle pour chaque tâche. Le tableau 4.3 présente le nombre de variables retirées à la suite de l'application de la méthode RFE sur le nombre de variables initiales. Les deux dernières colonnes présentent le MAPE moyen obtenu durant le processus de validation croisée (voir figure 3.4 pour plus de détails).

Dans tous les cas, l'algorithme RFE a permis une amélioration de la performance du modèle. L'amélioration la plus notable est pour la tâche déblai de 1^{re} classe. C'est une amélioration relative d'un peu plus de 22%. Ce qui est d'autant plus remarquable est le nombre élevé de variables retirées. La sélection retire entre 10 et 57 variables. Notons que l'application de l'encodage *one hot* ajoute une variable pour chaque valeur trouvée. Par exemple, pour la variable `type_travaux`, il y aura autant de variables de que de types de travaux trouvés dans l'ensemble d'entraînement. Or, un type peut revenir qu'une fois ou deux, ce qui peut rendre une variable additionnelle pratiquement vide d'information. Ainsi, beaucoup de variables du genre ont été automatiquement retirées par l'algorithme. En plus d'améliorer la performance des modèles, cette étape permet d'alléger la dimension du problème diminuant ainsi le temps d'entraînement, mais également la complexité du modèle. Cela contribuera positivement à l'explicabilité du modèle.

TABLEAU 4.3 – MAPE de validation moyen (%) des meilleurs modèles avant et après l'application de la méthode RFE

Tâche	Meilleur modèle	Nb de variables retirées par RFE / variables totales	MAPE après RFE (%)	MAPE avec toutes les variables (%)
Couche de surface, enrobé	EV*	30/74	23,8	24,0
Couche de base, enrobé	CB	10/71	23,5	24,0
Armature	FA	40/60	47,2	48,0
Déblai de 1 ^{re} classe	XB	47/61	87,4	113,0
Déblai de 2 ^e classe	XB	50/67	52,6	55,0
Sous-fondation de chaussée	EV*	19/62	23,3	24,0
Fondation de chaussée	EV*	31/64	19,1	20,0
Organisation de chantier	RN*	42/67	521,5	553,0

*Tel que mentionné en méthodologie, pour les modèles RN et EV, les résultats affichés sont ceux obtenus avec XB.

Durant l'application de RFE, l'importance des variables est calculée à maintes reprises dans différentes itérations. Ainsi, en plus de sélectionner un sous-ensemble de variables, l'algorithme fournit le rang d'importance de chacune des variables contenues dans ce dernier. Cette information peut nous permettre d'analyser l'impact de l'ajout des variables externes proposé. Le tableau 4.4 présente donc le rang d'importance de chacune d'entre elles pour chaque tâche. Le rang 1 est attribué à la variable ayant le plus d'importance. Rappelons que les variables « Distance usine » et « Prix historique du bitume » ne concernent que les tâches d'enrobage, d'où les mentions « N/A ». Une cellule contenant « — » indique que cette variable ne fut pas sélectionnée dans le meilleur ensemble choisi par RFE.

Le tableau 4.4 met en évidence que la variable « Indice économique » a eu un apport notable pour sept des huit tâches. La seule tâche ne l'ayant pas sélectionnée est Déblai de 1^{re} classe, tâche qui dépend surtout de la quantité estimée, de l'année, et de la localisation, selon les explications SHAP. À première vue, il est difficile d'expliquer ce phénomène. Une explication

possible pourrait être que cette tâche a tout simplement une dépendance moins forte envers la santé économique, de sorte que l'année soit suffisamment précise pour illustrer ce facteur. Les variables « Distance usine » et « Prix historique du bitume » semblent toutes les deux avoir contribué positivement à la précision des modèles pour les tâches d'enrobage. Le prix historique du bitume est la variable la plus influente des deux variables, peu importe la tâche. La seule variable ajoutée qui ne semble pas avoir eu une influence significative dans les prédictions est la variable « Salaires construction ». Cette dernière est possiblement moins précise que l'indice économique étant donné qu'elle contient les salaires établis selon des conventions émises d'avance. Elle sera donc moins sensible aux facteurs externes pouvant influencer la santé économique locale.

TABLEAU 4.4 – Rang d'importance des variables externes ajoutées à l'expérience pour chaque tâche

Tâche	Distance usine	Prix du bitume	Indice économique	Salaires construction
Couche de surface, enrobé	7	5	6	—
Couche de base, enrobé	—	—	6	5
Armature	N/A	N/A	6	5
Déblai de 1 ^{re} classe	N/A	N/A	—	—
Déblai de 2 ^e classe	N/A	N/A	5	—
Sous-fondation de chaussée	N/A	N/A	5	—
Fondation de chaussée	N/A	N/A	5	—
Organisation de chantier	N/A	N/A	4	—

Concordant avec ceux de [33], les résultats démontrent que l'application de la méthode RFE permet d'améliorer la performance des modèles. Nous soulevons également que le nombre de variables retirées permet de simplifier le modèle et la dimension du problème. L'ajout de variables externes a également été bénéfique, mais cela dépend des variables. La variable la plus influente parmi celles ajoutées est l'indice économique.

4.3 Détection automatique des valeurs aberrantes

Cette section permet de porter un regard sur les bénéfices offerts par le traitement de valeurs aberrantes. On peut également y comparer les différentes techniques utilisées. Le tableau 4.5 présente les métriques de performance des trois différentes techniques de détection automatique des valeurs aberrantes, ainsi que le cas où aucune technique n'est appliquée. Il s'agit du MAPE moyen (%) obtenu sur les différents ensembles de validation. La figure 3.4 détaille ces ensembles, au besoin.

Pour toutes les tâches, l'une ou l'autre des techniques a permis d'améliorer le score. Cela dit, pour les tâches d'enrobage et de chaussées, l'amélioration se situe entre 0,3% et 1,3%. Elle est donc peu significative. Les améliorations les plus notables peuvent être observées pour les tâches déblai de 1^{re} classe, déblai de 2^e classe, armature et organisation de chantier. L'amélioration relative est respectivement de 15%, 22%, 36% et 74%. Pour les trois premières, cela suggère une présence de valeurs extrêmes qui nuisent à l'apprentissage des modèles. Pour la dernière, il s'agit d'une tâche plus globale. Il y a donc moins de variables fournies aux modèles pour cette tâche. Ainsi, il a été observé que plusieurs données identiques n'avaient pas le même juste prix. Des informations additionnelles expliqueraient sans doute cette différence, mais dans le format actuel, ce n'est pas visible pour les modèles. Il est donc probable que les algorithmes de détection de valeurs aberrantes aient permis de pallier ce problème, aidant ainsi grandement à la performance du modèle.

Pour quatre des huit tâches, c'est l'algorithme LOF qui fut le plus efficace pour améliorer le score. Autrement, pour les quatre tâches restantes, IF et OCSVM se répartissent équitablement les premières places. Il est ardu d'expliquer ces résultats. Cela dit, on démontre l'avantage d'étudier diverses techniques via un pipeline expérimental tel que décrit ici. En effet, les résultats sont assez diversifiés à ce niveau. La meilleure technique dépend de l'ensemble de données et il est difficile de prédire celle qui sera la plus efficace à l'avance.

TABLEAU 4.5 – MAPE de validation moyen (%) des meilleurs modèles suite au retrait (ou non) des valeurs identifiées aberrantes

Tâche	Meilleur modèle	OCSVM	LOF	IF	Aucun
Couche de surface, enrobé	EV*	32,6	23,6	23,5	23,8
Couche de base, enrobé	CB	31,2	24,0	23,6	24,2
Armature	FA	33,4	31,9	33,4	50,2
Déblai de 1 ^{re} classe	XB	85,9	79,7	88,7	94,1
Déblai de 2 ^e classe	XB	42,1	43,1	45,9	53,7
Sous-fondation de chaussée	EV*	28,5	23,6	24,1	24,3
Fondation de chaussée	EV*	20,2	18,8	20,0	20,1
Organisation de chantier	RN*	171,3	247,8	269,2	652,5

*Tel que mentionné en méthodologie, pour les modèles RN et EV, les résultats affichés sont ceux obtenus avec XB.

Les résultats montrent donc qu'aucune technique n'est à préférer, car cela dépend de la distribution des données. On remarque toutefois que la suppression des valeurs aberrantes est une technique permettant d'améliorer la performance des modèles, peu importe ce dernier. Le degré d'amélioration peut varier, sans toutefois dégrader les performances en ce qui concerne nos huit ensembles de données.

4.4 Analyse comparative des techniques de validation croisée

Dans la littérature, différentes techniques de validation ont été proposées. En particulier, la technique utilisée pour séparer l'ensemble d'entraînement afin d'en conserver une partie pour évaluer la performance diffère d'une expérience à l'autre. Parfois fait de manière aléatoire, et d'autres selon un facteur temporel. L'hypothèse initiale stipulait que puisque pour notre problème la distribution des données varie d'une année à l'autre, il serait plus adéquat de séparer les ensembles selon un facteur temporel. Or, les deux manières de faire ont été testées. Le tableau 4.6 vise à comparer la performance des meilleurs modèles pour chaque tâche, selon les deux méthodes de séparation. La performance affichée est celle calculée sur l'ensemble de test, soit les données de l'année 2021. L'entraînement fut effectué sur les années 2014 à 2020. La meilleure performance pour chaque tâche apparaît en caractère gras.

TABLEAU 4.6 – MAPE (%) des meilleurs modèles selon les validations croisées aléatoires et temporelles

Tâche	Temporelle		Aléatoire	
	Modèle	MAPE (%)	Modèle	MAPE(%)
Couche de surface, enrobé	EV	20,63	EV	23,44
Couche de base, enrobé	CB	19,57	XB	18,99
Armature	FA	26,69	FA	28,55
Déblai de 1 ^{re} classe	XB	52,20	XB	52,96
Déblai de 2 ^e classe	XB	37,04	EV	35,06
Sous-fondation de chaussée	EV	19,96	XB	20,18
Fondation de chaussée	EV	16,96	XB	17,30
Organisation de chantier	RN	82,15	XB	83,19
Moyenne	—	34,40	—	34,96

Pour six des huit tâches, il semble y avoir une légère tendance favorisant la méthode de séparation temporelle. Cela dit, la différence absolue entre les performances obtenues selon les deux méthodes ne dépasse pas 2% pour l'ensemble des tâches. Celle entre les performances moyennes n'est que de 0,56%. La séparation temporelle est donc plus appropriée en moyenne pour les données que nous avons, mais il ne s'agit pas d'un facteur prépondérant dans l'optimisation du modèle. Une piste intéressante pourrait être de comparer la validation croisée telle que vue ici avec une validation standard, et de comparer encore une fois une séparation des ensembles aléatoire et temporelle.

4.5 Comparaison de la performance de l'IA avec celle de l'estimation humaine

Le tableau 4.7 met en relation les performances de l'estimation humaine provenant du MTQ avec celles des modèles d'IA. La métrique utilisée est le MAPE et il a été calculé à partir des données de l'ensemble de test dans les deux cas. Pour les modèles d'IA, il s'agit de la meilleure version du modèle optimisé. Pour sept des huit tâches, l'IA effectue une erreur moindre que celle de l'estimation humaine. La seule tâche pour laquelle l'estimation surpasse l'IA, et ce, de manière significative, est la tâche organisation de chantier. Cette tâche était en effet plutôt particulière puisque contrairement aux autres, elle est plus globale. Les mêmes variables de base ont été fournies pour cette tâche. Or, tel que mentionné un peu plus tôt, ce lot de variables ne semble pas suffisant pour distinguer toutes les particularités d'une soumission. Il a été observé que plusieurs lignes identiques avaient un prix estimé par l'humain complètement différent (parfois le double). Il s'agit d'une incohérence flagrante qui survient de multiples fois dans notre ensemble. Il n'est donc pas étonnant que l'estimation humaine performe mieux

pour cette tâche, puisque l'humain a accès à plus d'informations que ce qui est fourni au modèle. De telles incohérences n'apparaissent pas pour d'autres tâches.

Pour mieux apprécier cette amélioration, la dernière colonne à droite du tableau 4.7 permet de montrer que pour les tâches étudiées, l'IA réduit l'erreur humaine d'une proportion variante entre 3,42% et 31,56%, en excluant la tâche organisation de chantier. La meilleure amélioration est pour la tâche armature.

TABLEAU 4.7 – Résumé des meilleures performances des modèles d'IA comparées à celles de l'estimation humaine

Tâche	Modèle	MAPE du modèle (%)	MAPE de l'estimation humaine (%)	Amélioration relative (%)
Couche de surface, enrobé	EV	20,63	22,63	8,84
Couche de base, enrobé	CB	19,57	21,47	8,85
Armature	FA	26,69	39,00	31,56
Déblai de 1 ^{re} classe	XB	52,20	64,79	19,43
Déblai de 2 ^e classe	XB	37,04	38,35	3,42
Sous-fondation de chaussée	EV	19,96	28,25	29,35
Fondation de chaussée	EV	16,96	22,08	23,19
Organisation de chantier	RN	82,15	43,35	-89,50

En somme, l'expérience démontre que pour sept des huit tâches étudiées, l'IA apporte une amélioration aux estimations humaines. Or, pour la tâche organisation de chantier, il faudrait fournir plus de variables aux modèles afin d'observer le potentiel de l'IA. Également, les résultats mettent en évidence les faiblesses des RN, soit la difficulté d'entraînement et d'optimisation. À l'inverse, les prouesses du modèle XB ont pu être confirmées, étant globalement le meilleur modèle ayant été étudié. De plus, nous avons pu voir qu'une sélection de variables rigoureuse et un bon traitement des valeurs aberrantes permettent d'améliorer la qualité des prédictions de l'IA.

CHAPITRE 5 INTERPRÉTATION DES RÉSULTATS

Outre la performance, une autre qualité à laquelle on accorde une attention particulière est le niveau d'explicabilité d'un modèle. Nous avons discuté précédemment de l'utilisation des méthodes SHAP, qui permettent d'évaluer l'impact de chaque variable explicative sur les prédictions. Les valeurs SHAP permettent d'abord de faire cette évaluation de manière locale. C'est-à-dire que l'impact des variables est calculé pour chacune des prédictions individuelles. Dans notre cas, ce sera pour chaque soumission. En effet, selon la soumission, ce ne seront pas toujours les mêmes variables qui auront plus ou moins d'impact sur la prédiction finale.

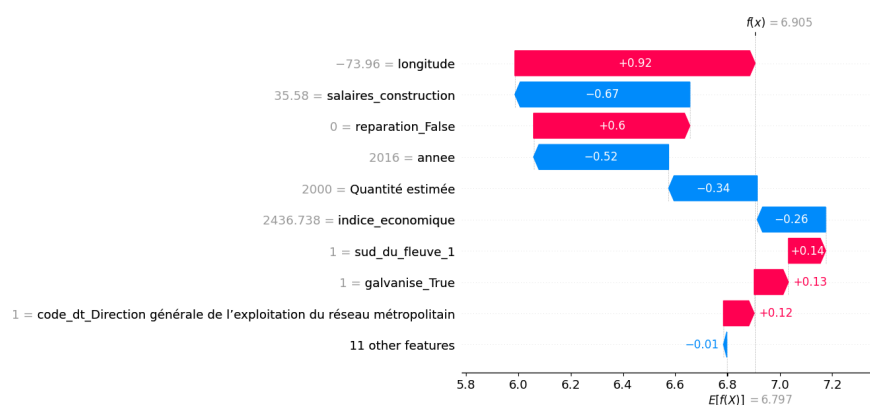


FIGURE 5.1 – Analyse de la prédiction du prix unitaire d'une seule soumission du modèle FA pour la tâche Armature.

La figure 5.1 présente un exemple d'interprétation d'une soumission aléatoire pour la tâche armature. On observe, selon le modèle SHAP, que toutes les prédictions commencent avec le prix unitaire moyen (espérance). En l'occurrence, il s'agit ici de 6,797\$, sous l'axe des abscisses (en bas de la figure). Ensuite, selon la valeur des différentes variables, la prédiction du prix unitaire de cette soumission se disperse autour de la moyenne. Allant de bas en haut sur la figure 5.1, on observe d'abord que onze variables explicatives non identifiées sur le graphique (« 11 other features ») réduisent le prix de 0,01\$. Il ne s'agit donc pas de variables ayant des valeurs influentes pour cette soumission. Ensuite, la variable « galvanise » est vraie (1), indiquant que l'armature est en acier galvanisé. Cela augmenterait le prix de 0,12\$. En montant à la variable « sud_du_fleuve ». Elle est égale à 1, ce qui indique un projet dans la grande région de Montréal (ni au nord ni au sud du fleuve). Cela augmenterait le prix de 0,12\$. On peut suivre cette même logique jusqu'à longitude égale à -73.96 (ouest du réseau métropolitain de Montréal), valeur qui aurait le plus d'impact pour cette soumission selon le modèle. Cette valeur importante aurait indiqué au modèle une augmentation de prix de

0,91\$. La somme de tous ces cheminements mène au résultat final de 6,905\$ (haut de la figure).

Afin d’alléger ce rapport, on ne considérera qu’un seul exemple d’explication d’une prédiction locale. Un exemple d’explication locale pour chaque tâche est fourni en annexes. Cela dit, SHAP propose également une manière élégante de résumer l’ensemble des explications locales en un seul graphique. Cela permet d’avoir une interprétation globale du modèle. La figure 5.2 en présente un exemple pour la tâche armature. Un exemple pour chaque tâche se trouve en annexes.

Si nous prenons la figure 5.2 en exemple, chaque point qu’on observe sur le graphique correspond à l’impact enregistré pour une prédiction en particulier. La couleur du point indique si la valeur de cette variable est plutôt basse (bleue) ou plutôt élevée (rose). On voit ainsi dans l’ensemble l’impact général de chaque variable, selon sa valeur. Le graphique à la figure 5.2 met ici en valeur que la variable ayant souvent le plus d’impact est la quantité estimée. On voit que les très grandes valeurs (rose) pour cette variable tirent l’estimation du prix unitaire vers le bas et vice-versa. C’est intéressant puisque cela semble concorder avec l’intuition que plus on achète un item en grande quantité, plus le prix unitaire serait avantageux.

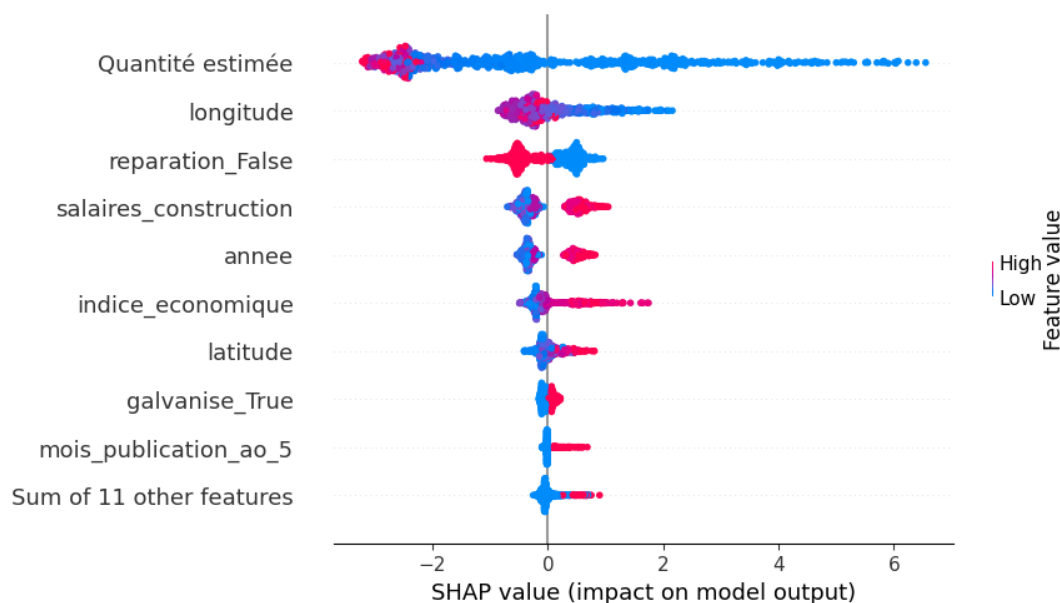


FIGURE 5.2 – Analyse de l’impact SHAP du modèle FA pour la tâche Armature.

Pour interpréter les variables de type binaire (vrai ou faux), le bleu sur le graphique correspond tout simplement à faux et le rose à vrai. Par exemple, « reparation_False » indique que la description du dossier ne contenait pas de mots liés au thème de la réparation lorsque le point est rose (vrai). On voit donc une claire tendance que l’absence de réparation tend

à diminuer le prix unitaire et inversement. C'est d'ailleurs une variable très influente pour cette tâche. Il est intéressant d'observer que parmi les variables les plus influentes se trouvent l'emplacement (longitude et latitude), la quantité estimée et des indices sur la situation économique ou l'inflation (année, mois, indice économique, salaires). À première vue, cela concorde avec l'intuition humaine. Il est également intéressant de constater que les salaires de construction ainsi que l'indice économique se trouvent parmi les variables les plus importantes pour cette tâche. En effet, l'algorithme de sélection de variables RFE conserve le meilleur sous-ensemble de variables. Cela veut donc dire que ces variables contribuent à l'amélioration de la qualité des prédictions.

Les exemples vus aux figures 5.1 et 5.2 sont pour le modèle FA, mais le graphique serait similaire, peu importe le modèle ayant obtenu la meilleure performance. Des exemples avec d'autres modèles sont d'ailleurs présentés en annexes. On voit ainsi la puissance des techniques agnostiques au modèle.

CHAPITRE 6 CONCLUSION

L'estimation préliminaire des coûts joue un rôle important quant au succès d'un projet de construction d'infrastructures routières. Au MTQ, la tâche d'estimation repose sur une analyse statistique des coûts historiques, combinée à l'avis de spécialistes. Cette technique a toutefois été remise en question par le VGQ lors d'un audit de 2017. Les inquiétudes soulevées concernaient autant l'exactitude que la justification derrière ces estimations. Une solution basée sur l'apprentissage supervisé explicable a été proposée pour répondre à ces questionnements. La présente section résume cette approche, les conclusions qui en ont découlé et ses limites. On termine également sur des pistes d'améliorations pouvant être suivies dans des travaux futurs.

6.1 Synthèse des travaux

Le point de départ de cette expérience fut l'obtention d'une base de données contenant plus de 10,000 soumissions. Or, un projet de construction étant divisé par tâches, cette base de données fut segmentée en huit ensembles. La première contribution apportée fut la préparation de ces ensembles en vue de leur utilisation en IA. Cela implique la suppression des données manquantes et la conversion des données catégoriques vers un format numérique. Nous avons également ajouté quatre variables externes à la base de données initiale afin d'en bonifier le contenu. Plus spécifiquement, il s'agit de l'ajout d'un indice économique, des salaires des employés de la construction, des prix historiques du bitume et de la distance entre les chantiers et l'usine d'enrobage la plus proche.

Ensuite, un outil en lignes de commandes fut développé afin de tirer le plus grand potentiel de l'IA. Ce dernier permet l'optimisation de la performance des modèles. Il permet d'appliquer diverses techniques d'amélioration de la qualité des données. En particulier, on applique l'algorithme RFE afin de sélectionner le meilleur sous-ensemble de variables parmi celles disponibles. On teste également deux techniques de mise à l'échelle. Trois algorithmes de détection automatique des valeurs aberrantes sont étudiés. Étant configurable, l'outil supporte deux méthodes de validation croisée pouvant ainsi être comparées. Il comporte également l'implémentation de six modèles d'IA, soient AD, FA, XB, CB, RN et EV. Cette diversité permet un comparatif de la performance de différents modèles. L'outil prend alors la forme d'un pipeline dans lequel chaque expérience optimise six modèles selon les différentes techniques de prétraitement supportées.

Le pipeline fut lancé avec chacune des huit tâches étudiées. Les résultats des différentes expériences ont révélé que les prétraitements les plus efficaces pour améliorer la précision des modèles sont l'algorithme RFE, la suppression des données aberrantes via l'un ou l'autre des algorithmes étudiés et l'ajout d'un indice économique à la base de données. Au final, ce pipeline a permis le développement de huit modèles d'IA optimisés (un pour chaque tâche). Bien que la nature du meilleur modèle varie d'une tâche à l'autre, on identifie l'algorithme XB comme étant le plus adapté à notre problème. Il s'agit d'un modèle ayant constamment démontré une bonne précision et un temps d'entraînement presque instantané. Sept modèles sur les huit développés permettent de faire des estimations plus précises que celles de l'humain dans des proportions variant entre 3,42% et 31,56%.

Afin d'apporter une justification à ces estimations, un deuxième outil en lignes de commandes fut développé dans l'optique d'expliquer les prédictions d'un modèle selon la méthode SHAP. Cet outil permet de générer deux types d'explications visuelles. La première explication montre l'apport que chaque variable a eu sur l'estimation finale d'un certain prix. La deuxième explication résume l'impact des différentes variables sur un ensemble de prédictions. Seize explications ont donc été générées par cet outil (deux explications pour chaque tâche). La cohérence des explications a d'ailleurs été validée par des experts en estimation provenant du MTQ.

En somme, plus de 2000 lignes de code réparties dans 26 fichiers ont été écrites. Ces outils informatiques ont généré des modèles qui fournissent des estimations précises et justifiées, répondant donc aux inquiétudes initiales soulevées par le VGQ.

6.2 Limitations de la solution proposée

La solution proposée connaît toutefois certaines limites. Tout d'abord, la précision d'un des huit modèles développés n'a pas pu être suffisante. Il semblerait que les variables fournies au modèle n'ont pas de relation évidente avec le prix estimé pour cette tâche. De manière plus générale, les modèles d'IA ne peuvent effectivement pas réagir à des facteurs externes qui ne seraient pas contenus dans la base de données. Par exemple, les modèles n'ont aucune manière de savoir qu'il y aurait une pandémie mondiale, ce qui influence toutefois grandement les prix soumis. Aussi, bien que l'IA s'annonce prometteuse pour la suite, son implémentation concrète est un défi logiciel d'envergure. Présentement, avec les ressources disponibles, l'extraction des données fut faite à la main pour toutes les tâches, procédure qui fut plutôt longue. Éventuellement, le logiciel effectuant l'entraînement des modèles s'abreuvait directement sur les données de BDP via une API web. Les modèles entraînés devront également être sauvegardés de manière plus robuste dans une base de données pour pouvoir accéder à

des prédictions en temps réel via des API. De plus, il faut penser à la maintenance desdits modèles. À mesure que des données sont disponibles, il faudrait refaire des entraînements et mettre à jour les prédictions. Une vérification systématique de la performance du modèle devrait également être faite, pour s'assurer que le modèle a toujours des métriques convenables à la suite de sa mise à jour. Ce ne sont ici que quelques exemples des différentes considérations qu'il y aurait à prendre pour mettre le tout en place. Il s'agit d'un véritable défi logiciel sur lequel beaucoup de temps et d'énergie devront être mis.

Il faut également souligner que l'IA s'est montrée efficace dans le contexte de sept tâches étudiées, qui contiennent beaucoup de données numériques. Or, il existe des dizaines d'autres tâches au MTQ, et elles ne contiennent pas nécessairement un nombre suffisant de données.

La technique utilisée pour les justifications connaît aussi certaines limites. En effet, certains auteurs ont observé qu'ils étaient capables de modifier une explication SHAP en ajoutant malicieusement des données ayant certaines propriétés à l'ensemble [52]. Il faudrait donc s'assurer que des soumissionnaires ne profitent pas de cette faiblesse pour expliquer un coût d'une manière alternative de sorte à les avantager.

Une autre limitation importante est liée à la cible à prédire. La définition de ce qu'est la parfaite estimation est nébuleuse. Dans le contexte du MTQ, il a été choisi de considérer la moyenne des trois meilleures soumissionnaires comme la meilleure estimation. Les modèles ont donc été entraînés à prédire cette valeur. Or, cette valeur est calculée selon une hypothèse et elle n'est possiblement pas aussi précise que ce qui est attendu.

6.3 Améliorations et travaux futurs

Plusieurs améliorations découlent directement des limites identifiées. Entre autres, il serait important de se pencher sur la fraude possible entourant les explications SHAP. Le risque semble à première vue faible, puisqu'il y a peu de motivation à lancer une telle attaque. De plus, il reste à voir s'il serait possible pour un soumissionnaire externe au MTQ d'ajouter autant de données. Bref, ce ne sont que des suppositions, mais il serait intéressant de se pencher davantage sur les risques possibles entourant cette limite, ainsi que sur la faisabilité d'une telle attaque dans ce contexte.

Alors que l'IA s'est montrée efficace pour sept tâches, il faudrait étendre les expérimentations à d'autres tâches au sein du MTQ. Aussi, il faudrait étudier plus en profondeur la tâche organisation de chantier, pour laquelle l'IA s'est montrée inefficace. En particulier, il serait intéressant de se pencher sur les variables fournies aux modèles pour cette tâche afin de cibler d'où provient le manque d'informations dans ce cas.

Maintenant que nous savons que l'apprentissage supervisé offre réellement un potentiel permettant d'améliorer la précision et la transparence des explications, cela justifie l'exploration de l'intégration concrète de la solution dans les logiciels internes du MTQ. Il s'agit toutefois d'un grand défi logiciel et il faudrait se pencher davantage sur la question dans d'éventuelles recherches. La recherche actuelle et le code associé sont de bonnes bases pour débiter une telle exploration.

Enfin, l'outil développé dans le cadre de cette recherche est fait de manière à être agnostique à l'ensemble de données. Cela permet de nombreuses possibilités de répéter l'expérience à d'autres domaines au sein du gouvernement du Québec. Pour nommer quelques exemples, on pourrait refaire l'expérience en estimation de coûts de construction de bâtiments ou même de développement de logiciels.

RÉFÉRENCES

- [1] V. Kouskoulas et E. Koehn, “Predesign cost-estimation function for buildings,” *Journal of the Construction Division*, vol. 100, n°. 4, p. 589–604, Dec 1974. [En ligne]. Disponible : <https://ascelibrary.org/doi/abs/10.1061/JCCEAZ.0000461>
- [2] P. A. Bowen et P. J. Edwards, “Cost modelling and price forecasting : practice and theory in perspective,” *Construction Management and Economics*, vol. 3, n°. 3, p. 199–215, Dec 1985. [En ligne]. Disponible : <https://doi.org/10.1080/01446198500000015>
- [3] C. Gouvernement du Nouveau-Brunswick, “Étapes de la construction d’une route,” May 2020. [En ligne]. Disponible : https://www2.gnb.ca/content/gnb/fr/ministeres/mti/Autoroute_route/content/construction_une_route.html
- [4] V. g. d. Québec, “Mission.” [En ligne]. Disponible : <https://www.vgq.qc.ca/fr/organisation/mission>
- [5] M. Shan *et al.*, “Assessing collusion risks in managing construction projects using artificial neural network,” *Technological and Economic Development of Economy*, vol. 24, n°. 5, p. 2003–2025, 2018.
- [6] V. général du Québec, *Rapport du Vérificateur général du Québec à l’Assemblée nationale pour l’année 2017-2018 - Audit particulier (partie 2) Ministère des Transports, de la Mobilité durable et de l’Électrification des transports : gestion contractuelle - Faits saillants*. Bibliothèque et Archives nationales du Québec, 2017.
- [7] R. A. McKim, “Neural network applications to cost engineering,” *Cost Engineering*, vol. 35, n°. 7, p. 31, 1993.
- [8] M. Arafa et M. Alqedra, “Early stage cost estimation of buildings construction projects using artificial neural networks,” *Journal of Artificial Intelligence*, vol. 4, n°. 1, 2011.
- [9] V. Belle et I. Papantonis, “Principles and practice of explainable machine learning,” *Frontiers in big Data*, p. 39, 2021.
- [10] J. R. Quinlan, “Induction of decision trees,” *Machine learning*, vol. 1, n°. 1, p. 81–106, 1986.
- [11] Y. LeCun, Y. Bengio et G. Hinton, “Deep learning,” *nature*, vol. 521, n°. 7553, p. 436–444, 2015.
- [12] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, n°. 1, p. 5–32, Oct 2001. [En ligne]. Disponible : <https://doi.org/10.1023/A:1010933404324>

- [13] 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, Aug 2016, arXiv : 1603.02754. [En ligne]. Disponible : <http://arxiv.org/abs/1603.02754>
- [14] L. Prokhorenkova *et al.*, “Catboost : unbiased boosting with categorical features,” *arXiv :1706.09516 [cs]*, Jan 2019, arXiv : 1706.09516. [En ligne]. Disponible : <http://arxiv.org/abs/1706.09516>
- [15] T. G. Dietterich, “Ensemble methods in machine learning,” dans *International workshop on multiple classifier systems*. Springer, 2000, p. 1–15.
- [16] S. Lundberg et S.-I. Lee, “A unified approach to interpreting model predictions,” *arXiv :1705.07874 [cs, stat]*, Nov 2017, arXiv : 1705.07874. [En ligne]. Disponible : <http://arxiv.org/abs/1705.07874>
- [17] F. Khosrowshahi et A. Kaka, “Estimation of project total cost and duration for housing projects in the u.k.” *Building and Environment*, 1996.
- [18] H. Li, “Neural networks for construction cost estimation,” *Building Research & Information*, vol. 23, n°. 5, p. 279–284, Oct 1995. [En ligne]. Disponible : <https://doi.org/10.1080/09613219508727476>
- [19] J. de la Garza M et K. G. Rouhana, “Neural networks versus parameter-based applications in cost,” *American Association of Cost Engineers*, vol. 37, n°. 2, p. 14, 1995. [En ligne]. Disponible : <https://www.proquest.com/scholarly-journals/neural-networks-versus-parameter-based/docview/220470091/se-2?accountid=40695>
- [20] G.-H. Kim, S.-H. An et K.-I. Kang, “Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning,” *Building and Environment*, vol. 39, n°. 10, p. 1235–1242, Oct 2004. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S0360132304000812>
- [21] M. Barakchi, O. Torp et A. M. Belay, “Cost estimation methods for transport infrastructure : A systematic literature review,” *Procedia Engineering*, vol. 196, p. 270–277, Jan 2017. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S1877705817330588>
- [22] G. Mahalakshmi et C. Rajasekaran, “Early cost estimation of highway projects in india using artificial neural network,” dans *Sustainable Construction and Building Materials*, B. B. Das et N. Neithalath, édit. Singapore : Springer, 2019, p. 659–672.
- [23] G. N. Aretoulis, “Neural network models for actual cost prediction in greek public highway projects,” *International Journal of Project Organisation and*

- Management*, vol. 11, n^o. 1, p. 41–64, Jan 2019. [En ligne]. Disponible : <https://www.inderscienceonline.com/doi/abs/10.1504/IJPOM.2019.098712>
- [24] C. L. C. Roxas *et al.*, “Modeling road construction project cost in the philippines using the artificial neural network approach,” dans *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, Nov 2019, p. 1–5.
- [25] K. Tijanić, D. Car-Pušić et M. Šperac, “Cost estimation in road construction using artificial neural network,” *Neural Computing and Applications*, vol. 32, n^o. 13, p. 9343–9355, Jul 2020. [En ligne]. Disponible : <https://doi.org/10.1007/s00521-019-04443-y>
- [26] S. Kim *et al.*, “Improving accuracy in predicting city-level construction cost indices by combining linear arima and nonlinear anns,” *Journal of Management in Engineering*, vol. 38, n^o. 2, p. 04021093, Mar 2022. [En ligne]. Disponible : <https://ascelibrary.org/doi/full/10.1061/%28ASCE%29ME.1943-5479.0001008>
- [27] B. Xiong *et al.*, “Hybrid approach to reducing estimating overfitting and collinearity,” *Engineering, Construction and Architectural Management*, vol. 26, n^o. 10, p. 2170–2185, Jan 2019. [En ligne]. Disponible : <https://doi.org/10.1108/ECAM-08-2018-0353>
- [28] S. Petrusheva, D. Car-Pušić et V. Zileska-Pancovska, “Support vector machine based hybrid model for prediction of road structures construction costs,” *IOP Conference Series : Earth and Environmental Science*, vol. 222, p. 012010, Jan 2019. [En ligne]. Disponible : <https://iopscience.iop.org/article/10.1088/1755-1315/222/1/012010>
- [29] H. H. Elmousalami, “Artificial intelligence and parametric construction cost estimate modeling : State-of-the-art review,” *Journal of Construction Engineering and Management*, vol. 146, n^o. 1, p. 03119008, Jan 2020. [En ligne]. Disponible : <https://ascelibrary.org/doi/full/10.1061/%28ASCE%29CO.1943-7862.0001678>
- [30] J. H. Friedman, “Greedy function approximation : A gradient boosting machine.” *The Annals of Statistics*, vol. 29, n^o. 5, p. 1189–1232, Oct 2001. [En ligne]. Disponible : <https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full>
- [31] R. Shwartz-Ziv et A. Armon, “Tabular data : Deep learning is not all you need,” *arXiv*, n^o. arXiv :2106.03253, Nov 2021, arXiv :2106.03253 [cs]. [En ligne]. Disponible : <http://arxiv.org/abs/2106.03253>
- [32] S. B. Jabeur *et al.*, “Catboost model and artificial intelligence techniques for corporate failure prediction,” *Technological Forecasting and Social Change*, vol. 166, p. 120658,

- May 2021. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S0040162521000901>
- [33] A. Mahdavian *et al.*, “Data-driven predictive modeling of highway construction cost items,” *Journal of Construction Engineering and Management*, vol. 147, n°. 3, p. 04020180, Mar 2021. [En ligne]. Disponible : <https://ascelibrary.org/doi/full/10.1061/%28ASCE%29CO.1943-7862.0001991>
- [34] G. Hinton, O. Vinyals et J. Dean, “Distilling the knowledge in a neural network,” *arXiv*, n°. arXiv :1503.02531, Mar 2015, arXiv :1503.02531 [cs, stat]. [En ligne]. Disponible : <http://arxiv.org/abs/1503.02531>
- [35] N. Frosst et G. Hinton, “Distilling a neural network into a soft decision tree,” *arXiv :1711.09784 [cs, stat]*, Nov 2017, arXiv : 1711.09784. [En ligne]. Disponible : <http://arxiv.org/abs/1711.09784>
- [36] T. Vidal, T. Pacheco et M. Schiffer, “Born-again tree ensembles,” *arXiv :2003.11132 [cs, stat]*, Aug 2020, arXiv : 2003.11132. [En ligne]. Disponible : <http://arxiv.org/abs/2003.11132>
- [37] L. S. Shapley, *17. A Value for n-Person Games*. Princeton University Press, Dec 1953, p. 307–318. [En ligne]. Disponible : <https://www.degruyter.com/document/doi/10.1515/9781400881970-018/html>
- [38] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature Machine Intelligence*, vol. 1, n°. 5, p. 206–215, May 2019. [En ligne]. Disponible : <https://www.nature.com/articles/s42256-019-0048-x>
- [39] F. de solidarité FTQ, “Iq30 : Tout savoir sur l’indice boursier du québec.” [En ligne]. Disponible : <https://www.fondsftq.com/fr-ca/particuliers/epargne-positive/iq30-indice-boursier-quebec>
- [40] Google. [En ligne]. Disponible : <https://developers.google.com/maps/documentation/distance-matrix/overview>
- [41] OSRM, “General options.” [En ligne]. Disponible : <http://project-osrm.org/docs/v5.5.1/api/#general-options>
- [42] I. Guyon *et al.*, “Gene selection for cancer classification using support vector machines,” *Machine Learning*, vol. 46, n°. 1, p. 389–422, Jan 2002. [En ligne]. Disponible : <https://doi.org/10.1023/A:1012487302797>
- [43] Scikit, 2022. [En ligne]. Disponible : <https://scikit-learn.org/stable/>
- [44] F. T. Liu, K. M. Ting et Z.-H. Zhou, “Isolation forest,” dans *2008 Eighth IEEE International Conference on Data Mining*, Dec 2008, p. 413–422.

- [45] M. M. Breunig *et al.*, “Lof : identifying density-based local outliers,” *ACM SIGMOD Record*, vol. 29, n°. 2, p. 93–104, May 2000. [En ligne]. Disponible : <https://doi.org/10.1145/335191.335388>
- [46] R. Zhang *et al.*, “One class support vector machine for anomaly detection in the communication network performance data,” dans *Proceedings of the 5th conference on Applied electromagnetics, wireless and optical communications*, ser. ELECTROSCIENCE’07. Stevens Point, Wisconsin, USA : World Scientific and Engineering Academy and Society (WSEAS), Dec 2007, p. 31–37.
- [47] “State-of-the-art open-source gradient boosting library with categorical features support.” [En ligne]. Disponible : <https://catboost.ai/>
- [48] “Xgboost documentation.” [En ligne]. Disponible : <https://xgboost.readthedocs.io/en/stable/>
- [49] “Pytorch.” [En ligne]. Disponible : <https://pytorch.org/>
- [50] “Welcome to the shap documentation.” [En ligne]. Disponible : <https://shap.readthedocs.io/en/latest/index.html>
- [51] “Welcome to click¶.” [En ligne]. Disponible : <https://click.palletsprojects.com/en/8.1.x/>
- [52] D. Slack *et al.*, “Fooling lime and shap : Adversarial attacks on post hoc explanation methods,” *arXiv :1911.02508 [cs, stat]*, Feb 2020, arXiv : 1911.02508. [En ligne]. Disponible : <http://arxiv.org/abs/1911.02508>

ANNEXE A HYPERPARAMÈTRES DES MEILLEURS MODÈLES

TABLEAU A.1 – Valeurs de hyperparamètres du modèle EnsembleVoter entraîné sur la tâche couche de surface, enrobé. Les paramètres non mentionnés portent les valeurs par défaut.

Modèle	Paramètre	Valeur
Arbre de décision	min_samples_split	2
	max_features	0.3
	ccp_alpha	0.0
	min_samples_leaf	8
	max_depth	15
Réseau de neurones	lr	1e-3
	batch_size	4
	epochs	2000
	layers [size, dropout]	[20, 0.6], [10, 0.6]
Forêt aléatoire	min_samples_split	2
	max_features	0.8
	ccp_alpha	0.0
	min_samples_leaf	6
	max_depth	None
XGBoost	objective	reg :squarederror
	max_depth	6
	min_child_weight	1
	subsample	0.8
	colsample_bytree	1
	gamma	1
	reg_lambda	1
	n_estimators	100
	alpha	5
learning_rate	0.1	
CatBoost	od_type	IncToDec
	od_pval	0.04
	l2_leaf_reg	1

TABLEAU A.2 – Valeurs de hyperparamètres du modèle CatBoost entraîné sur la tâche couche de base, enrobé. Les paramètres non mentionnés portent les valeurs par défaut.

Paramètre	Valeur
od_type	IncToDec
learning_rate	0.01
od_pval	0.04
l2_leaf_reg	1
random_seed	42

TABLEAU A.3 – Valeurs de hyperparamètres du modèle forêt aléatoire entraîné sur la tâche armature. Les paramètres non mentionnés portent les valeurs par défaut.

Paramètre	Valeur
min_samples_split	2
max_features	0.5
ccp_alpha	0.0
min_samples_leaf	8
max_depth	None
random_state	42

TABLEAU A.4 – Valeurs de hyperparamètres du modèle XGBoost entraîné sur la tâche déblais de 1^{re} classe. Les paramètres non mentionnés portent les valeurs par défaut.

Paramètre	Valeur
objective	reg :squarederror
eval_metric	mape
max_depth	7
min_child_weight	1
subsample	0.7
colsample_bytree	0.8
gamma	0
reg_lambda	1
n_estimators	100
alpha	0
learning_rate	0.01
seed	42

TABLEAU A.5 – Valeurs de hyperparamètres du modèle XGBoost entraîné sur la tâche déblais de 2^e classe. Les paramètres non mentionnés portent les valeurs par défaut.

Paramètre	Valeur
objective	reg :squarederror
eval_metric	mape
max_depth	4
min_child_weight	1
subsample	0.9
colsample_bytree	1
gamma	0
reg_lambda	10
n_estimators	100
alpha	0
learning_rate	0.1
seed	42

TABLEAU A.6 – Valeurs de hyperparamètres du modèle EnsembleVoters entraîné sur la tâche sous-fondation de chaussée. Les paramètres non mentionnés portent les valeurs par défaut.

Modèle	Paramètre	Valeur
Arbre de décision	min_samples_split	2
	max_features	0.3
	ccp_alpha	0.0
	min_samples_leaf	6
	max_depth	None
Réseau de neurones	lr	1e-4
	batch_size	4
	epochs	2000
	layers [size, dropout]	[30, 0.6], [10, 0.4]
Forêt aléatoire	min_samples_split	8
	max_features	auto
	ccp_alpha	0.0
	min_samples_leaf	6
	max_depth	None
XGBoost	objective	reg :squarederror
	max_depth	5
	min_child_weight	5
	subsample	0.7
	colsample_bytree	0.6
	gamma	0
	reg_lambda	1
	n_estimators	100
	alpha	1
	learning_rate	0.1
CatBoost	od_type	IncToDec
	od_pval	0.04
	l2_leaf_reg	5

TABLEAU A.7 – Valeurs de hyperparamètres du modèle EnsembleVoters entraîné sur la tâche fondation de chaussée. Les paramètres non mentionnés portent les valeurs par défaut.

Modèle	Paramètre	Valeur
Arbre de décision	min_samples_split	2
	max_features	0.1
	ccp_alpha	0.0095
	min_samples_leaf	6
	max_depth	None
Réseau de neurones	lr	1e-4
	batch_size	4
	epochs	2000
	layers [size, dropout]	[30, 0.6], [10, 0.4]
Forêt aléatoire	min_samples_split	2
	max_features	0.5
	ccp_alpha	0.0
	min_samples_leaf	4
	max_depth	None
XGBoost	objective	reg :squareerror
	max_depth	4
	min_child_weight	1
	subsample	1
	colsample_bytree	0.6
	gamma	0
	reg_lambda	10
	n_estimators	100
	alpha	0
	learning_rate	0.1
CatBoost	od_type	IncToDec
	od_pval	0.04
	learning_rate	0.01
	l2_leaf_reg	1

TABLEAU A.8 – Valeurs de hyperparamètres du modèle réseau de neurones entraîné sur la tâche organisation de chantier. Les paramètres non mentionnés portent les valeurs par défaut.

Paramètre	Valeur
lr	1e-4
batch_size	4
epochs	2000
layers [size, dropout]	[30, 0.6], [10, 0.4]

ANNEXE B EXPLICATIONS SHAP LOCALES ET GLOBALES PAR TÂCHES

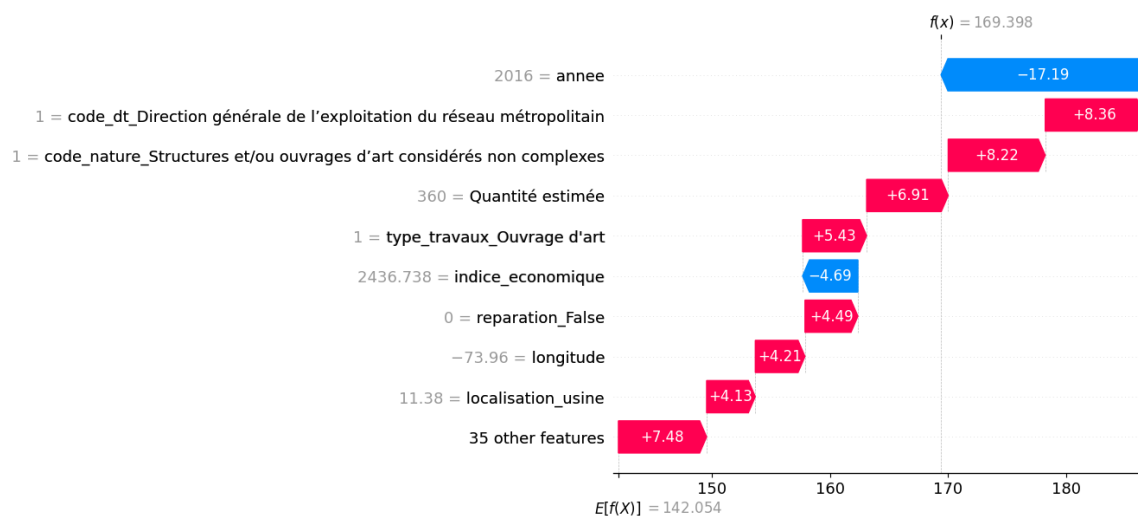


FIGURE B.1 – Analyse de la prédiction du prix unitaire d'une seule soumission du modèle EnsembleVoters pour la tâche couche de surface, enrobé.

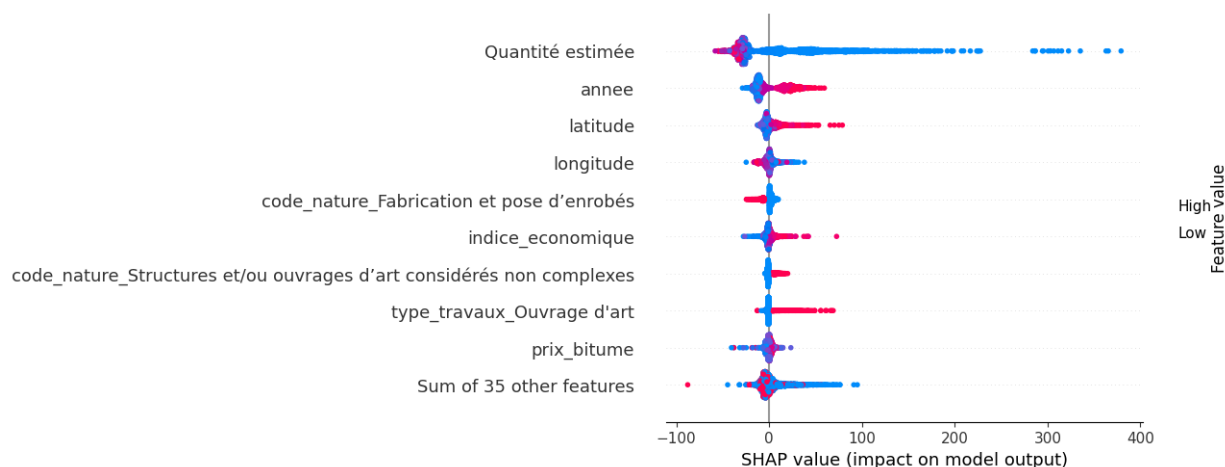


FIGURE B.2 – Analyse de l'impact des variables du modèle EnsembleVoters pour la tâche couche de surface, enrobé.

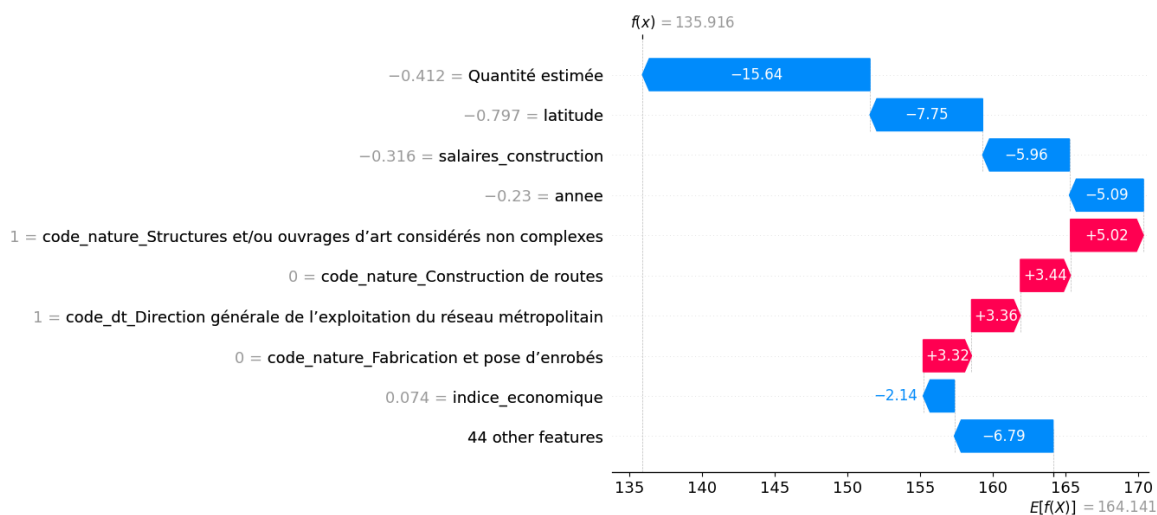


FIGURE B.3 – Analyse de la prédiction du prix unitaire d'une seule soumission du modèle CatBoost pour la tâche couche de base, enrobé.

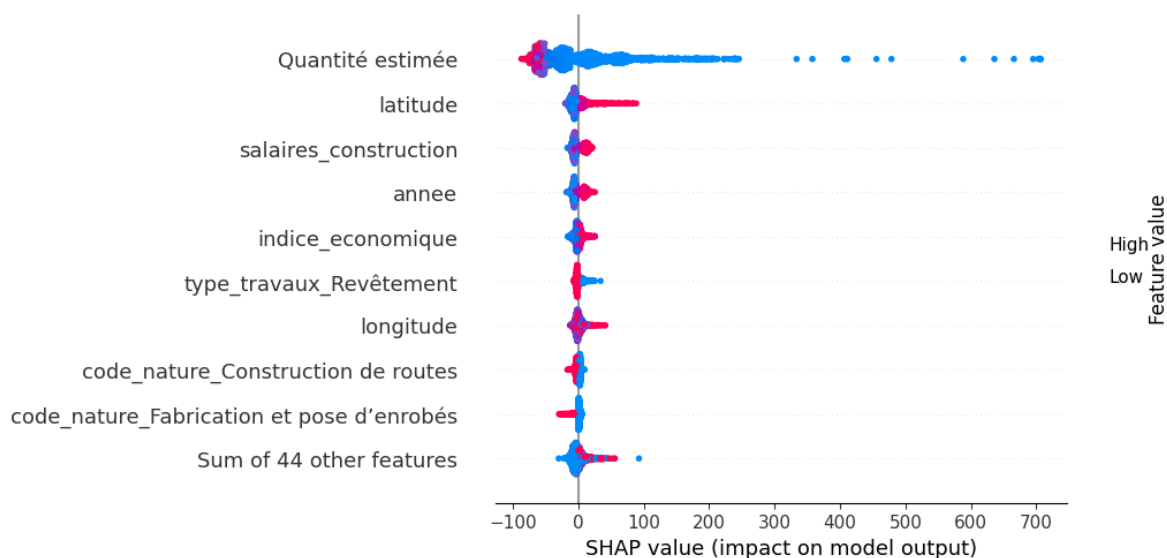


FIGURE B.4 – Analyse de l'impact des variables du modèle CatBoost pour la tâche couche de base, enrobé.

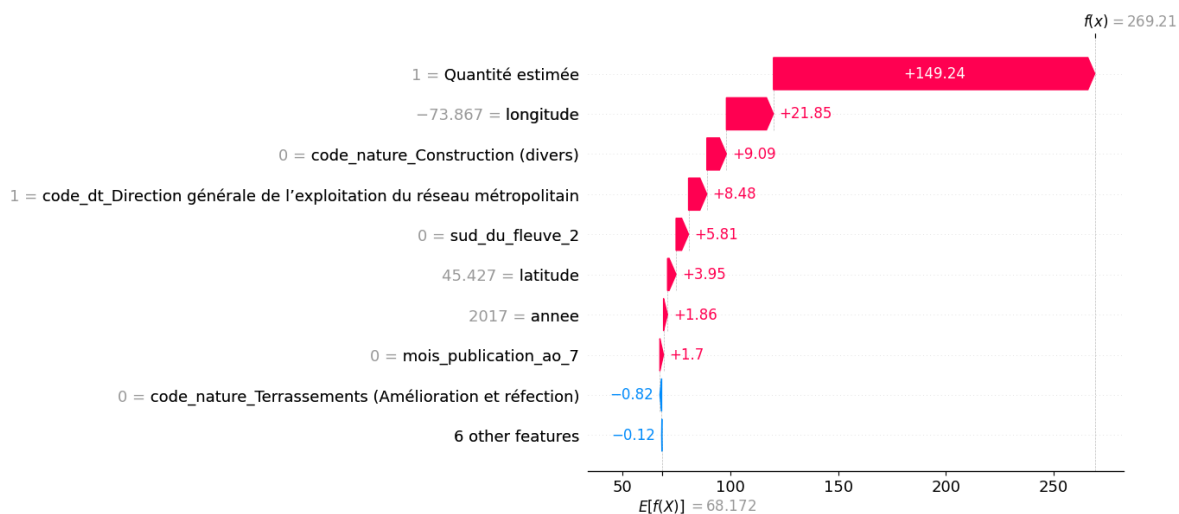


FIGURE B.5 – Analyse de la prédiction du prix unitaire d'une seule soumission du modèle XGBoost pour la tâche déblai de 1^{re} classe.

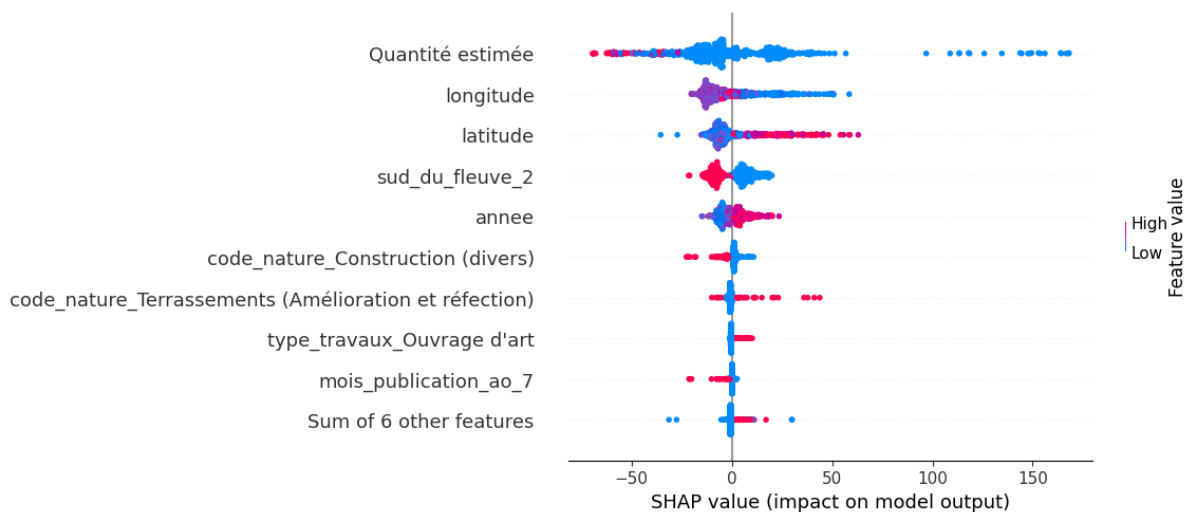


FIGURE B.6 – Analyse de l'impact des variables du modèle XGBoost pour la tâche déblai de 1^{re} classe

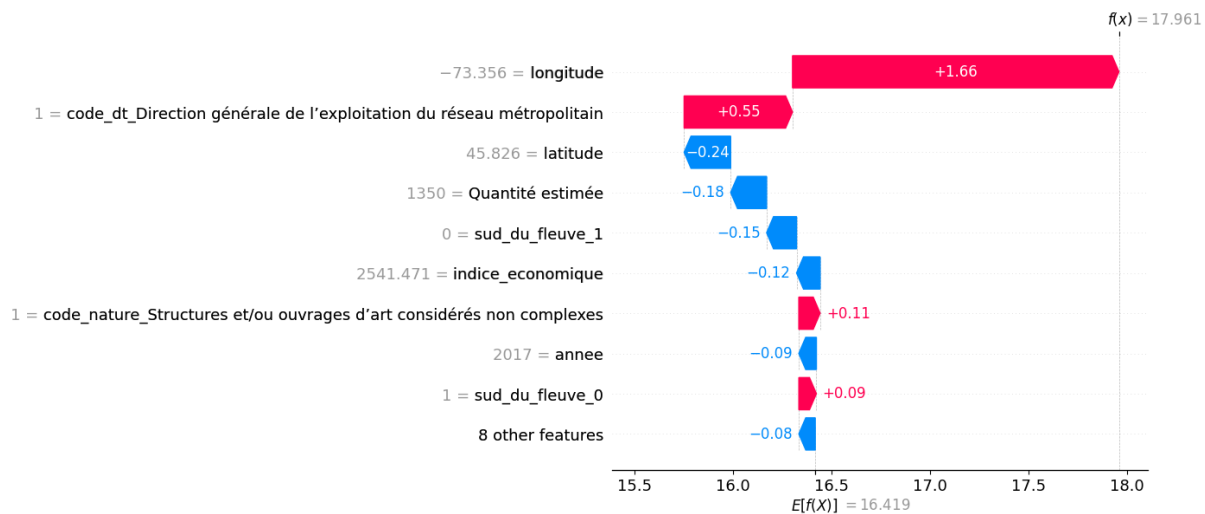


FIGURE B.7 – Analyse de la prédiction du prix unitaire d'une seule soumission du modèle XGBoost pour la tâche déblai de 2^e classe.

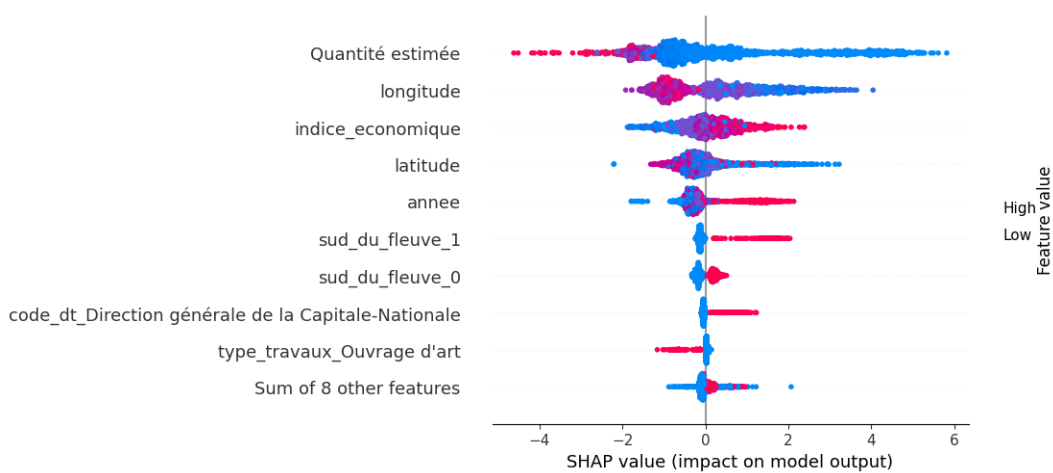


FIGURE B.8 – Analyse de l'impact des variables du modèle XGBoost pour la tâche déblai de 2^e classe.

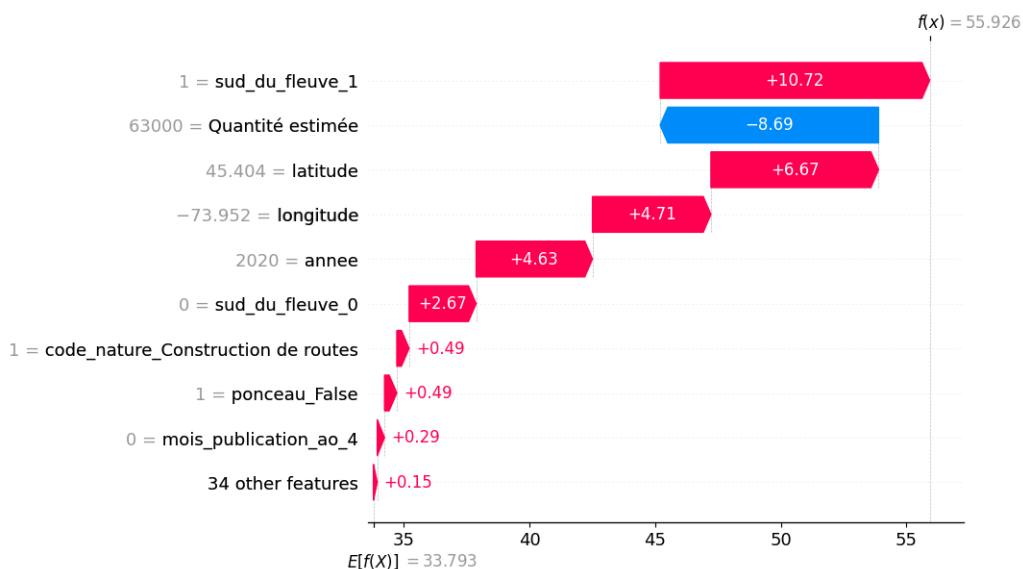


FIGURE B.9 – Analyse de la prédiction du prix unitaire d’une seule soumission du modèle EnsembleVoters pour la tâche sous-fondation de chaussée.

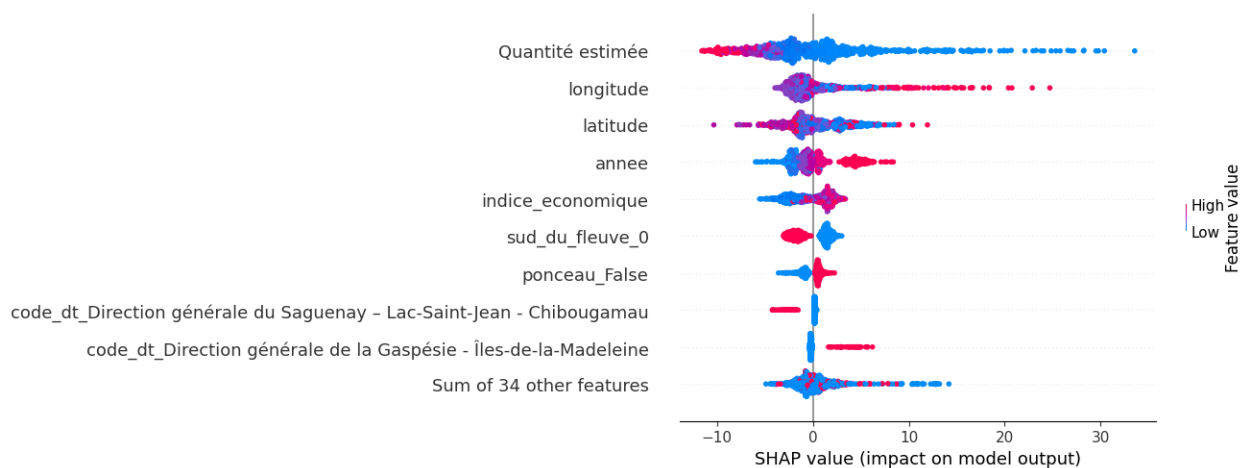


FIGURE B.10 – Analyse de l’impact des variables du modèle EnsembleVoters pour la tâche sous-fondation de chaussée.

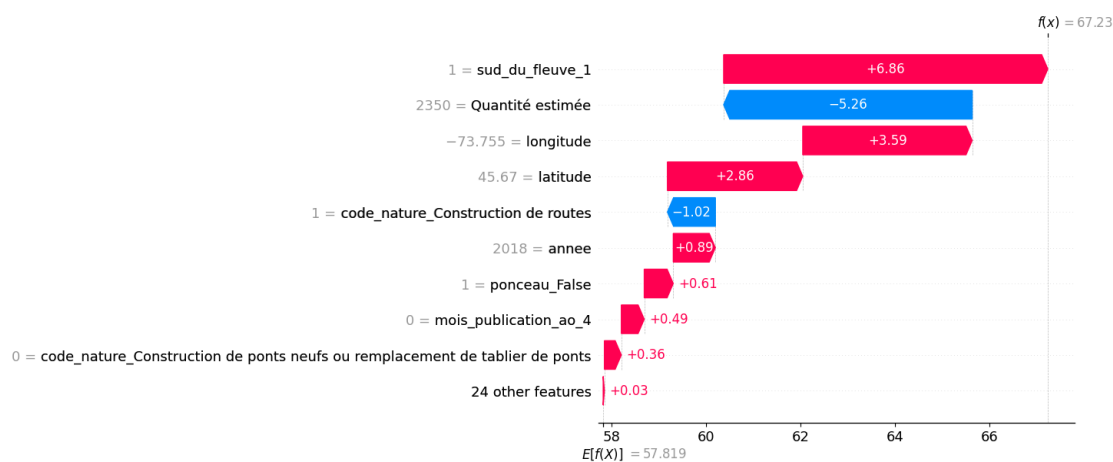


FIGURE B.11 – Analyse de la prédiction du prix unitaire d'une seule soumission du modèle EnsembleVoters pour la tâche fondation de chaussée.

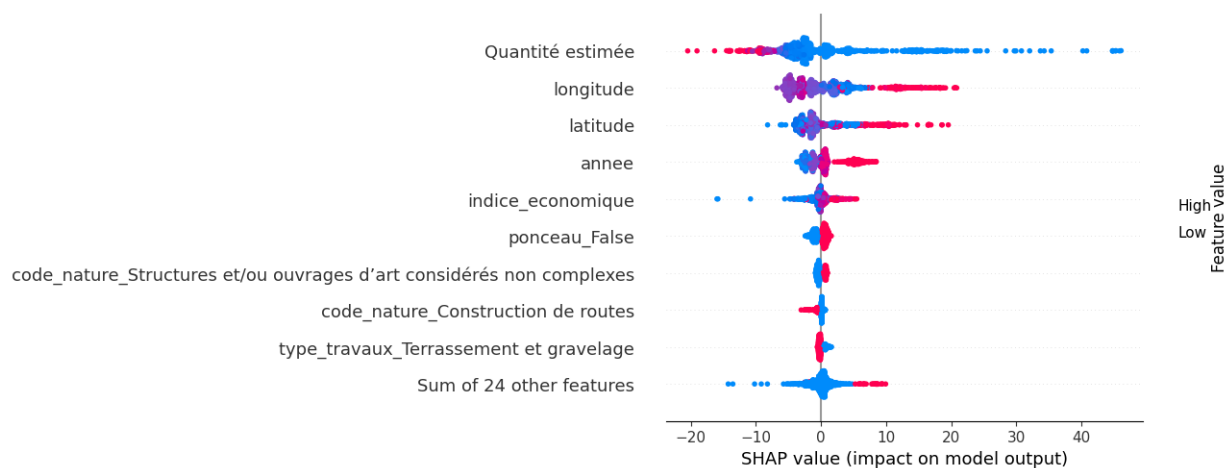


FIGURE B.12 – Analyse de l'impact des variables du modèle EnsembleVoters pour la tâche fondation de chaussée.

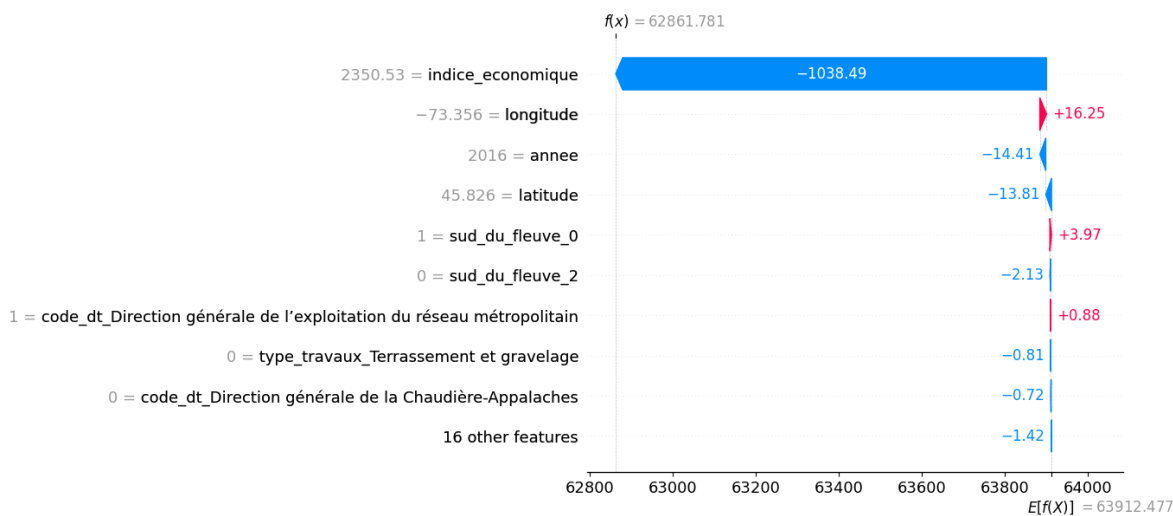


FIGURE B.13 – Analyse de la prédiction du prix unitaire d'une seule soumission du modèle réseau de neurones pour la tâche organisation de chantier.

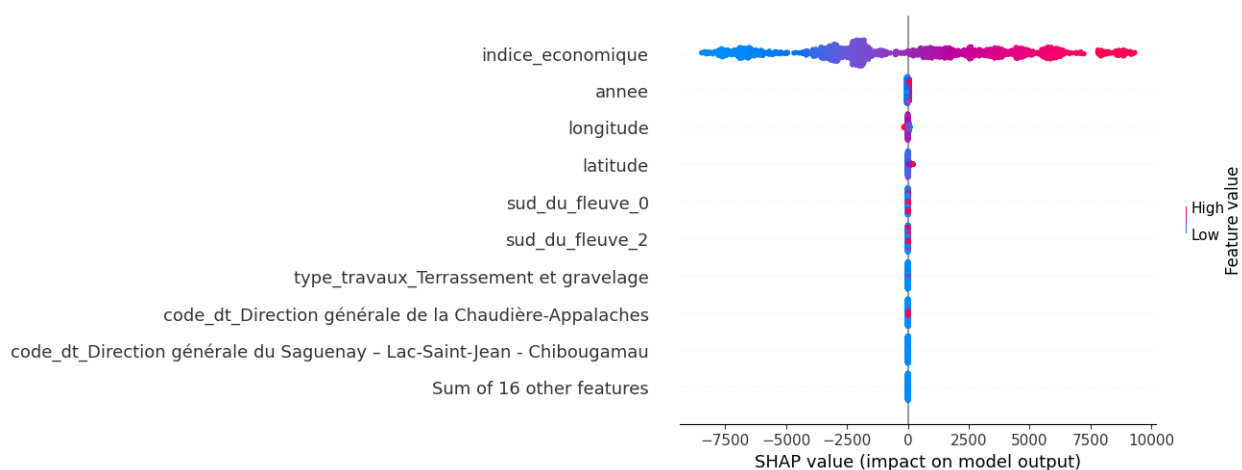


FIGURE B.14 – Analyse de l'impact des variables du modèle réseau de neurones pour la tâche organisation de chantier.