

INTÉGRATION DE L'INTELLIGENCE ARTIFICIELLE (IA) À LA BASE DE DONNÉES BORDEREAU ET DEMANDE DE PAIEMENT (BDP)

Projet R859.1

Présenté par
Rosanne Larocque, étudiante à la maîtrise recherche
en génie informatique

Sous la direction de
Quentin Cappart, M. Sc., Ph. D., professeur adjoint Département de génie
informatique et génie logiciel Polytechnique Montréal

Téléphone : 514 340-4711, poste 7192
Courriels : {quentin.cappart, rosanne.larocque}@polymtl.ca

Réalisé pour le compte du ministère des Transports

Juin 2022

INTÉGRATION DE L'INTELLIGENCE ARTIFICIELLE (IA) À LA BASE DE DONNÉES
BORDEREAU ET DEMANDE DE PAIEMENT (BDP)

INTÉGRATION DE L'INTELLIGENCE ARTIFICIELLE (IA) À LA BASE DE DONNÉES
BORDEREAU ET DEMANDE DE PAIEMENT (BDP)

La présente étude a été réalisée à la demande du ministère des Transports et a été financée par le Ministère.

Les opinions exprimées dans le présent rapport n'engagent que la responsabilité de leurs auteurs et ne reflètent pas nécessairement les positions du ministère des Transports.

Collaborateurs

Quentin Cappart, M. Sc., Ph. D., professeur adjoint Département de génie informatique et génie logiciel Polytechnique Montréal

INTÉGRATION DE L'INTELLIGENCE ARTIFICIELLE (IA) À LA BASE DE DONNÉES
BORDEREAU ET DEMANDE DE PAIEMENT (BDP)

SOMMAIRE

Pour donner suite à des questionnements du Vérificateur général du Québec quant à l'exactitude de l'estimation des coûts des projets, le ministère des Transports s'est mandaté d'améliorer autant la performance que la transparence des estimations. Les estimations actuelles en coûts de projets utilisent la base de données BDP afin de prédire un coût moyen historique. Or, les travaux antérieurs consultés se tournent beaucoup vers l'intelligence artificielle (IA) pour ce genre de tâches. Pour user du plein potentiel de l'IA, il faut généralement une bonne quantité de données sous format numérique afin de les offrir à des modèles comme source d'apprentissage. BDP pourra donc être un excellent point de départ où appliquer l'IA. Contenant de multiples exemples de soumissions antérieures, BDP permettra au modèle d'IA d'apprendre la relation entre diverses informations sur un projet et son coût unitaire. Pour chaque soumission, BDP contient des informations sur le lieu, le type de travaux, les matériaux utilisés, la période des travaux, etc. Il s'agira donc d'implémenter un algorithme d'IA étant capable d'effectuer des prédictions de qualité, tout en justifiant la logique de sa prédiction. Dans la littérature actuelle, il existe généralement une tension entre la performance et l'explicabilité d'un modèle d'IA.

On explore ici cinq modèles d'IA offrant différents équilibres entre ces deux caractéristiques. Les modèles testés sont les réseaux de neurones, les arbres de décision, les forêts aléatoires, l'algorithme XGBoost, ainsi qu'un modèle ensembliste hybride regroupant les quatre modèles précédents. Afin d'améliorer la précision des prédictions, on propose différentes techniques permettant d'améliorer des données. En l'occurrence, il s'agit principalement de traiter les valeurs aberrantes, de mettre les données à l'échelle, d'ajouter des variables externes et d'effectuer une sélection de variables rigoureuse parmi celles disponibles. Pour offrir une manière plus transparente d'expliquer les prédictions, la méthode SHAP est proposée. Elle offre une manière élégante d'expliquer le rôle de chaque estimation en posant une valeur numérique pour quantifier l'impact de chaque variable pour une prédiction donnée.

En tout, huit expériences différentes ont été lancées avec BDP et les cinq modèles. Chaque expérience utilise un code d'ouvrage différent. En moyenne, le meilleur modèle d'IA est XGBoost, mais cela varie d'un code d'ouvrage à l'autre. L'utilisation d'une technique de détection de valeurs aberrantes et de sélection de variables se sont également avérées utiles pour améliorer la précision des prédictions. L'explication donnée par SHAP est également une proposition fort intéressante qui semble concorder avec l'intuition humaine. Pour tous les codes d'ouvrage locaux, l'IA améliore l'estimation dans des proportions variant entre 3,42% et 31,56%. Or, pour un code d'ouvrage plus global (Organisation de chantier), l'estimation humaine surpasse celle artificielle notablement. Il faudrait davantage d'information pour profiter du plein potentiel de l'IA dans ce cas. Bref, l'IA paraît être une alternative intéressante aux outils d'estimation actuels, mais il faudrait explorer davantage avec d'autres codes d'ouvrage. La performance semble même dépasser celle de l'homme pour sept codes d'ouvrage étudiés, tout en offrant une explication des estimations. Cela

INTÉGRATION DE L'INTELLIGENCE ARTIFICIELLE (IA) À LA BASE DE DONNÉES
BORDEREAU ET DEMANDE DE PAIEMENT (BDP)

dit, son application concrète est en réalité un véritable défi logiciel qui sera à analyser en profondeur.

SYNTHÈSE DES CONCLUSIONS ET RECOMMANDATIONS

Les principales conclusions et recommandations s'articulent autour de trois thématiques. La première concerne la bonification des données de BDP en vue de son utilisation en IA. La deuxième vise le développement d'un modèle d'IA des plus précis possible. La troisième et dernière thématique est en lien avec l'explicabilité des modèles d'IA étudiés.

Pour bonifier les données de BDP, on suggère les apports suivants :

- Suppression des soumissions contenant des valeurs nulles
- Conversion des variables catégoriques vers un format numérique compatible avec la plupart des modèles d'IA
- Mise à l'échelle des données lorsque nécessaire
- Ajout de variables externes (indice économique, prix historique du bitume, distance entre l'usine d'enrobé la plus proche et le chantier)
- Utilisation d'un algorithme de sélection du meilleur sous-ensemble de variables (RFE)
- Utilisation de techniques de détection automatique des valeurs aberrantes

Afin de maximiser la précision des prédictions en utilisant l'apprentissage automatique, les recommandations sont les suivantes :

- Prioriser les modèles d'IA ensemblistes (XGBoost, forêts aléatoires, modèles hybrides)
- Répéter l'exploration de divers modèles pour chaque nouveau code d'ouvrage
- Effectuer une sélection rigoureuse des hyperparamètres en utilisant une technique de validation croisée
- Refaire de nouveaux entraînements au fur et à mesure que la quantité de données augmente avec le temps

Finalement, l'utilisation de l'algorithme SHAP est conseillé afin de fournir une explication de l'estimation faite par un modèle d'IA.

TABLE DES MATIÈRES

1. INTRODUCTION	8
2. REVUE DE LITTÉRATURE	10
2.1 INTELLIGENCE ARTIFICIELLE EN ESTIMATION DE COÛT DE PROJETS DE CONSTRUCTION	10
2.2 EXPLICABILITÉ EN APPRENTISSAGE AUTOMATIQUE	12
3. MÉTHODOLOGIE	14
3.1 ENSEMBLES DE DONNÉES	14
3.2 MODÈLES D'IA	17
3.3 MÉTRIQUES, MÉTHODES DE VALIDATION ET D'ÉVALUATION FINALE	19
3.3.1 MÉTRIQUE.....	19
3.3.2 MÉTHODES DE VALIDATION ET D'ÉVALUATION FINALE	20
3.4 PIPELINE D'OPTIMISATION DES MODÈLES	21
3.4.1 PRÉPARATION DES DONNÉES (ÉTAPE 1)	21
3.4.2 SÉLECTION DE LA MISE À L'ÉCHELLE (ÉTAPE 2)	22
3.4.3 SÉLECTION DES VARIABLES (ÉTAPE 3).....	22
3.4.4 TRAITEMENT DES VALEURS ABERRANTES (ÉTAPE 4).....	23
3.4.5 CONFIGURATION DU MODÈLE (ÉTAPE 5).....	24
3.5 MÉTHODES POUR EXPLIQUER LES PRÉDICTIONS	24
4. ANALYSE DES RÉSULTATS	25
4.1 COMPARAISON DES MODÈLES D'IA ÉTUDIÉS	25
4.2 SÉLECTION DE VARIABLES AVEC LA MÉTHODE RFE	26
4.3 DÉTECTION AUTOMATIQUE DES VALEURS ABERRANTES	28
4.4 ANALYSE COMPARATIVE DES TECHNIQUES DE VALIDATION CROISÉE ...	29
4.5 COMPARAISON DE LA PERFORMANCE DE L'IA AVEC CELLE DE L'ESTIMATION HUMAINE	30
5. INTERPRÉTATION DES RÉSULTATS	32
6. MISE EN ŒUVRE ET RETOMBÉES POUR LE MTQ	35
7. CONCLUSION	37
ANNEXE I : HYPERPARAMÈTRES DES MEILLEURS MODÈLES PAR CODE D'OUVRAGE	39

**ANNEXE II : VARIABLES SÉLECTIONNÉES (PAR ORDRE
D'IMPORTANCE) POUR CHAQUE CODE D'OUVRAGE42**

**ANNEXE III : EXPLICATIONS LOCALES ET GLOBALES PAR CODE
D'OUVRAGE.....47**

RÉFÉRENCES BIBLIOGRAPHIQUES58

LISTE DES TABLEAUX

Tableau 1	Les sept codes d'ouvrage étudiés	15
Tableau 2	Les différentes variables utilisées.....	16
Tableau 3	Variables utilisées par code d'ouvrage.....	17
Tableau 4	Comparaison du MAPE sur l'ensemble de test (année 2021) des modèles d'IA par code d'ouvrage	26
Tableau 5	Analyse de l'étape de sélection de variables avec la méthode RFE	27
Tableau 6	Rang d'importance des variables externes dans l'entrée des meilleurs modèles	28
Tableau 7	Comparaison des techniques de détection de valeurs aberrantes	29
Tableau 8	Comparaison des validations croisées aléatoires et temporels	30
Tableau 9	Résumé des meilleures performances des modèles d'IA comparées à celles de l'estimation humaine	31
Tableau 10	Résumé de l'amélioration apportée par les modèles d'IA par code d'ouvrage	31

LISTE DES FIGURES

Figure 1	Séparation de l'ensemble de données dans les quatre phases de validation.	20
Figure 2	Schéma des six étapes du pipeline d'optimisation pour un ensemble de données	21
Figure 3	Comparaison du MAPE provenant de l'estimation humaine avec le MAPE des modèles d'IA.....	32
Figure 4	Analyse de la prédiction du prix unitaire d'une seule soumission selon le modèle XGBoost pour le code d'ouvrage Armature	33
Figure 5	Analyse de l'impact SHAP du modèle XGBoost pour le code d'ouvrage Armature.....	34
Figure 6	Importance des variables explicatives pour le code d'ouvrage armature.....	35

GLOSSAIRE

Ensemble d'entraînement : Dans un problème d'apprentissage automatique, il s'agit des exemples utilisés par le modèle pour lui permettre d'étudier la relation entre les variables explicatives et sa cible pour qu'il puisse éventuellement prédire de nouveaux exemples.

Ensemble de test : Dans un problème d'apprentissage automatique, il s'agit d'exemples mis à part au début qui ne seront jamais utilisés d'aucune manière durant l'entraînement du modèle. Ces exemples servent à évaluer la performance finale d'un modèle sur des exemples qu'il n'a jamais vus. Cela simule le comportement du modèle tel qu'il pourrait être utilisé concrètement en production.

Ensemble de validation : Dans un problème d'apprentissage automatique, il s'agit d'exemples mis à part au début qui serviront à choisir la configuration du modèle utilisé. Ces données ne sont pas utilisées pour l'entraînement du modèle, mais pour évaluer sa performance avec différents paramètres. Cela dit, une fois la configuration optimale choisie, la performance finale retenue est celle calculée à partir de l'ensemble de test, qui lui, n'est jamais intervenu lors de l'optimisation ni lors de l'entraînement du modèle.

Explicabilité : En IA, cela caractérise la capacité d'un algorithme à décrire sa logique et sa prise de décision de manière intelligible à une personne.

Hyperparamètres : Il s'agit d'un ensemble de valeurs statiques attribuées à un modèle d'IA avant son entraînement permettant de préciser ou de modifier certains comportements lors de cet entraînement. Par exemple, on pourrait décider de limiter la profondeur d'un arbre de décision à un certain nombre, pour éviter qu'il ne soit trop spécifique.

Modèles linéaires : « Le terme « modèle linéaire » implique que le modèle est spécifié sous la forme d'une combinaison linéaire d'entités. En fonction des données de formation, le processus d'apprentissage calcule un poids pour chaque entité pour former un modèle pouvant prédire ou estimer la valeur cible. Par exemple, si votre cible est le montant d'assurance qu'un client acquerra, et vos variables l'âge et le revenu, un modèle linéaire simple serait le suivant¹ » :

$$\text{Estimation} = 0.2 + 5 * \text{âge} + 0.0003 * \text{revenu}$$

¹ « Modèles linéaires - Amazon Machine Learning ».

https://docs.aws.amazon.com/fr_fr/machine-learning/latest/dg/linear-models.html (consulté le 24 mai 2022).

INTÉGRATION DE L'INTELLIGENCE ARTIFICIELLE (IA) À LA BASE DE DONNÉES
BORDEREAU ET DEMANDE DE PAIEMENT (BDP)

1. INTRODUCTION

Lors de son rapport de 2017, le Vérificateur général du Québec (VGQ) a questionné l'exactitude de l'estimation des coûts des projets au ministère des transports (MTQ). De ce fait, le MTQ s'est donné comme mission d'améliorer la précision et la transparence de ces estimations. Une première technique mise en place fut l'utilisation du coût moyen historique calculé grâce aux informations recueillies dans la base de données Bordereau et Demande de paiement (BDP). Or, cet outil ne fait qu'une moyenne ce qui rend ardue l'interprétation du prix estimé. De plus, 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 qui peuvent influencer les prix de soumissions.

Un outil plus moderne permettant d'améliorer à la fois l'explicabilité et la performance de cette estimation serait sans doute l'intelligence artificielle (IA). Plus particulièrement, le domaine de l'apprentissage automatique supervisé propose divers algorithmes permettant l'analyse des relations entre les variables explicatives et le prix estimé, pour ensuite permettre des estimations futures des plus précises. L'un des éléments soulevés par le VGQ concerne notamment le manque de documentation permettant de justifier les prix estimés des projets. Cela dit, bien que les modèles d'apprentissage automatique simples se veuillent naturellement explicables, des modèles plus complexes sont parfois plus performants en matière de précision. En effet, l'explicabilité des modèles les plus performants est encore un défi dans la littérature moderne sur l'intelligence artificielle. Il faudra donc trouver un ou plusieurs modèles offrant un équilibre intéressant entre performance et explicabilité.

Étant donné ce contexte, ce rapport présente la solution qui a été conçue et réalisée pour estimer précisément les coûts unitaires d'un projet de construction. Particulièrement, les apports que nous proposons pour bonifier les données de BDP sont de transformer les données brutes provenant en un format compatible avec l'IA, d'ajouter des variables explicatives externes aux soumissions et d'utiliser des algorithmes de détection automatique des données aberrantes. Les modèles proposés pour expérimenter avec l'apprentissage automatique supervisé sont les réseaux de neurones, l'arbre de décision, les forêts aléatoires, XGBoost et un modèle ensembliste hybride contenant les quatre premiers. Pour expliquer les prédictions des divers modèles, l'algorithme SHAP est suggéré. De plus, on propose deux outils simples permettant d'abord de reproduire les résultats, mais également de poursuivre l'exploration en offrant une manière simple d'ajouter d'autres modèles, d'autres variables ou encore d'autres codes d'ouvrage. Ces outils visent à renforcer de bonnes pratiques en IA lors des expériences ultérieures.

Ce rapport présentera d'abord brièvement une revue de littérature (section 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. La section 3 présentera les ensembles de données, les modèles d'IA qui seront étudiés et les métriques utilisées pour quantifier la performance d'un algorithme. La méthodologie complète d'expérimentation y sera également détaillée. La section 4 vise à présenter les différents résultats obtenus avec différentes combinaisons de modèles et de prétraitements, et à comparer les meilleurs résultats avec ceux obtenus par une estimation humaine. La section 5 présentera la manière d'interpréter les prédictions des meilleurs modèles. La section 6 détaille les retombées possibles du projet pour le MTQ. Le rapport se terminera à la section 7, dans laquelle on met en lumière les faits saillants, les limites des méthodes utilisées et les pistes possibles pour la suite du projet.

2. REVUE DE LITTÉRATURE

Le présent projet 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. De ce fait, il serait ardu de les appliquer dans un contexte gouvernemental où on espère le plus grand niveau de transparence. 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 performantes 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], [3]. C'est vers les années 1990 que des méthodes plus proches de celles d'intelligence artificielle moderne ont commencé à être utilisées à cette fin. Par exemple, [4] propose une méthode basée sur le «*Case-based reasoning*» (CBR). Il s'agit d'une méthode visant 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. Également, de plus en plus d'estimateurs se tournent vers les réseaux de neurones (RN) [4], [5], [6]. Bien qu'il y ait eu un engouement évident pour les RN, un article de 2004 observe déjà qu'il s'agit d'une technique peu transparente [7]. Ainsi, leur expérience visait à comparer les performances d'une technique basée sur le CBR, d'une régression linéaire multiple et des RN. 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.

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. En première position, ce sont les techniques dites paramétriques, ce qui inclut entre autres la régression linéaire [8]. 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 de ces infrastructures. C'est notamment le cas de l'Inde [9], de la Grèce [10], des Philippines [11] et même de la Croatie [12]. 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.

Récemment, plusieurs auteurs ont observé la tendance que peuvent avoir les RN à faire du surapprentissage [13]. Il s'agit d'un phénomène observable lorsqu'un modèle d'IA apprend par cœur l'ensemble de données qui lui est fourni pour apprendre. 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 [13], [14], [15]. Par exemple, on pourrait vouloir combiner une régression linéaire et un RN pour une prédiction finale, afin de profiter des avantages respectifs de chacun. En l'occurrence, la régression linéaire tend plus rarement à faire du surapprentissage, mais les RN sont plus performants.

Outre les RN, [16] recense une vingtaine de modèles d'intelligence artificielle ayant été utilisés afin d'estimer les coûts de projets de constructions. Cette étude comparative identifie l'algorithme XGBoost comme étant le plus performant. XGBoost, introduit pour la première fois en 2016 [17], se base d'abord sur les arbres de décision, mais apporte un concept d'ensemble. Au lieu de n'utiliser qu'un seul arbre prédictif, l'idée est d'en utiliser plusieurs qui se complètent mutuellement (*boosting*). Cette idée en fait un estimateur robuste qui s'adapte bien à différents types de données. En effet, bien que les réseaux de neurones surpassent actuellement toutes les méthodes dans des domaines tels la vision par ordinateur ou le traitement automatique de la langue, les algorithmes ensemblistes semblent mieux adaptés pour des données mixtes contenant des variables continues et des variables discrètes. Un autre algorithme un peu moins connu, mais également basé sur le *boosting* semble encore mieux adapté lorsque l'on rencontre beaucoup de données qualitatives. Il s'agit de la technique CatBoost, introduit en 2017 [18]. Elle a notamment surpassé la performance des RN et de l'algorithme XGBoost dans le domaine de la finance [19].

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 prédictif 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 [13], [20]. L'objectif de [13] 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. L'expérimentation compare trois modèles; un linéaire, un RN et un modèle hybride combinant les deux premiers. C'est le modèle hybride qui s'est avéré le plus performant. L'utilisation d'un modèle hybride ne fait que suivre la tendance qu'ont les articles récents à se tourner vers ce genre de technique pour pallier les faiblesses des RN. Toutefois, leur méthode d'évaluation 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 calculé la performance correspond respectivement aux données de la dernière année, des trois dernières années et des cinq dernières années de l'ensemble initial. Bien que cette méthode soit intéressante d'un point de vue analytique, la performance court terme est sans doute la plus appropriée considérant qu'en pratique, on mettrait sans doute le modèle à jour fréquemment, selon les nouvelles données cumulées. L'approche proposée dans [20] suit davantage cette optique. La méthode d'évaluation utilisée est légèrement plus complexe et s'inspire de la technique de validation croisée. C'est d'ailleurs de cette méthode que s'inspire la méthode utilisée dans la présente expérience. Elle sera expliquée et détaillée dans la section méthodologie. Quant à l'expérience en soi, elle est fortement similaire à la présente. Il s'agit d'estimer les coûts d'items en construction d'autoroutes 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. Les résultats obtenus indiquent que les modèles linéaires ont mieux performé que les RN. Il s'agit d'un résultat surprenant considérant les résultats antérieurs dans la littérature, mais tout de même fort intéressant considérant qu'ils font partie de ceux qui ont porté une attention particulière à la qualité des données et à avoir une méthode d'évaluation tenant compte du fait que la distribution des données évolue dans le temps.

En résumé, il n'existe actuellement pas de consensus sur 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, tout comme les autres modèles d'IA linéaires plus classiques. Les modèles hybrides sont également à explorer, bien que la complexité de ce genre de modèle décuple.

2.2 Explicabilité en apprentissage automatique

Les modèles dont nous avons discutés 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 interprétables. En IA, l'interprétabilité (ou explicabilité) caractérise la capacité d'un algorithme à décrire sa logique et sa prise de décision de manière intelligible à une personne.

Il existe des méthodes qui ne fonctionnent qu'avec certains modèles. Par exemple, pour les réseaux de neurones, on peut tenter de copier le comportement d'un réseau avec un modèle plus simple, de sorte à améliorer les performances de ce dernier, tout en rendant les prédictions du réseau plus interprétables. L'équipe de Google Brain a tenté ce principe pour traduire un réseau de neurones en arbre de décision [21]. Cette méthode, nommée distillation, connaît toutefois des limites. En effet, dans cette expérience, l'arbre de décision copiant le RN dépasse les performances de prédictions d'un arbre classique par seulement 2% sur un ensemble de données plutôt simple. On ne réduit donc pas beaucoup la tension entre performance et explicabilité en utilisant la distillation. Elle est d'autant plus limitée par le fait qu'elle ne fonctionne que pour les réseaux de neurones. 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 [22]. 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 *Shapley values*. 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 [23]. Son grand avantage est de pouvoir s'adapter à différents modèles, ce qui 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 [24]. 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 interprétable. 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.

3. MÉTHODOLOGIE

Le processus d'expérimentation s'inscrit en quatre phases subséquentes, décrites par chaque sous-section de la présente. D'abord, il s'agira de l'extraction et l'exploration des données provenant de BDP. La deuxième phase décrit les modèles d'IA qui seront pertinents pour cette expérience. La phase 3, qui est au cœur de l'expérimentation, servira à optimiser la performance de chaque modèle et d'en effectuer une comparaison robuste et objective. On y propose une approche standardisée constituée d'une série d'étapes pouvant d'ailleurs s'appliquer à divers problèmes d'IA. La quatrième et dernière phase permettra de faire une interprétation claire des prédictions des meilleurs modèles choisis dans la phase 3. Il s'agit donc de la phase qui s'intéresse au concept d'explicabilité en IA.

3.1 Ensembles de données

Les ensembles utilisés sont directement extraits de la base de données BDP. Les données choisies ont été recueillies entre les années 2014 et 2021. On utilisera les données de 2014 à 2020 pour l'entraînement et celles de 2021 pour évaluer un modèle de manière finale. Plus de détails sur l'évaluation seront donnés dans la section méthodologie. L'objectif avec cette base de données est de prédire les prix unitaires de différents items liés à un projet. 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. Comme vérité absolue du prix unitaire, on calcule la moyenne du prix unitaire soumis par les trois meilleurs soumissionnaires. Il s'agira de notre prédiction cible.

Une autre particularité importante est qu'on effectuera des prédictions pour huit ensembles de données distincts. Chaque ensemble correspond à ce qu'on appelle un code d'ouvrage. Un code d'ouvrage caractérise en quelque sorte une sous-tâche d'un projet routier du MTQ. Par exemple, pour une chaussée on retrouve généralement des codes d'ouvrages distincts pour la sous-fondation, la fondation et l'enrobé bitumineux. La distribution des données de chaque code d'ouvrage pourrait être différente d'un code d'ouvrage à l'autre. Les variables explicatives pourraient également différer selon le code d'ouvrage. Il importera donc d'entraîner et d'utiliser un modèle indépendant pour chaque code d'ouvrage. Par manque de données pour certains codes d'ouvrage, on étudiera ici huit codes d'ouvrage, d'où les huit ensembles de données. Ces derniers serviront éventuellement à produire huit modèles d'IA. Les codes d'ouvrage étudiés sont énumérés dans le Tableau 1. On y mentionne également le nombre de données disponibles. Le nombre de données indiqué ne contient que les informations de la meilleure soumission pour un même appel d'offres. Les soumissions qui n'ont pas obtenu le premier rang au classement ont été supprimées, pour éviter la redondance des informations pour un même dossier. Les codes correspondent tous à une sous-tâche ou à une sous-section d'un

projet, à l'exception du code Organisation de chantier. Ce dernier est effectivement plus global et un des objectifs de l'expérience est justement de voir si l'IA est aussi efficace pour ce genre de cas.

Tableau 1 Les huit codes d'ouvrage étudiés

Code d'ouvrage	Nom du code	Nombre de données
613205	Couche de surface, enrobé	2596
613200	Couche de base, enrobé	1819
635005	Armature	936
510300	Déblai de 1ère classe	675
510400	Déblai de 2ème classe	1733
600505	Sous-fondation de chaussée	1029
600525	Fondation de chaussée	1191
630100	Organisation de chantier	1667

Les variables explicatives qui seront utilisées pour prédire le prix unitaire sont présentées dans le tableau 2. Certaines des variables sont d'ailleurs un ajout à celles initialement fournies dans BDP. Le tableau 2 identifie ces dernières par la mention « (ajout) ». Certaines variables ont été ajoutées dans l'espoir d'offrir un indicateur de santé économique ou d'inflation aux modèles. C'est notamment le cas des variables « Prix historique du bitume » et « Salaires construction » et « IQ-30 ». 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. Tel que mentionné précédemment, certaines variables sont spécifiques à l'un ou l'autre des codes d'ouvrage, le tableau 3 spécifie les variables utilisées par code d'ouvrage.

INTÉGRATION DE L'INTELLIGENCE ARTIFICIELLE (IA) À LA BASE DE DONNÉES
BORDEREAU ET DEMANDE DE PAIEMENT (BDP)

Tableau 2 Les différentes variables utilisées

Variables	Description
Quantité estimée	Quantité estimée (selon les devis)
Type travaux	Type de travaux (Plantation, engazonnement, écran anti-bruit, 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 : 2 si au sud du fleuve, 1 si Montréal (ni au nord ni au sud) et 0 si 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' <i>asphalte</i> (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 [25]
Distance usine (ajout)	Distance (en min.) entre la localisation d'un chantier et l'usine d'enrobage la plus proche

Tableau 3 Variables utilisées par code d'ouvrage

Variables\Codes d'ouvrage	Couche de surface, enrobé	Couche de base, enrobé	Armature	Déblai de 1ère classe	Déblai de 2ème classe	Sous-fondation de chaussée	Fondation de chaussée	Organisation de chantier
Quantité estimée	x	x	x	x	x	x	x	
Type travaux	x	x	x	x	x	x	x	x
Mois publication ao	x	x	x	x	x	x	x	x
Code dt	x	x	x	x	x	x	x	x
Sud du fleuve	x	x	x	x	x	x	x	x
Latitude et longitude	x	x	x	x	x	x	x	x
Code nature	x	x	x	x	x	x	x	x
Année	x	x	x	x	x	x	x	
Ponceau	x	x	x	x	x	x	x	
Formule enrobe	x	x						
Réparation	x	x	x					
Construction	x	x	x					
Galvanisé			x					
Rivière				x	x			
Revêtement				x	x			
Sécurité				x	x			
Talus					x			
Matériau granulaire						x	x	
Prix du bitume	x	x						
Salaires construction	x	x	x	x	x	x	x	x
IQ-30	x	x	x	x	x	x	x	x
Distance usine	x	x						

On peut observer dans le tableau 3 que tous les codes d'ouvrage utilisent les informations spatio-temporelles, ainsi que les informations de quantité ou de style d'ouvrage.

Initialement, toutes les variables ont été fournies aux divers modèles d'IA. Toutefois, la troisième phase du projet comprend une sélection plus rigoureuse des variables parmi celles disponibles, car il arrive que certaines variables soient si peu pertinentes qu'elles puissent nuire aux prédictions en ajoutant du bruit inutilement dans la distribution des données. De plus, l'utilisation d'un nombre trop élevé de variables augmente les chances de surapprentissage.

3.2 Modèles d'IA

Lors de cette expérience, deux approches sont envisageables. On pourrait notamment partir d'un modèle simple et tenter d'en améliorer la performance au maximum en effectuant un meilleur traitement préalable des données. 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. En effet, il n'est pas impossible en théorie de rendre un arbre de décision plus performant qu'un RN en effectuant un bon prétraitement des données. Ainsi, un des focus de cette expérimentation visera à travailler les données en vue d'améliorer la performance des modèles. Ces techniques seront détaillées un peu plus loin. On étudiera donc les arbres de décision dans cette optique. On naviguera ensuite vers les forêts aléatoires, pour lesquelles il est plus difficile d'expliquer une prédiction. Cela dit, il est facile d'expliquer le fonctionnement de l'algorithme. En effet, il s'agit tout simplement un groupe d'arbres de décision indépendants dont la prédiction finale est la moyenne de chaque prédiction individuelle. Nous avons aussi vu que la littérature contient une méthode permettant de les traduire en un seul arbre unique, au besoin [22].

À l'opposé, les RN ont 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 incontournable en intelligence artificielle à ce jour en matière de performance. Cette méthode fera donc partie de l'expérience.

Un autre modèle étudié qui s'est avéré très performant avec des ensembles de données qui ressemblent aux nôtres est XGBoost. Il est attendu qu'il soit également plus facile à entraîner que les RN, tout en ayant des performances semblables. Le modèle CatBoost, très similaire à XGBoost, avait été reconnu comme ayant des performances semblables à XGBoost. Cela dit, des expériences préalables avec nos données ont révélé que les performances étaient effectivement similaires à celles de XGBoost. Ces essais sont présentés dans les deux rapports d'étape associés au présent projet. Ainsi, pour réduire l'ensemble des modèles testés, il a été décidé de le laisser de côté pour l'expérience finale.

Une dernière famille de modèle qui a démontré d'excellents résultats en expériences similaires est celle des modèles hybrides. Ils semblent en effet avoir permis de limiter le surapprentissage tout en améliorant la performance. Ainsi, les forces des différents modèles d'IA utilisés (Arbre de décision, Forêt aléatoire, XGBoost et RN) seront combinées dans un dernier modèle. Celui-ci effectue donc d'abord quatre entraînements (un pour chaque modèle). Ensuite, lors d'une prédiction, chaque modèle effectue une prédiction du juste prix. Pour la prédiction finale, c'est la médiane des quatre prix prédits qui sera retenue.

En somme, l'expérience comparera cinq modèles ayant des niveaux de complexité différents. Cette variété permettra donc de mettre en évidence les différents équilibres possibles entre facilité et temps d'entraînement, performance et transparence.

3.3 Métriques, méthodes de validation et d'évaluation finale

3.3.1 Métrique

Afin d'évaluer la performance des modèles, on utilisera le pourcentage d'erreur absolue moyen (MAPE). Il s'agit d'une métrique communément utilisée dans les expériences similaires et elle est facile à interpréter [9], [12], [14], [15]. Il se calcule de la manière suivante :

$$MAPE = \frac{1}{n} \sum_{i \in n} \frac{abs(\hat{y}_i - y_i)}{y_i} * 100\%$$

Où y_i est le prix moyen des trois meilleures soumissions (notre objectif), \hat{y}_i est la prédiction du modèle et n correspond au nombre de données dans l'ensemble utilisé pour faire le calcul. On utilisera cette même métrique pour évaluer l'estimation humaine, qui servira de comparatif pour bien évaluer la performance des modèles.

3.3.2 Méthodes de validation et d'évaluation finale

Dans le monde de l'IA, on nomme méthode de validation ce qui sert à comparer différentes configurations d'un même modèle en vue d'en sélectionner la plus performante. Une manière possible est de séparer l'ensemble d'entraînement initiale en deux parties. La première est utilisée pour entraîner le modèle et la deuxième est utilisée pour évaluer ce dernier avec cette configuration. On dit du dernier qu'il est l'ensemble de validation. On répète cette procédure d'entraînement et de validation avec différentes configurations. On choisit ensuite la configuration ayant permis d'avoir le meilleur score sur l'ensemble de validation. L'idée étant d'éviter de biaiser le choix de configuration en la choisissant selon les scores sur l'ensemble de test. On veut éviter toute forme de surapprentissage de l'ensemble de test. Cet ensemble doit être conservé pour évaluer la dernière version du modèle de manière objective, sans qu'il soit impliqué dans aucune phase de l'entraînement du modèle.

La méthode qu'on utilisera se base sur la méthode décrite précédemment, combinée avec la méthode utilisée par [20]. On procèdera à une validation en quatre temps. Tel qu'illustré à la figure 1, 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.

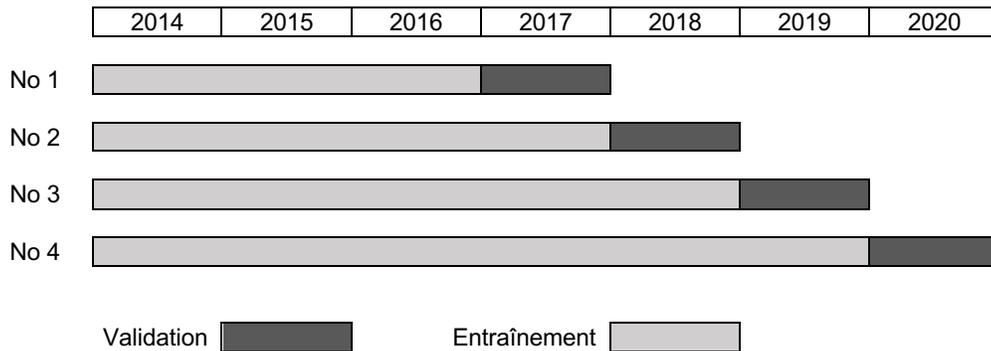


Figure 1 Séparation de l'ensemble de données dans les quatre phases de validation.

Remarquez que l'année 2021 ne figure pas 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. Il est donc important que cette année ne soit pas impliquée d'aucune manière dans le processus de

configuration du modèle. Ainsi, cette évaluation finale se fera avec les données de 2014 à 2020 pour l'entraînement et l'année 2021 servira à calculer le MAPE final du modèle.

3.4 Pipeline d'optimisation des modèles

Une grande partie de ce projet vise à présenter une manière standardisée d'évaluer et d'entraîner divers modèles d'IA. On propose donc un pipeline d'exécution, supporté par un outil en ligne de commandes² permettant de le reproduire avec divers modèles et divers ensembles de données. Il comporte six grandes étapes, qui apparaissent à la figure 1.

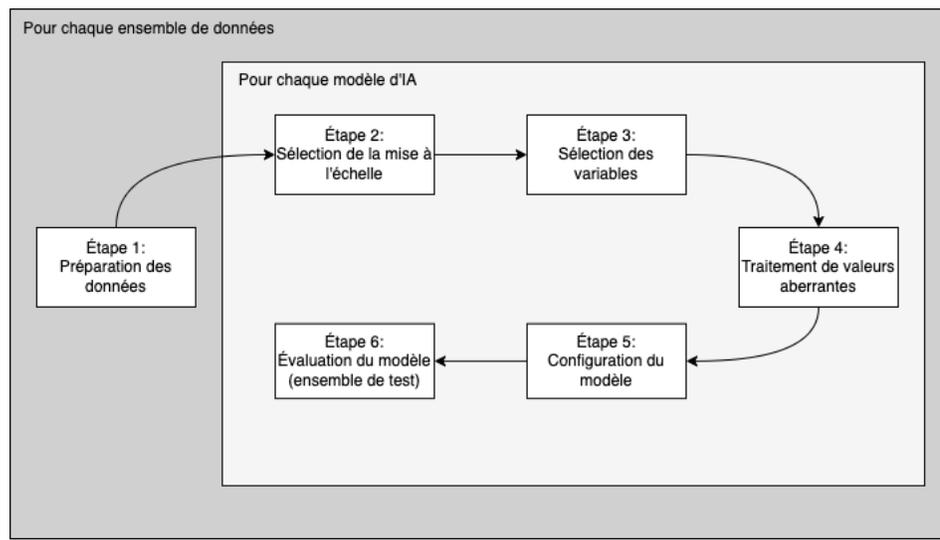


Figure 2 Schéma des six étapes du pipeline d'optimisation pour un ensemble de données

Les différentes étapes seront détaillées dans les sous-sections du même nom, à l'exception de l'étape 6, qui a été expliquée à la section 3.3.2.

3.4.1 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, il manquait entre 3% et 5% des données, dépendamment des codes d'ouvrage. 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.

² <https://github.com/rolare/Tunai>

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.

3.4.2 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 de sorte 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.

Toutefois, un seul entraînement d'un RN peut prendre entre 30 minutes et une heure dans le cadre de cette expérience. Pour cette étape, chaque modèle passerait par le processus de validation complet (4 entraînements par validation) trois fois pour cette étape. 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. Il a donc été préféré d'utiliser un modèle alternatif aux RN pour sélectionner la mise à l'échelle. XGBoost fut choisi pour ce faire. Le modèle EnsembleVoters 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.

3.4.3 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 XGBoost y sont généralement moins sensibles, mais les arbres de régression peuvent grandement bénéficier d'une pré-sélection 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 se nommant « Recursive Feature Elimination » (RFE) [26]. Le principe général est de commencer avec l'ensemble complet des variables disponibles, de faire un entraînement et d'en évaluer la performance. On procède au calcul de l'importance de chaque variable. On identifie la variable possédant la moindre importance, puis on la retire. On répète finalement les premières étapes avec cet ensemble réduit de variable. Le processus itératif s'arrête lorsqu'il ne reste plus aucune variable à enlever. Il y aura donc autant d'entraînement et d'évaluation qu'il y a de variables initialement. L'idée est finalement de sélectionner l'ensemble de variables qui a obtenu la meilleure performance selon le processus de validation.

Dépendamment des codes d'ouvrage, chaque ensemble contient environ 60 variables explicatives. Il y aurait donc 240 entraînements pour chaque modèle et chaque code d'ouvrage (60 variables * 4 entraînements par validation). Tel que pour la sélection de la mise à l'échelle, c'est le modèle XGBoost qui sera utilisé pour la sélection des variables des RN et du modèle EnsembleVoters pour éviter une explosion des coûts computationnels.

L'ensemble de variables ayant rapporté le meilleur score de validation sera conservé pour la suite du pipeline.

3.4.4 Traitement des valeurs aberrantes (étape 4)

Une manipulation qui peut s'avérer très efficace pour améliorer la performance est le traitement des valeurs aberrantes. En effet, elle peut permettre de contrôler l'uniformité des données. Cela dit, le défi est de savoir si une valeur extrême l'est par erreur ou si elle permet réellement d'illustrer une relation qui survient plus rarement. Étant donné qu'il est difficile d'en décider à l'œil, 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. L'implémentation a été fournie directement de la librairie python Scikit-learn [27].

IsolationForest – 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 anomalies [28].

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 [29].

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 rapprocher 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 [29].

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

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

3.4.5 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, toutes les combinaisons possibles de ces listes de paramètres formeront une configuration. Ensuite, chaque configuration passera au travers du processus de validation, puis on conservera la meilleure pour l'évaluation finale.

3.5 Méthodes pour expliquer les prédictions

Étant donné que plusieurs modèles testés ne seront pas nativement explicables, il importe de choisir une méthode qui permettra de pallier cet inconvénient. Étant donné la variété de nos modèles complexes, il serait plus approprié d'utiliser une technique pouvant s'adapter à différentes techniques d'IA. Autrement, l'analyse des résultats serait plus complexe et une comparaison inter-modèles concernant l'interprétation ne serait pas bien uniformisée. C'est pour cette raison que les méthodes agnostiques au modèle ont été ciblées en priorité. Plus particulièrement, on s'intéressera ici à la méthode SHAP, qui semble s'être approchée le plus d'une explication qu'un expert aurait pu apporter.

4. ANALYSE DES RÉSULTATS

Les résultats de l'expérience seront analysés séparément selon chaque phase de l'expérience. Ainsi, cette section présentera d'abord une analyse comparative de la performance des modèles. On étudiera ensuite l'apport de l'algorithme RFE, permettant d'étudier du même coup l'impact des variables externes ajoutées. Subséquemment, ce sera l'apport des algorithmes de détection automatique des valeurs aberrantes. La technique de validation pour l'optimisation des modèles sera aussi étudiée. La mise à l'échelle ne semble pas avoir fait une différence notable dans l'optimisation des modèles. En effet, Les modèles XGBoost et RandomForest sont insensibles à la mise à l'échelle étant basés sur des arbres de régression. Les résultats de comparaisons des différentes mises à l'échelle ne seront donc pas détaillés dans ce rapport afin de l'alléger. Les métriques des différentes combinaisons d'hyperparamètres ne seront également pas discutées puisque c'est plutôt technique et ce n'est pas le focus de la présente expérience. Cela dit, pour des fins de reproductibilité, vous trouverez en annexe les hyperparamètres choisis des meilleurs modèles pour chaque code d'ouvrage. L'analyse des résultats terminera donc avec un comparatif de la performance des meilleurs modèles pour chaque code d'ouvrage avec l'estimation humaine.

4.1 Comparaison des modèles d'IA étudiés

L'entraînement et l'optimisation des cinq modèles a donc été effectué pour chacun des codes d'ouvrage, de manière indépendante. Le tableau 4 contient le MAPE obtenu avec la meilleure version de chaque modèle, pour chaque code d'ouvrage. Il s'agit du résultat obtenu sur l'ensemble de test (année 2021), en utilisant le total des données restantes pour l'entraînement du modèle (années 2014 à 2020). Pour chaque code, le meilleur modèle est mis en évidence par caractère gras et fond ombragé. Les résultats du tableau 4 semblent concorder avec ceux de [16] concernant les données en notre genre. En effet, l'algorithme XGBoost a majoritairement été le plus performant, dans quatre des huit codes d'ouvrage testés. Pour les quatre codes d'ouvrage restants, les résultats sont plutôt variés. Chacun des autres modèles a su se tailler au moins une place au premier rang à son tour, à l'exception du modèle d'arbre de décision classique. Toutefois, ce modèle classique a tout de même surpassé les performances des RN dans la plupart des cas. Ce résultat est plutôt surprenant et s'explique sans doute par le fait que la recherche des hyperparamètres et de la configuration a été faite de manière bien moins rigoureuse qu'avec les autres modèles, vu le coût computationnel élevé de son entraînement. Cela met en lumière le désavantage d'utiliser les RN. Ils sont plus difficiles à entraîner et à optimiser. De plus, dans notre cas il s'agit globalement du modèle le moins performant. Il n'y a donc aucune raison évidente d'aller vers ce genre de modèle pour notre problème.

Tableau 4 Comparaison du MAPE sur l'ensemble de test (année 2021) des modèles d'IA par code d'ouvrage

Code d'ouvrage	DecisionTree	RandomForest	XGBoost	RN	EnsembleVoters
Couche de surface, enrobé	22,37%	22,64%	22,60%	30,91%	20,65%
Couche de base, enrobé	25,68%	19,62%	20,87%	40,59%	20,91%
Armature	32,10%	26,69%	30,76%	40,98%	31,88%
Déblai de 1ère classe	77,10%	60,53%	52,20%	339,58%	53,06%
Déblai de 2ème classe	44,63%	39,38%	37,04%	39,64%	38,65%
Sous-fondation de chaussée	26,00%	21,02%	20,37%	55,07%	20,40%
Fondation de chaussée	20,36%	17,76%	17,15%	47,89%	17,39%
Organisation de chantier	103,23%	113,82%	83,34%	82,15%	99,15%

4.2 Sélection de variables avec la méthode RFE

Le tableau 5 présente l'apport qu'a eu la sélection de variables sur la performance des modèles. Bien que la sélection fût faite sur tous les modèles pour chaque code d'ouvrage, on s'intéresse ici seulement au meilleur modèle de chaque code. Le tableau 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 1 pour plus de détails). L'annexe II détaille les variables sélectionnées pour chaque code.

Dans tous les cas, l'algorithme RFE a permis une amélioration de la performance de notre modèle. L'amélioration la plus notable est pour le code d'ouvrage déblai de 1^{ère} 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 19 et 57 variables. Rappelons que l'utilisation du «One hot encoding» ajoutait une variable pour chaque valeur trouvée, ce qui explique le nombre élevé de variables. 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 5 Analyse de l'étape de sélection de variables avec la méthode RFE

Code d'ouvrage	Meilleur modèle	Nb de variables retirées par RFE / variables totales	MAPE de validation	
			Application de RFE	Toutes les variables
Couche de surface, enrobé	EnsembleVoters*	30/74	23,8%	24,0%
Couche de base, enrobé	RandomForest	18/71	23,0%	23,4%
Armature	RandomForest	40/60	47,2%	48,0%
Déblai de 1ère classe	XGBoost	47/61	87,4%	113,0%
Déblai de 2ème classe	XGBoost	50/67	52,6%	55,0%
Sous-fondation de chaussée	XGBoost	19/62	23,3%	24,0%
Fondation de chaussée	XGBoost	31/64	19,1%	20,0%
Organisation de chantier	NeuralNetwork*	42/67	521,5%	553,0%

*Tel que mentionné en méthodologie, pour les modèles NeuralNetwork et EnsembleVoters, les résultats affichés sont ceux obtenus avec XGBoost.

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 6 présente donc le rang d'importance de chacune d'entre elles pour chaque code d'ouvrage. 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 codes d'enrobé. Une cellule contenant « --- » indique que cette variable ne fut pas sélectionnée dans le meilleur ensemble choisi par RFE.

Le tableau 6 met en évidence que la variable « Indice économique » a eu un apport notable pour sept des huit codes d'ouvrage. Le seul code ne l'ayant pas sélectionnée est le code Déblai de 1^{ère} classe, code 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 ce code 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 codes d'enrobé. Le prix historique du bitume est la plus influente des deux variables peu importe le code. 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 6 Rang d'importance des variables externes fournies aux meilleurs modèles

Code d'ouvrage	Distance usine	Prix historique du bitume	Indice économique	Salaires construction
Couche de surface, enrobé	7	5	6	---
Couche de base, enrobé	8	5	7	6
Armature	Non disponibles pour ces codes d'ouvrage		6	5
Déblai de 1ère classe			---	---
Déblai de 2ème classe			5	---
Sous-fondation de chaussée			5	---
Fondation de chaussée			5	---
Organisation de chantier			4	---

4.3 Détection automatique des valeurs aberrantes

De manière similaire au tableau 5, le tableau 7 étale les MAPE moyens de validation obtenus selon les différentes techniques de détection de valeurs aberrantes utilisées. Pour sept des huit codes d'ouvrage étudiés, l'utilisation d'un algorithme de détection de valeurs aberrantes a permis d'améliorer la performance des modèles. Les améliorations les plus notables peuvent être observées pour les codes d'ouvrage déblai de 1^{ère} classe, armature et organisation de chantier. L'amélioration relative est respectivement de 15%, 36% et 74%. Pour les deux premiers, cela suggère une présence de valeurs extrêmes qui nuisent à l'apprentissage des modèles. Pour le dernier, il s'agit d'un code d'ouvrage global. Il y a donc moins de variables fournies aux modèles pour ce code. 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 cela 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.

Tableau 7 Comparaison des techniques de détection de valeurs aberrantes

Code d'ouvrage	Meilleur modèle	OCSVM	LOF	IsolationForest	Aucun
Couche de surface, enrobé	EnsembleVoters*	32,6%	23,6%	23,5%	23,8%
Couche de base, enrobé	RandomForest	31,7%	24,0%	23,5%	23,4%
Armature	RandomForest	33,4%	31,9%	33,4%	50,2%
Déblai de 1ère classe	XGBoost	85,9%	79,7%	88,7%	94,1%
Déblai de 2ème classe	XGBoost	42,1%	43,1%	45,9%	53,7%
Sous-fondation de chaussée	XGBoost	28,5%	23,6%	24,1%	24,3%
Fondation de chaussée	XGBoost	20,2%	18,8%	20,0%	20,1%
Organisation de chantier	NeuralNetwork*	171,3%	247,8%	269,2%	652,5%

*Tel que mentionné en méthodologie, pour les modèles NeuralNetwork et EnsembleVoters, les résultats affichés sont ceux obtenus avec XGBoost.

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 différerait 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 8 vise à comparer la performance des meilleurs modèles pour chaque code d'ouvrage, selon les deux méthodes de séparation. La performance affichée est celle calculée sur l'ensemble de test (non utilisé lors de la validation ou des entraînements). Il semble y avoir une légère tendance favorisant la méthode de séparation temporelle. Cela dit, la différence absolue entre les performances moyennes des deux méthodes est de seulement 0,48%. 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.

Tableau 8 Comparaison des validations croisées aléatoires et temporels

Code d'ouvrage	Temporel		Aléatoire	
	Modèle	MAPE	Modèle	MAPE
Couche de surface, enrobé	EnsembleVoters	20,65%	EnsembleVoters	23,44%
Couche de base, enrobé	RandomForest	19,62%	XGBoost	18,99%
Armature	RandomForest	26,69%	RandomForest	28,55%
Déblai de 1ère classe	XGBoost	52,20%	XGBoost	52,96%
Déblai de 2ème classe	XGBoost	37,04%	EnsembleVoters	35,06%
Sous-fondation de chaussée	XGBoost	20,37%	XGBoost	20,18%
Fondation de chaussée	XGBoost	17,15%	XGBoost	17,30%
Organisation de chantier	NeuralNetwork	82,15%	XGBoost	83,19%
Moyenne	---	34,48%	---	34,96%

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

Le tableau 9 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 codes d'ouvrage, l'IA effectue une erreur moindre que celle de l'estimation humaine. Le seul code d'ouvrage pour lequel l'estimation surpasse l'IA, et ce, de manière significative, est le code organisation de chantier. Ce code était en effet plutôt particulier puisque contrairement aux autres, il s'agit d'un code d'ouvrage plus global. Les mêmes variables de base ont été fournies pour ce code (voir tableau 3). 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 ce code, puisque ce dernier a accès à plus d'informations que ce qui est fourni au modèle. De telles incohérences n'apparaissent pas pour les autres codes.

Tableau 9 Résumé des meilleures performances des modèles d'IA comparées à celles de l'estimation humaine

Code d'ouvrage	Modèle	MAPE du modèle	MAPE de l'estimation humaine
Couche de surface, enrobé	EnsembleVoters	20,65%	22,63%
Couche de base, enrobé	RandomForest	19,62%	21,47%
Armature	RandomForest	26,69%	39,00%
Déblai de 1ère classe	XGBoost	52,20%	64,79%
Déblai de 2ème classe	XGBoost	37,04%	38,35%
Sous-fondation de chaussée	XGBoost	20,37%	28,25%
Fondation de chaussée	XGBoost	17,15%	22,08%
Organisation de chantier	NeuralNetwork	82,15%	43,35%

Pour mieux apprécier cette amélioration, le tableau 10 permet de montrer que pour les codes d'ouvrages étudiés l'IA réduit l'erreur humaine d'une proportion variante entre 3,42% et 31,56%, en excluant le code d'ouvrage organisation de chantier. Toutefois, pour les codes d'ouvrage d'enrobé et de déblais de 2^{ème} classe, l'amélioration n'est pas significative. La meilleure amélioration est de près de 32%, pour le code d'ouvrage armature. La figure 3 illustre également une comparaison entre le MAPE de l'humain et celui de l'IA.

Tableau 10 Résumé de l'amélioration apportée par les modèles d'IA par code d'ouvrage

Code d'ouvrage	% d'amélioration
Couche de surface, enrobé	8,75%
Couche de base, enrobé	8,62%
Armature	31,56%
Déblai de 1ère classe	19,43%
Déblai de 2ème classe	3,42%
Sous-fondation de chaussée	27,89%
Fondation de chaussée	22,33%
Organisation de chantier	-89,50%

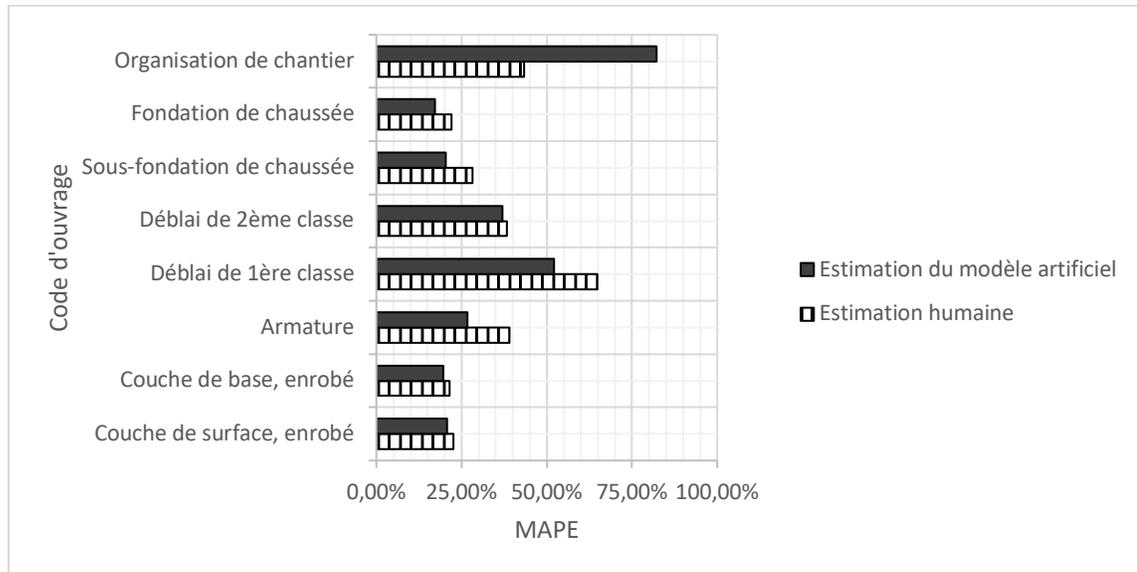


Figure 3 Comparaison du MAPE provenant de l'estimation humaine avec le MAPE des modèles d'IA

En somme, l'expérience démontre que pour sept des huit codes d'ouvrage étudiés, l'IA apporte une amélioration aux estimations humaines. Or, pour le code d'ouvrage 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 réseaux de neurones, soit la difficulté d'entraînement et d'optimisation. À l'inverse, les prouesses du modèle XGBoost ont pu être confirmées, étant globalement le meilleur modèle ayant été étudié. De plus, nous avons pu voir qu'une sélection de variable rigoureuse et un bon traitement des valeurs aberrantes permet d'améliorer la qualité des prédictions de l'IA.

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. La figure 4 présente un exemple d'interprétation d'une soumission aléatoire pour le code d'ouvrage 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 4, 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 prédit par le modèle, 6,905\$ (haut de la figure).

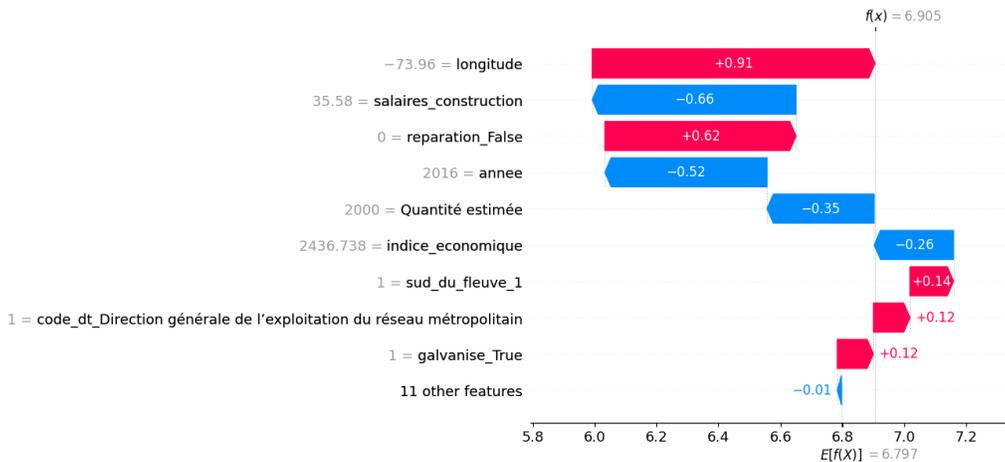


Figure 4 Analyse de la prédiction du prix unitaire d'une seule soumission selon le modèle XGBoost pour le code d'ouvrage Armature

Afin d'alléger ce rapport, on ne considèrera qu'un seul exemple d'explication d'une prédiction locale. Un exemple d'explication locale pour chaque code d'ouvrage 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 en présente un exemple pour le code armature. Un exemple pour chaque code se trouve en annexes.

Si nous prenons la figure 5 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 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. 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 tout simplement la description du dossier

contenait des mots liés au thème de la réparation. 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 ce code. 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 ce code d'ouvrage. 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.

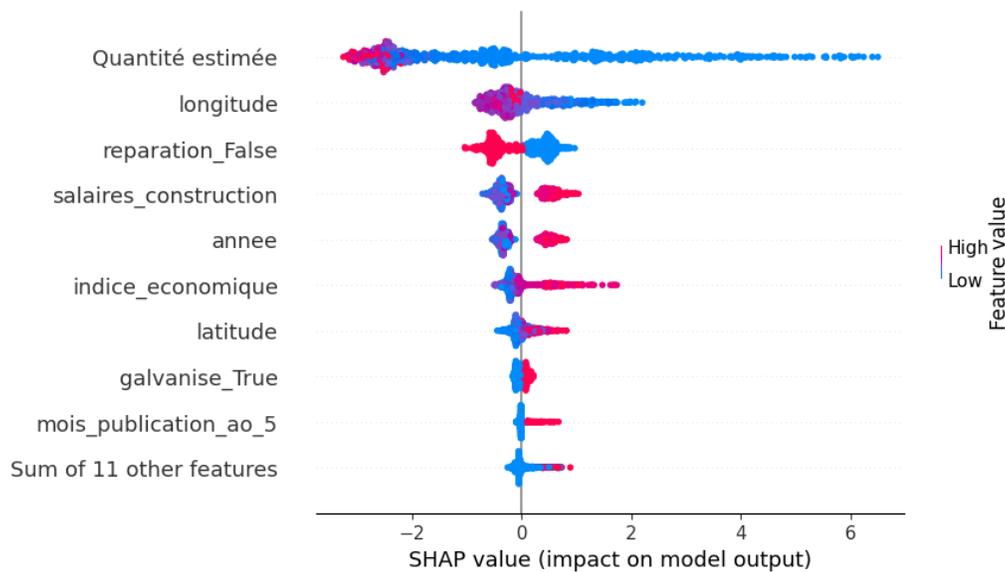


Figure 5 Analyse de l'impact SHAP du modèle XGBoost pour le code d'ouvrage Armature

Les exemples vus aux figures 4 et 5 sont pour le modèle XGBoost, mais le graphique serait similaire peu importe le modèle ayant obtenu la meilleure performance. On voit ainsi la puissance des techniques agnostiques au modèle.

Une autre manière d'observer l'importance des variables de manière globale est d'observer leur pourcentage d'impact sur les prédictions. Pour ce faire, on peut calculer la valeur SHAP moyenne pour l'ensemble des prédictions de nos données, et normaliser le tout de sorte à obtenir un pourcentage permettant de comparer les variables explicatives. On peut ensuite présenter ces résultats dans un histogramme. La figure 6 en montre un exemple avec les variables du code d'ouvrage armature. Les variables ayant une importance nulle ont été retirées de la figure. Les résultats sont similaires à ceux de la figure 5, en y ajoutant une notion de quantité. En effet, la quantité et la longitude sont toujours les deux variables les plus importantes pour ce code. Il est toutefois intéressant de noter que la quantité est responsable de près de la moitié des prédictions finales, avec un impact moyen de 45.03%. D'autres figures semblables sont présentées pour chaque code en annexes.

INTÉGRATION DE L'INTELLIGENCE ARTIFICIELLE (IA) À LA BASE DE DONNÉES BORDEREAU ET DEMANDE DE PAIEMENT (BDP)

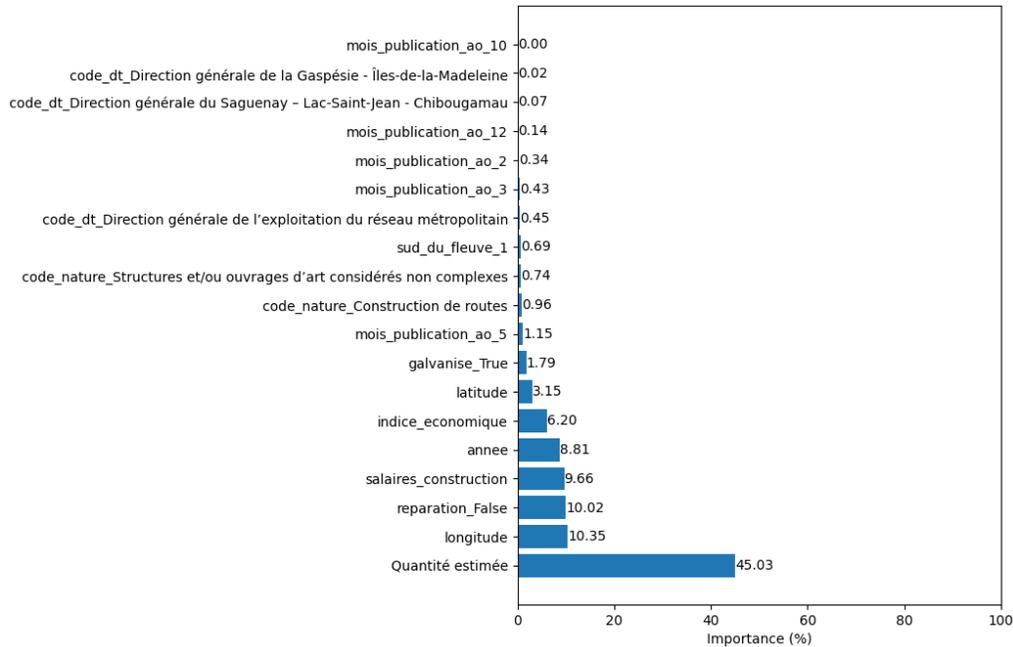


Figure 6 Importance des variables explicatives pour le code d'ouvrage armature

6. MISE EN ŒUVRE ET RETOMBÉES POUR LE MTQ

D'abord, le projet démontre le potentiel de l'IA d'améliorer l'estimation humaine. Ultimement, cette estimation plus précise pourrait compléter l'estimation fournie par le système actuel et faire office de point de référence lorsqu'un appel d'offres est lancé. De ce fait, on augmente les chances d'obtenir des soumissions proposant des prix qui sont justes. Cela mène éventuellement à une meilleure planification au niveau des budgets. En plus d'améliorer l'estimation humaine, les algorithmes de la méthode SHAP permettent de fournir une explication standardisée de chaque estimation pour une soumission. On pourrait donc facilement ajouter cette explication dans les archives d'un projet et la faire modifier par un expert en cas de besoin. Cela enrichirait le dossier et le rendrait plus transparent par rapport au VGQ. De plus, l'explication permet d'identifier les facteurs augmentant le prix de manière significative et de possiblement utiliser ces informations afin d'optimiser le coût des projets ou encore d'analyser plus précisément les coûts annuels de divers projets.

Aussi, les résultats de ce projet amènent l'importance d'user de bonnes pratiques en IA pour tout son potentiel. En l'occurrence, diverses techniques ont été proposées et conçues concernant l'amélioration de la qualité des données. Nous suggérons notamment la mise à l'échelle des données lorsque nécessaire, le traitement des valeurs aberrantes, l'ajout des variables externes bonifiant l'information contenue dans BDP et l'application de l'algorithme RFE pour la sélection de variables. Ces techniques implémentées pourront d'ailleurs directement être réutilisée par le MTQ lors d'investigations ultérieures ou même lors de son application concrète. En effet, deux outils simples (lignes de

commande) sont joints au présent projet. Le premier outil³ fournit un guide complet débutant avec l'extraction des données de BDP, et transforme les données brutes en un format consommable par les modèles d'IA. L'outil est facilement configurable pour qu'on ajoute ou qu'on enlève des variables explicatives. Ainsi, diverses expériences supplémentaires pourront y être faites très facilement. Il sera également simple d'ajouter d'autres codes d'ouvrage sans réécrire du code supplémentaire. Cet outil pourra être utilisé par d'autres chercheurs. Une certaine familiarité avec un terminal est recommandée pour l'utilisation. Un deuxième outil⁴ du même genre est également fourni. Ce deuxième outil consomme les données générées par le premier et sert ultimement à lancer le pipeline d'expérimentation proposé. Ainsi, pour tester un nouveau code d'ouvrage il suffirait de spécifier certaines informations dans un fichier d'une quinzaine de lignes (noms des variables explicatives, endroit où trouver les ensembles de données, etc.). Ensuite, le petit programme lancera la sélection de la mise à l'échelle, la sélection des variables, supprimera les valeurs manquantes ou aberrantes et optimisera les 5 modèles d'IA proposés pour ce nouveau code. Cet outil génère également les graphiques d'explications locales et globales au besoin. Une documentation plus détaillée est fournie pour chacun d'entre eux dans leurs répertoires respectifs. Bref, ces deux outils permettront de faciliter l'expérimentation supplémentaire qui sera sans doute nécessaire avant d'implanter officiellement l'IA en production.

En effet, 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 tous les codes d'ouvrage, procédure qui fut plutôt longue. Éventuellement, le logiciel effectuant l'entraînement des modèles s'abreuverait 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 y 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.

³ <https://github.com/rolare/bdp-to-train-test>

⁴ <https://github.com/rolare/Tunai>

7. CONCLUSION

Huit codes d'ouvrages ont été étudiés dans l'optique d'utiliser l'IA pour estimer le prix unitaire de différents projets de constructions. Ainsi, on a d'abord présenté plusieurs manières d'améliorer la qualité des données de BDP afin d'atteindre le plein potentiel que l'IA peut avoir. En l'occurrence, des techniques de mise à l'échelle, de détection de valeurs aberrantes et de sélection de variables ont été présentées puis testées avec tous les codes d'ouvrage. La détection de valeurs aberrantes et l'algorithme RFE ont un impact particulièrement positif sur la performance des modèles. C'est également le cas de l'ajout des variables externes « Indice économique », « Prix historique du bitume » et « Distance usine ».

Cinq modèles d'IA offrant différents équilibres entre complexité et performance ont été présentés. Parmi ces derniers, le modèle XGBoost a globalement été celui offrant la meilleure performance. Le modèle le plus simple qui a été étudié est l'arbre de décision. Cela dit, malgré un prétraitement des données rigoureux, sa performance a tout de même été surpassée par la plupart des autres modèles, à l'exception du réseau de neurones. En effet, l'expérience a mis en évidence la difficulté d'optimisation et d'entraînement des réseaux de neurones, ce qui implique qu'il est difficile d'atteindre le plein potentiel de performance qu'ils pourraient avoir. En somme, le meilleur modèle pour chaque code d'ouvrage a surpassé les performances de l'estimation humaine, à l'exception du code d'ouvrage organisation de chantier. Ce dernier est légèrement différent puisqu'il est plus global. De plus, certaines observations sur les données de ce code global laissent croire qu'il manquerait d'additionnelles variables à fournir au modèle pour qu'il puisse apprendre la relation et faire une estimation raisonnable.

La technique SHAP a été utilisée pour produire des explications de prédictions locales pour chaque code d'ouvrage, peu importe le modèle utilisé. C'est en effet une technique dite agnostique au modèle qui étudie la variation des prédictions en modifiant successivement les données fournies en entrée au modèle. En concaténant les explications de plusieurs prédictions locales, la méthode SHAP permet d'avoir une idée de l'importance des variables selon le modèle de manière plus globale. Cette technique connaît certaines limites dans la mesure où certains articles ont observé qu'ils étaient capables de modifier une explication SHAP en ajoutant malicieusement des données ayant certaines propriétés à l'ensemble [31]. Il faudrait donc s'assurer que des soumissionnaires ne profitent pas de cette faiblesse pour expliquer un coût d'une manière alternative de sorte que cela les avantage. Il serait intéressant de se pencher sur ce risque dans une recherche ultérieure.

Outre l'explication SHAP, l'utilisation de l'apprentissage automatique connaît également des limites. D'abord, le modèle ne perçoit pas les facteurs extérieurs à l'ensemble de données. Par exemple, il n'y a aucune manière que le modèle connaisse l'existence d'une pandémie avant même que cette dernière ait lieu. Ensuite, le comportement du modèle doit fréquemment être surveillé et mis à

jour. D'ailleurs, dans le cadre du problème actuel, il y aura un modèle à maintenir pour chaque code d'ouvrage, ce qui décuple l'énergie nécessaire. D'autant plus que la qualité de prédiction des modèles dépend de celle des données. Ainsi, une énergie supplémentaire doit être mise quant à la collecte de données. Bref, tel que mentionné précédemment, l'application concrète du projet représente un défi logiciel d'envergure qui nécessitera assurément beaucoup de ressources.

**ANNEXE I : HYPERPARAMÈTRES DES MEILLEURS MODÈLES PAR
CODE D'OUVRAGE**

Couche de surface, enrobé (EnsembleVoters)

Modèle	Paramètre	Valeur
DecisionTreeRegressor	min_samples_split	2
	max_features	0.3
	ccp_alpha	0.0
	min_samples_leaf	8
	max_depth	15
	random_state	42
NeuralNetwork	lr	1 ^e -3
	batch_size	4
	epochs	2000
	layers [size, dropout]	[20, 0.6], [10, 0.6]
RandomForestRegressor	min_samples_split	2
	max_features	0.8
	ccp_alpha	0.0
	min_samples_leaf	6
	max_depth	None
	random_state	42
XGBRegressor	objective	reg:squarederror
	eval_metric	mape
	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
	seed	42

Couche de base, enrobé (RandomForestRegressor)

Paramètre	Valeur
min_samples_split	2
max_features	0.8
ccp_alpha	0.0
min_samples_leaf	4
max_depth	None
random_state	42

Armature (RandomForestRegressor)

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

Déblai de 1^{ère} classe (XGBRegressor)

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

Déblai de 2^{ème} classe (XGBRegressor)

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

Sous-fondation de chaussée (XGBRegressor)

Paramètre	Valeur
objective	reg:squarederror
eval_metric	mape
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
seed	42

Fondation de chaussée (XGBRegressor)

Paramètre	Valeur
objective	reg:squarederror
eval_metric	mape
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
seed	42

Organisation de chantier (NeuralNetwork)

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

ANNEXE II : VARIABLES SÉLECTIONNÉES (PAR ORDRE D'IMPORTANCE) POUR CHAQUE CODE D'OUVRAGE

Armature :

- Quantité estimée
- latitude
- longitude
- annee
- salaires_construction
- indice_economique
- mois_publication_ao_2
- mois_publication_ao_3
- mois_publication_ao_5
- mois_publication_ao_9
- mois_publication_ao_10
- mois_publication_ao_12
- code_dt_Direction générale de la Gaspésie - Îles-de-la-Madeleine
- code_dt_Direction générale de l'exploitation du réseau métropolitain
- code_dt_Direction générale du Saguenay – Lac-Saint-Jean - Chibougamau
- sud_du_fleuve_1
- code_nature_Construction de routes
- code_nature_Structures et/ou ouvrages d'art considérés non complexes
- galvanise_True
- reparation_False

Déblai de 2^{ème} classe :

- Quantité estimée
- latitude
- longitude
- annee
- indice_economique
- type_travaux_Ouvrage d'art
- type_travaux_Revêtement
- mois_publication_ao_7
- code_dt_Direction générale de la Capitale-Nationale
- code_dt_Direction générale de l'exploitation du réseau métropolitain
- sud_du_fleuve_0
- sud_du_fleuve_1
- code_nature_Aéroports et héliports – Capital
- code_nature_Fabrication et pose d'enrobés
- code_nature_Structures et/ou ouvrages d'art considérés non complexes
- code_nature_Terrassements (Amélioration et réfection)
- securite_False

Déblai de 1^{ère} classe

- Quantité estimée
- latitude
- longitude
- type_travaux_Ouvrage d'art
- type_travaux_Revêtement
- mois_publication_ao_7
- mois_publication_ao_10
- code_dt_Direction générale de la Mauricie
- code_dt_Direction générale de l'exploitation du réseau métropolitain
- code_dt_Direction générale des projets et de l'exploitation aéroportuaire
- code_dt_Direction générale du Saguenay – Lac-Saint-Jean - Chibougamau
- sud_du_fleuve_2
- code_nature_Construction (divers)
- code_nature_Terrassements (Amélioration et réfection)

Couche de base, enrobé :

- Quantité estimée
- latitude
- longitude
- annee
- prix_bitume
- salaires_construction
- indice_economique
- localisation_usine
- type_travaux_Ouvrage d'art
- type_travaux_Revêtement
- type_travaux_Terrassement et gravelage
- mois_publication_ao_2
- mois_publication_ao_3
- mois_publication_ao_4
- mois_publication_ao_5
- mois_publication_ao_6
- mois_publication_ao_7
- mois_publication_ao_8
- mois_publication_ao_9
- mois_publication_ao_10
- mois_publication_ao_11
- mois_publication_ao_12
- code_dt_Direction générale de la Capitale-Nationale
- code_dt_Direction générale de la Chaudière-Appalaches
- code_dt_Direction générale de la Côte-Nord
- code_dt_Direction générale de la Gaspésie - Îles-de-la-Madeleine
- code_dt_Direction générale de la Mauricie
- code_dt_Direction générale de l'Estrie
- code_dt_Direction générale de l'Outaouais
- code_dt_Direction générale de l'exploitation du réseau métropolitain
- code_dt_Direction générale des Laurentides - Lanaudière
- code_dt_Direction générale du Bas-Saint-Laurent
- code_dt_Direction générale du Centre-du-Québec
- code_dt_Direction générale du Saguenay – Lac-Saint-Jean - Chibougamau
- sud_du_fleuve_0
- sud_du_fleuve_1
- sud_du_fleuve_2
- code_nature_Construction (divers)
- code_nature_Construction de ponts neufs ou remplacement de tablier de ponts
- code_nature_Construction de routes
- code_nature_Fabrication et pose d'enrobés
- code_nature_Signalisation, éclairage, feux, etc.
- code_nature_Structures et/ou ouvrages d'art considérés non complexes
- code_nature_Terrassements (Amélioration et réfection)
- formule_enrobe_ESG-10
- formule_enrobe_ESG-14
- formule_enrobe_GB-20
- ponceau_False
- ponceau_True
- reparation_False
- reparation_True
- construction_False
- construction_True

Organisation de chantier :

- latitude
- longitude
- annee
- indice_economique
- type_travaux_Terrassement et gravelage
- type_travaux_Tous les types
- mois_publication_ao_2
- mois_publication_ao_7
- mois_publication_ao_10
- mois_publication_ao_11
- mois_publication_ao_12
- code_dt_Direction générale de la Chaudière-Appalaches
- code_dt_Direction générale de la Gaspésie - Îles-de-la-Madeleine
- code_dt_Direction générale de l'Abitibi-Témiscamingue
- code_dt_Direction générale de l'exploitation du réseau métropolitain
- code_dt_Direction générale des grands projets routiers de Montréal et de l'Ouest
- code_dt_Direction générale du Saguenay – Lac-Saint-Jean - Chibougamau
- sud_du_fleuve_0
- sud_du_fleuve_1
- sud_du_fleuve_2
- code_nature_Construction (divers)
- code_nature_Construction de ponts neufs ou remplacement de tablier de ponts
- code_nature_Construction de routes
- code_nature_Signalisation, éclairage, feux, etc.
- code_nature_Terrassements (Amélioration et réfection)

Fondation de chaussée

- Quantité estimée
- latitude
- longitude
- annee
- indice_economique
- type_travaux_Ouvrage d'art
- type_travaux_Services de nature technique
- type_travaux_Système électrotechnique
- type_travaux_Terrassement et gravelage
- mois_publication_ao_1
- mois_publication_ao_2
- mois_publication_ao_3
- mois_publication_ao_4
- mois_publication_ao_6
- mois_publication_ao_7
- mois_publication_ao_8
- mois_publication_ao_9
- code_dt_Direction générale de la Capitale-Nationale
- code_dt_Direction générale de la Gaspésie - Îles-de-la-Madeleine
- code_dt_Direction générale des projets et de l'exploitation aéroportuaire
- code_dt_Direction générale du Centre-du-Québec
- sud_du_fleuve_0
- sud_du_fleuve_1
- code_nature_Construction (divers)
- code_nature_Construction de ponts neufs ou remplacement de tablier de ponts
- code_nature_Construction de routes
- code_nature_Signalisation, éclairage, feux, etc.
- code_nature_Structures et/ou ouvrages d'art considérés non complexes
- code_nature_Terrassements (Amélioration et réfection)
- materiau_granulaire_MG 112
- materiau_granulaire_MG 20
- materiau_granulaire_MG 20b modifié
- ponceau_False

Sous-fondation de chaussée

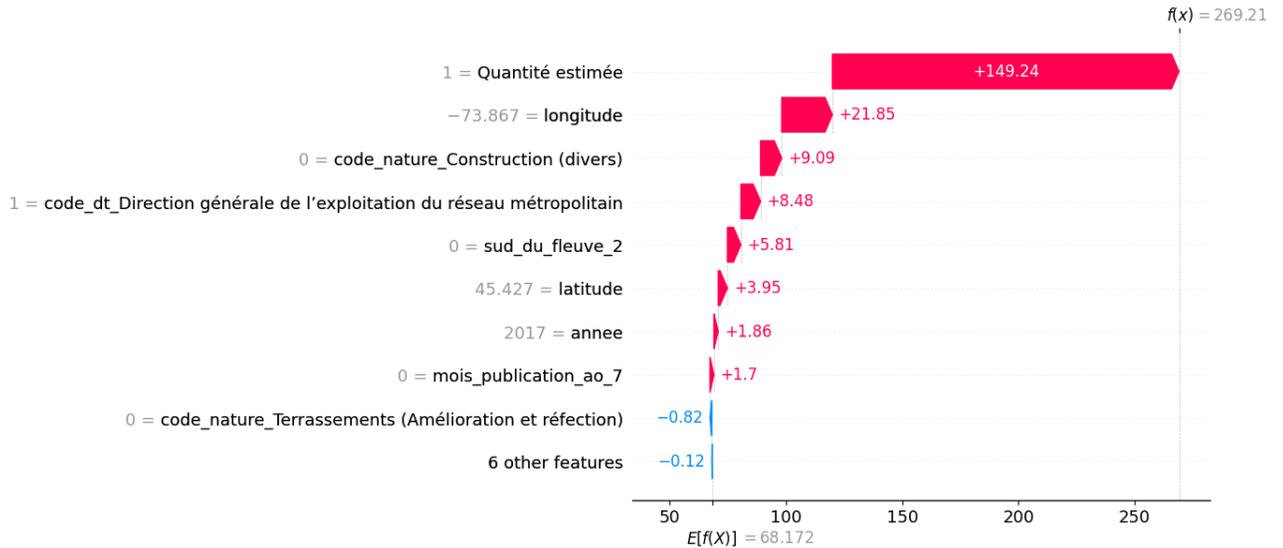
- Quantité estimée
- latitude
- longitude
- annee
- indice_economique
- type_travaux_Ouvrage d'art
- type_travaux_Système électrotechnique
- mois_publication_ao_2
- mois_publication_ao_3
- mois_publication_ao_4
- mois_publication_ao_5
- mois_publication_ao_6
- mois_publication_ao_7
- mois_publication_ao_8
- mois_publication_ao_9
- mois_publication_ao_11
- code_dt_Direction des parcs routiers
- code_dt_Direction générale de la Capitale-Nationale
- code_dt_Direction générale de la Chaudière-Appalaches
- code_dt_Direction générale de la Côte-Nord
- code_dt_Direction générale de la Gaspésie - Îles-de-la-Madeleine
- code_dt_Direction générale de la Mauricie
- code_dt_Direction générale de l'Abitibi-Témiscamingue
- code_dt_Direction générale de l'Estrie
- code_dt_Direction générale de l'Outaouais
- code_dt_Direction générale de l'exploitation du réseau métropolitain
- code_dt_Direction générale des Laurentides - Lanaudière
- code_dt_Direction générale des grands projets routiers de Montréal et de l'Ouest
- code_dt_Direction générale du Bas-Saint-Laurent
- code_dt_Direction générale du Centre-du-Québec
- code_dt_Direction générale du Saguenay – Lac-Saint-Jean - Chibougamau
- sud_du_fleuve_0
- sud_du_fleuve_1
- code_nature_Construction (divers)
- code_nature_Construction de ponts neufs ou remplacement de tablier de ponts
- code_nature_Construction de routes
- code_nature_Signalisation, éclairage, feux, etc.
- code_nature_Structures et/ou ouvrages d'art considérés non complexes
- code_nature_Terrassements (Amélioration et réfection)
- materiau_granulaire_MG 112
- materiau_granulaire_MG 20
- materiau_granulaire_MG 56
- ponceau_False

Couche de surface, enrobé :

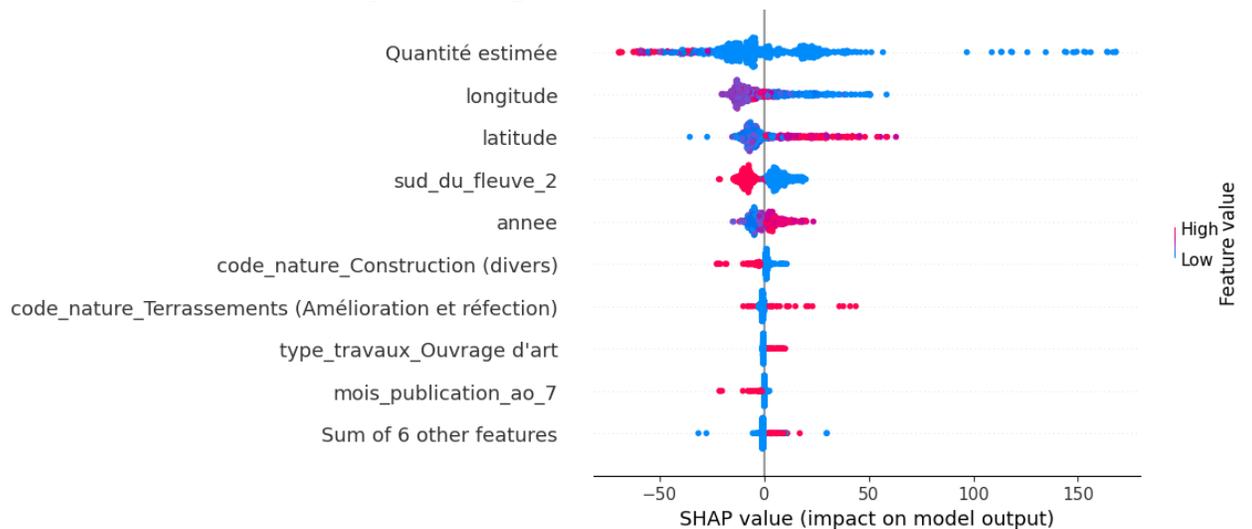
- Quantité estimée
- latitude
- longitude
- annee
- prix_bitume
- indice_economique
- localisation_usine
- type_travaux_Ouvrage d'art
- type_travaux_Revêtement
- type_travaux_Terrassement et gravelage
- mois_publication_ao_1
- mois_publication_ao_3
- mois_publication_ao_4
- mois_publication_ao_5
- mois_publication_ao_6
- mois_publication_ao_9
- mois_publication_ao_11
- code_dt_Direction générale de la Capitale-Nationale
- code_dt_Direction générale de la Côte-Nord
- code_dt_Direction générale de la Gaspésie - Îles-de-la-Madeleine
- code_dt_Direction générale de la Mauricie
- code_dt_Direction générale de l'Abitibi-Témiscamingue
- code_dt_Direction générale de l'Estrie
- code_dt_Direction générale de l'exploitation du réseau métropolitain
- code_dt_Direction générale des Laurentides - Lanaudière
- code_dt_Direction générale du Bas-Saint-Laurent
- code_dt_Direction générale du Saguenay – Lac-Saint-Jean - Chibougamau
- sud_du_fleuve_0
- sud_du_fleuve_1
- sud_du_fleuve_2
- code_nature_Construction (divers)
- code_nature_Construction de ponts neufs ou remplacement de tablier de ponts
- code_nature_Construction de routes
- code_nature_Fabrication et pose d'enrobés
- code_nature_Signalisation, éclairage, feux, etc.
- code_nature_Structures et/ou ouvrages d'art considérés non complexes
- code_nature_Terrassements (Amélioration et réfection)
- formule_enrobe_EG-10
- formule_enrobe_ESG-10
- formule_enrobe_ESG-14
- formule_enrobe_autre
- ponceau_False
- reparation_False
- construction_False

ANNEXE III : EXPLICATIONS LOCALES ET GLOBALES PAR CODE D'OUVRAGE

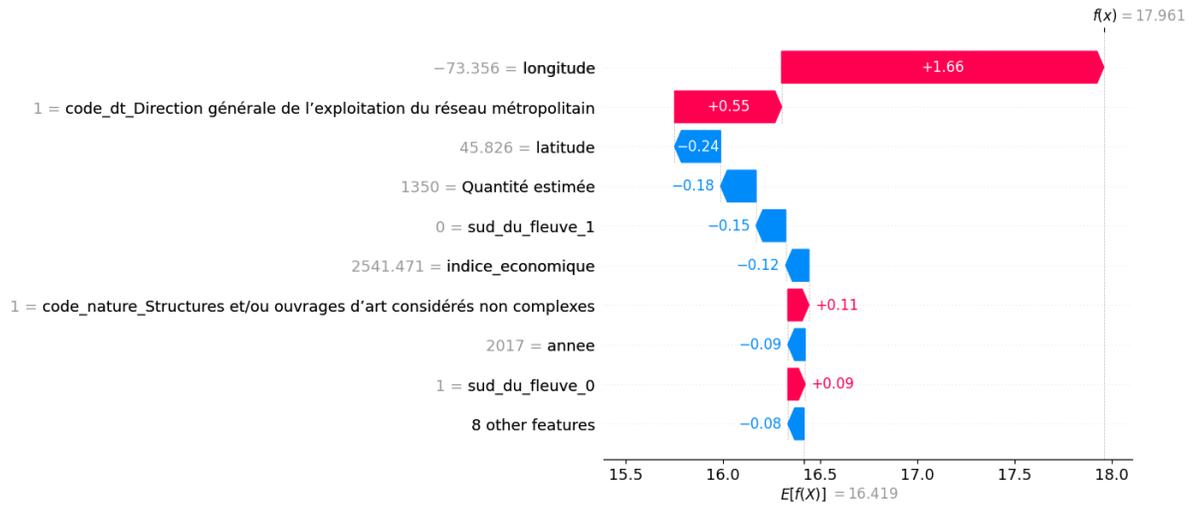
Déblai de 1^{ère} classe : explication locale



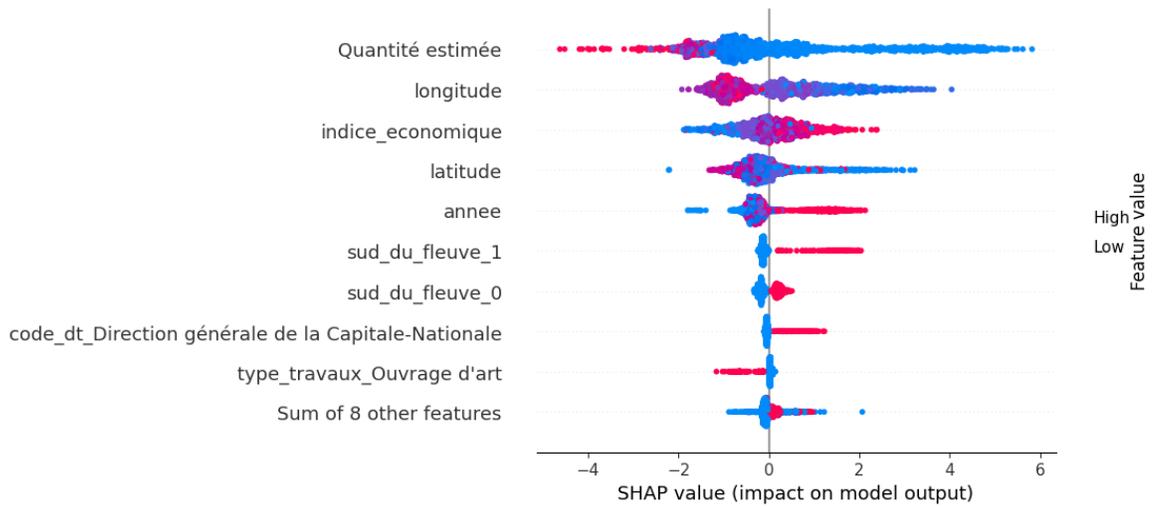
Déblai de 1^{ère} classe : explication globale



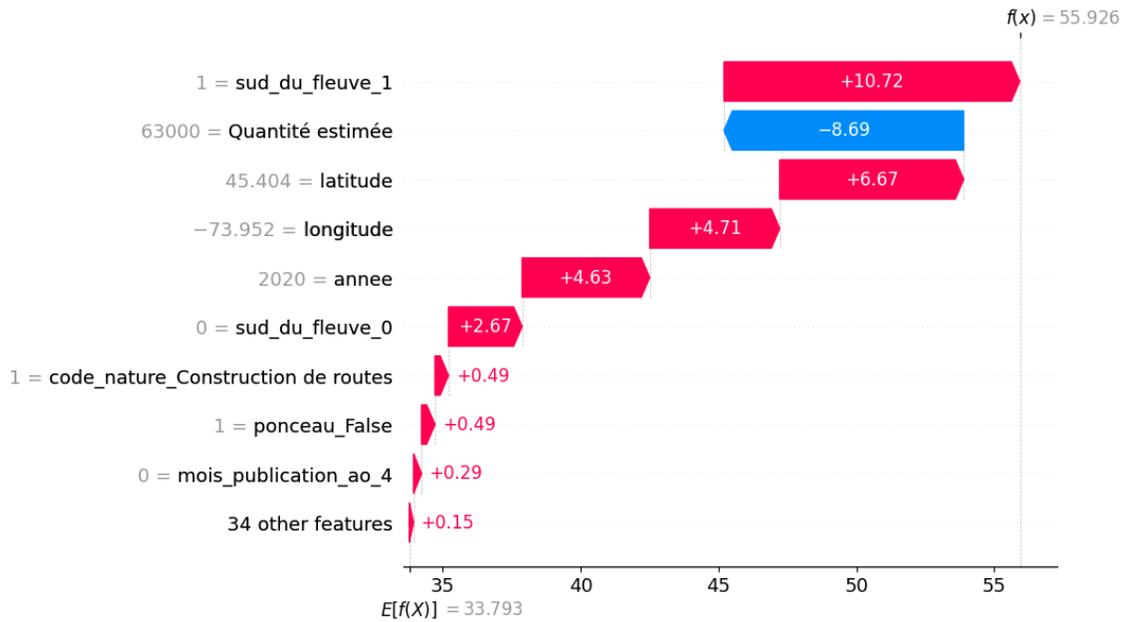
Déblai de 2^{ème} classe : explication locale



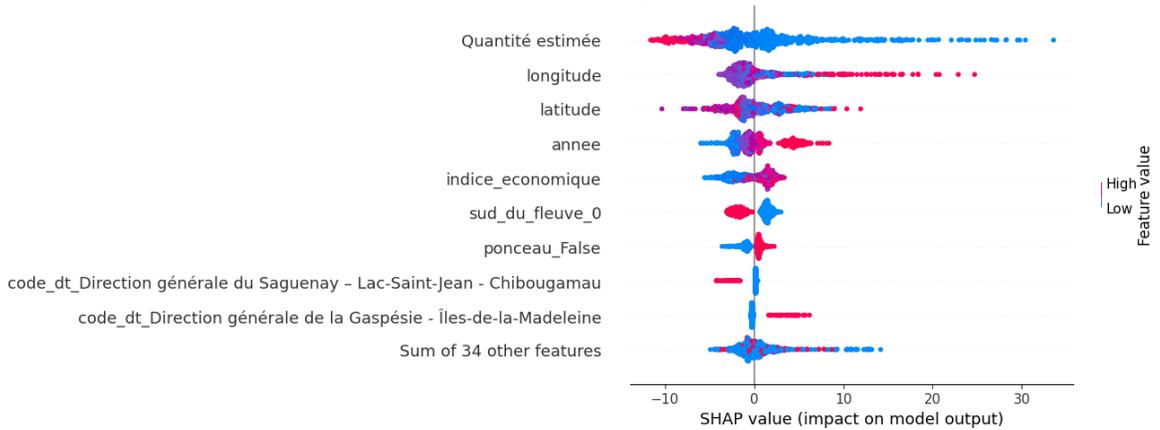
Déblai de 2^{ème} classe : explication globale



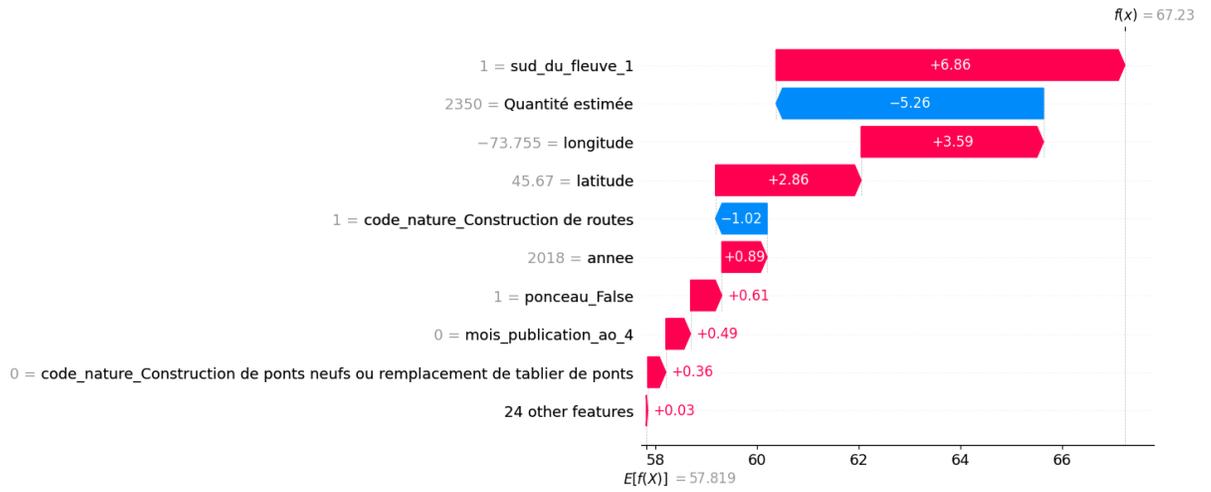
Sous-fondation de chaussée : explication locale



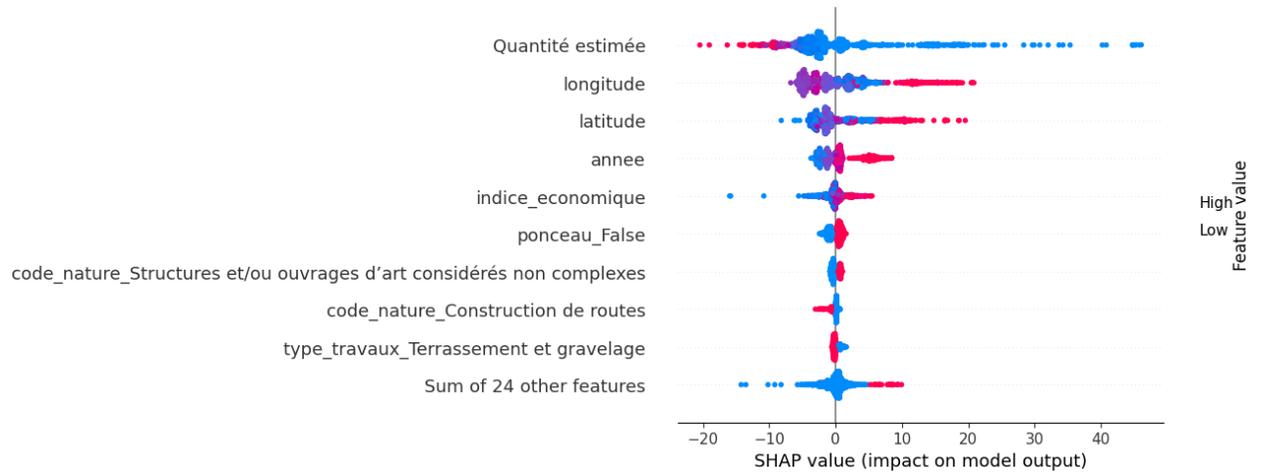
Sous-fondation de chaussée : explication globale



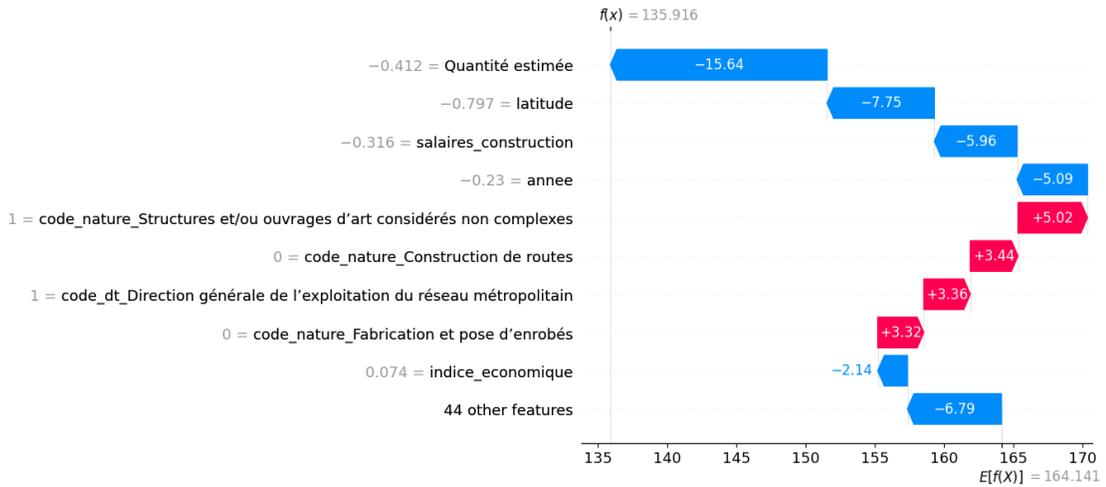
Fondation de chaussée : explication locale



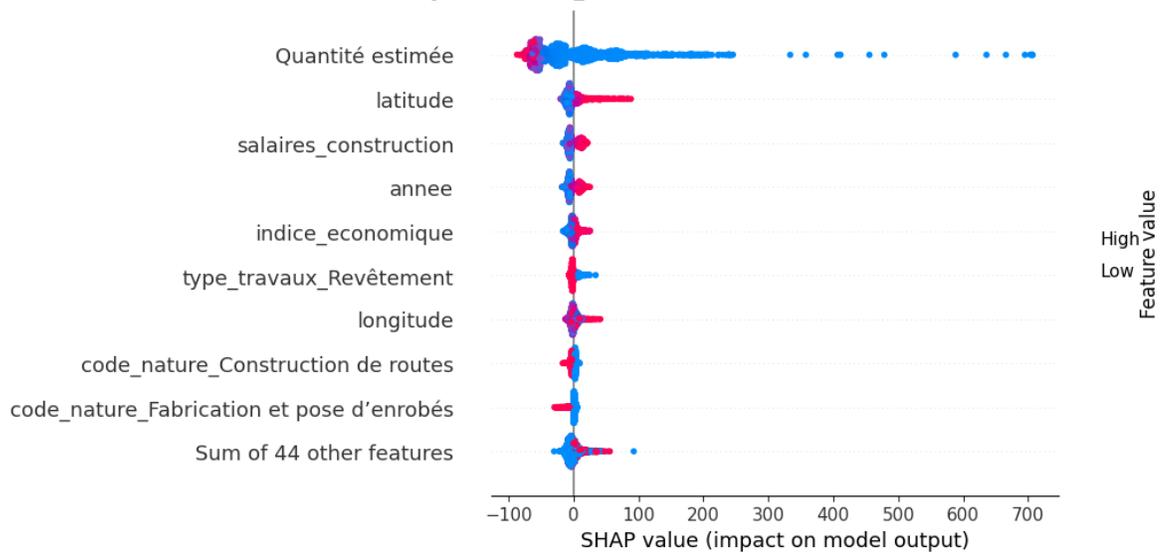
Fondation de chaussée : explication globale



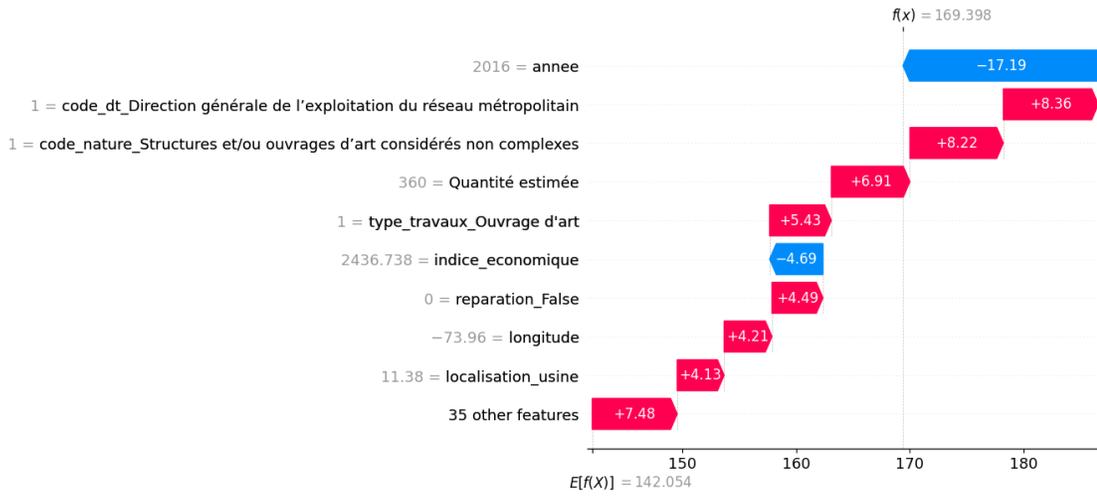
Couche de base, enrobé : explication locale



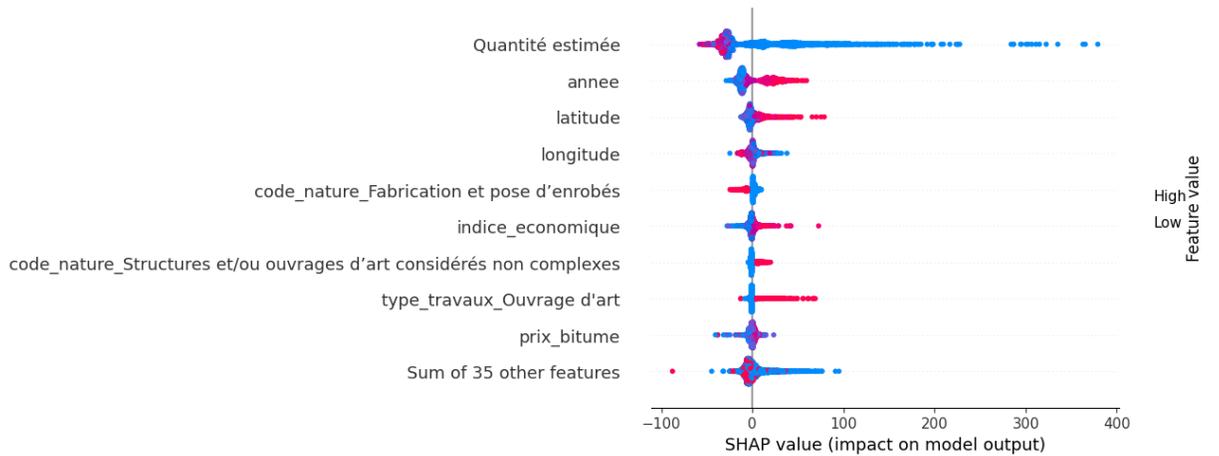
Couche de base, enrobé : explication globale



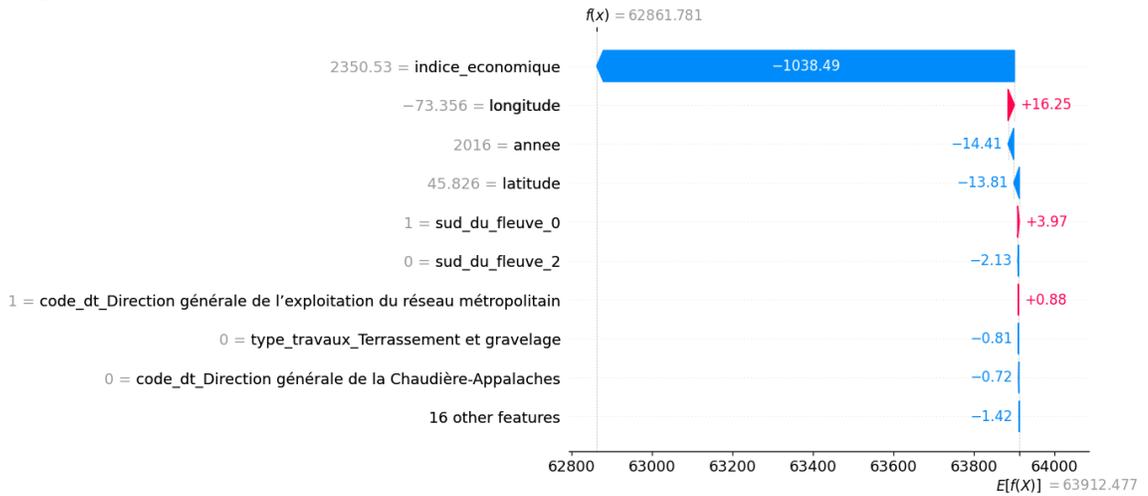
Couche de surface, enrobé : explication locale



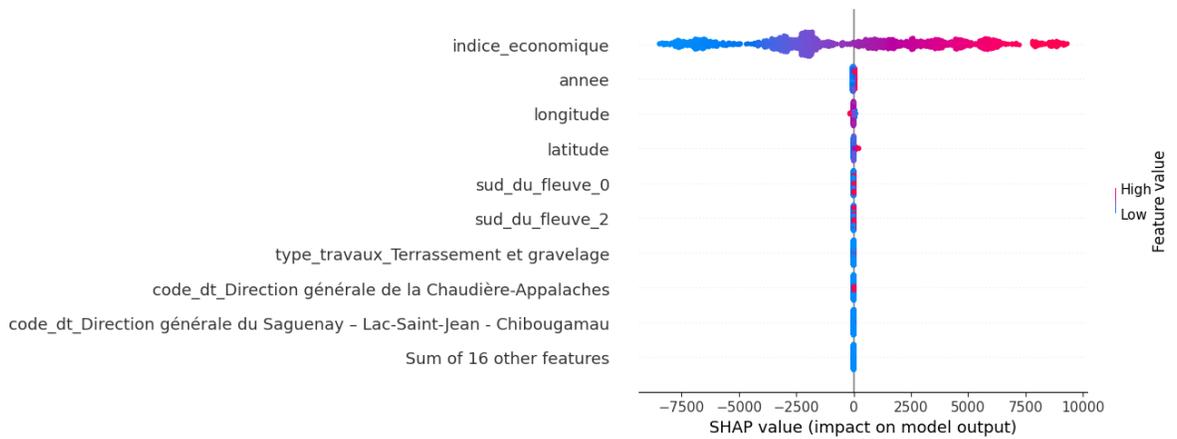
Couche de surface, enrobé : explication globale



Organisation de chantier : explication locale

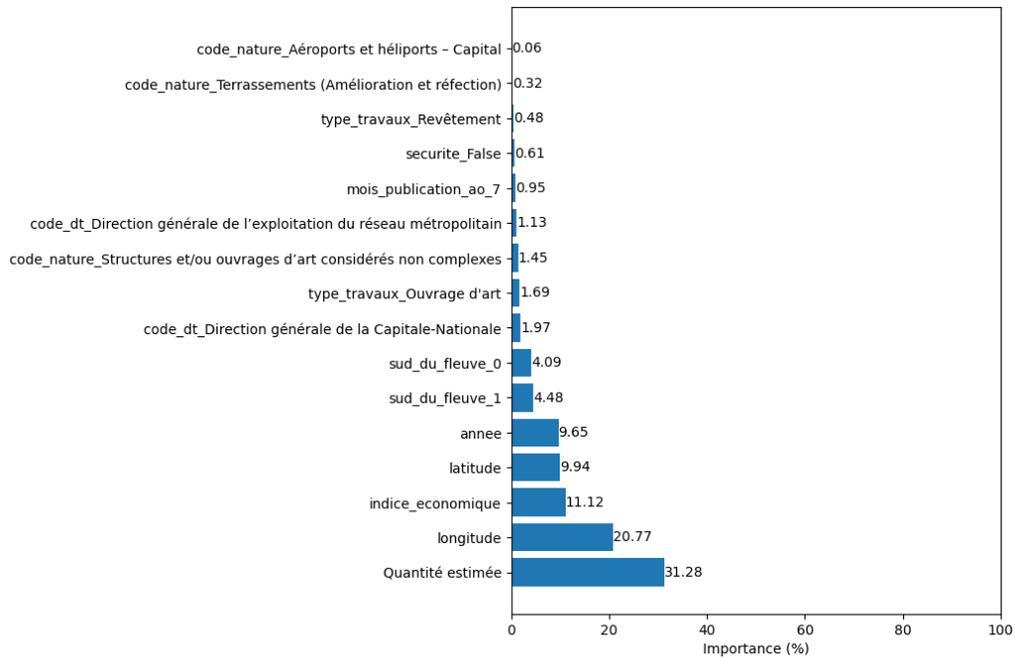


Organisation de chantier : explication globale

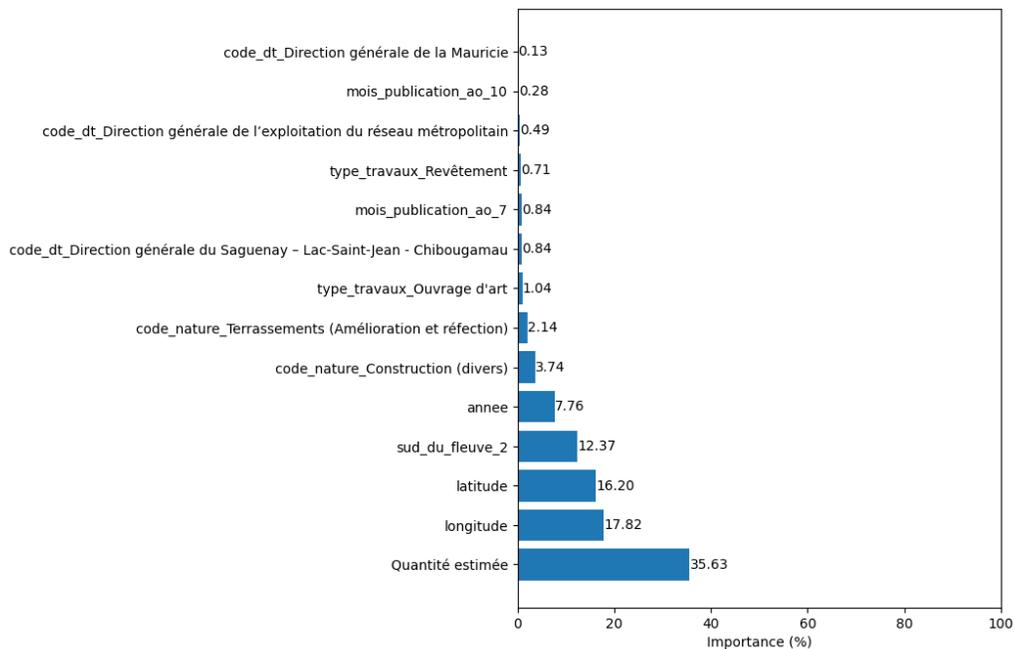


ANNEXE IV : POURCENTAGE D'IMPORTANCE DES VARIABLES EXPLICATIVES PAR CODE D'OUVRAGE SELON LA MÉTHODE SHAP

Déblai de 1^{ère} classe

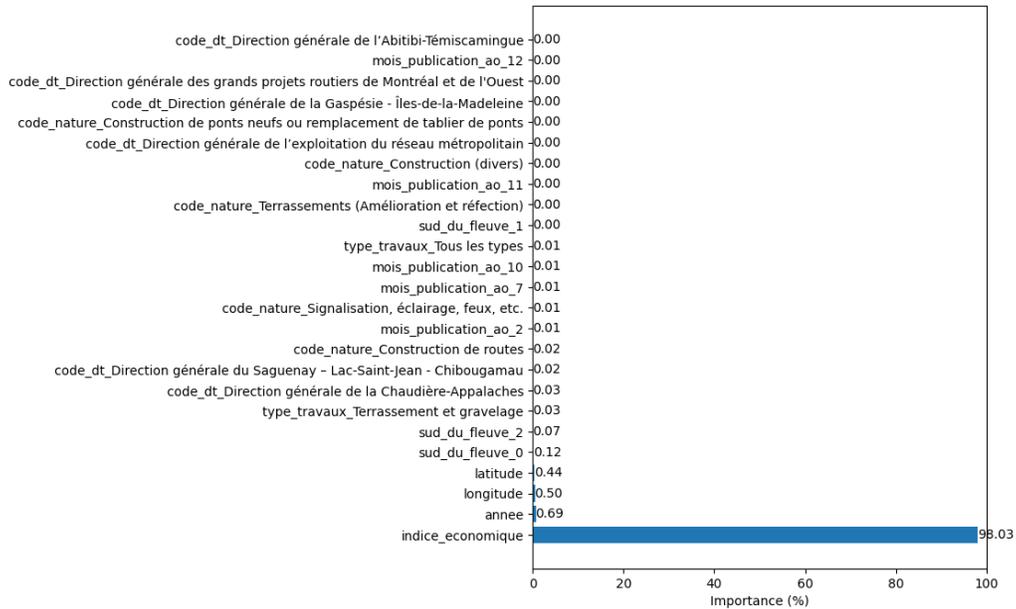


Déblai de 2^{ème} classe

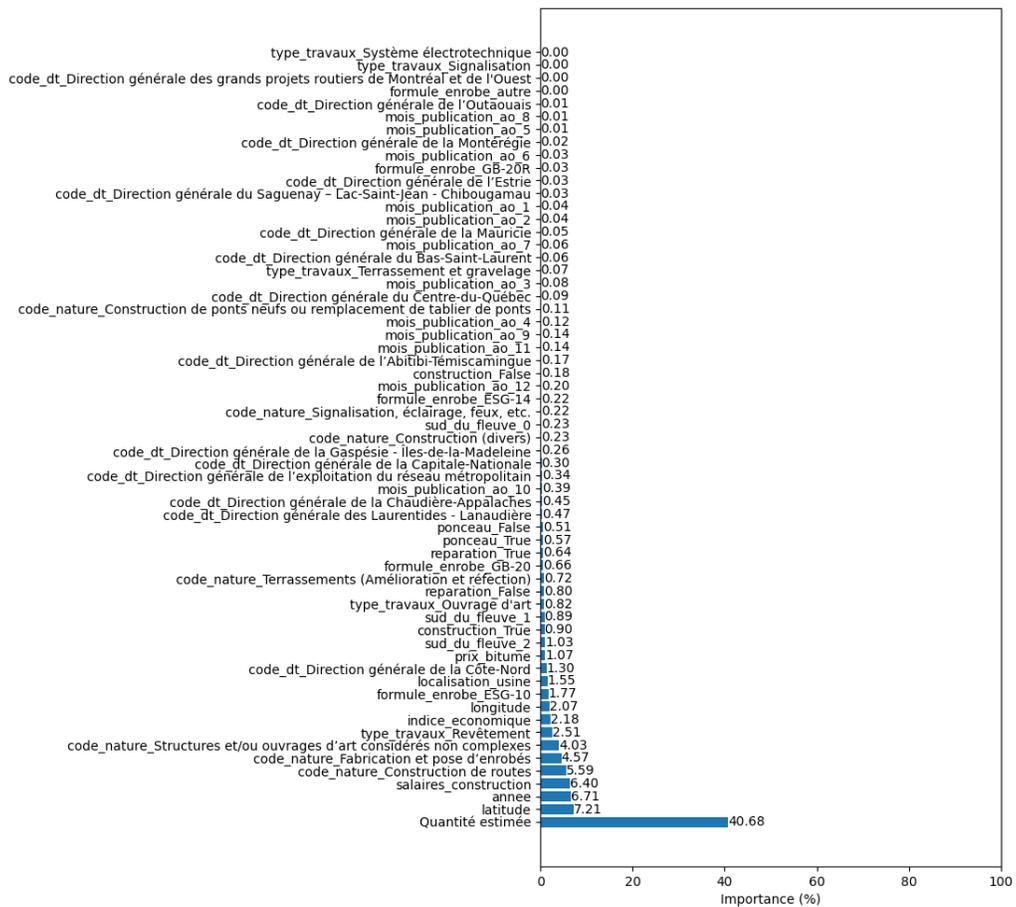


INTÉGRATION DE L'INTELLIGENCE ARTIFICIELLE (IA) À LA BASE DE DONNÉES BORDEREAU ET DEMANDE DE PAIEMENT (BDP)

Organisation de chantier

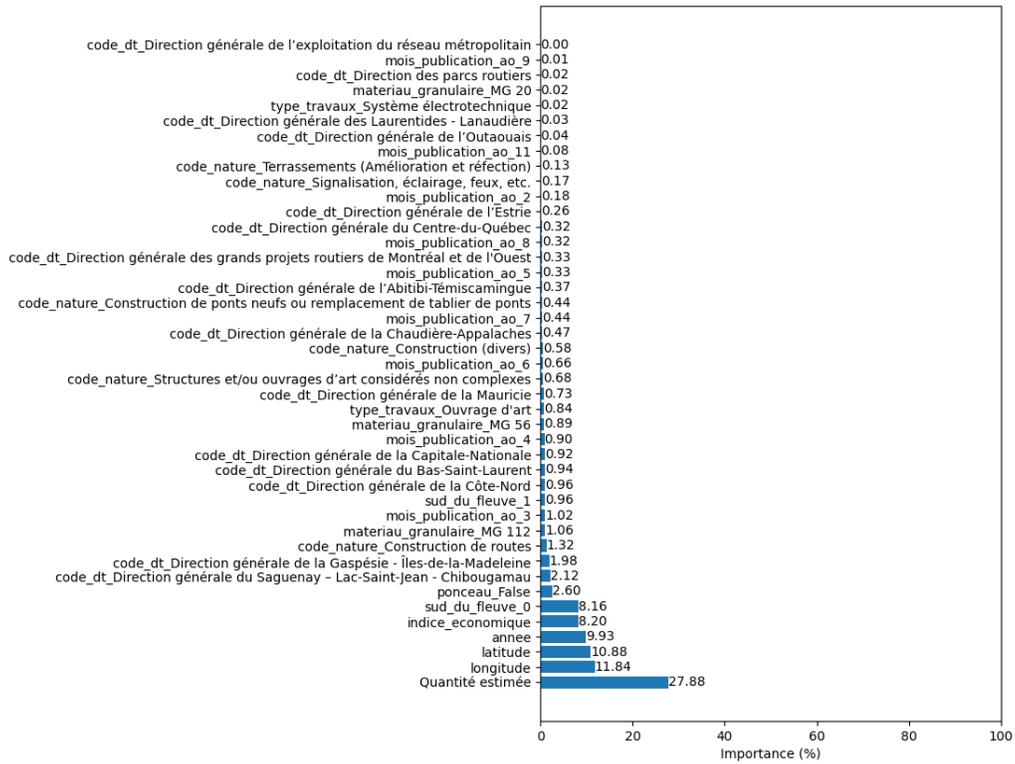


Couche de base, enrobé

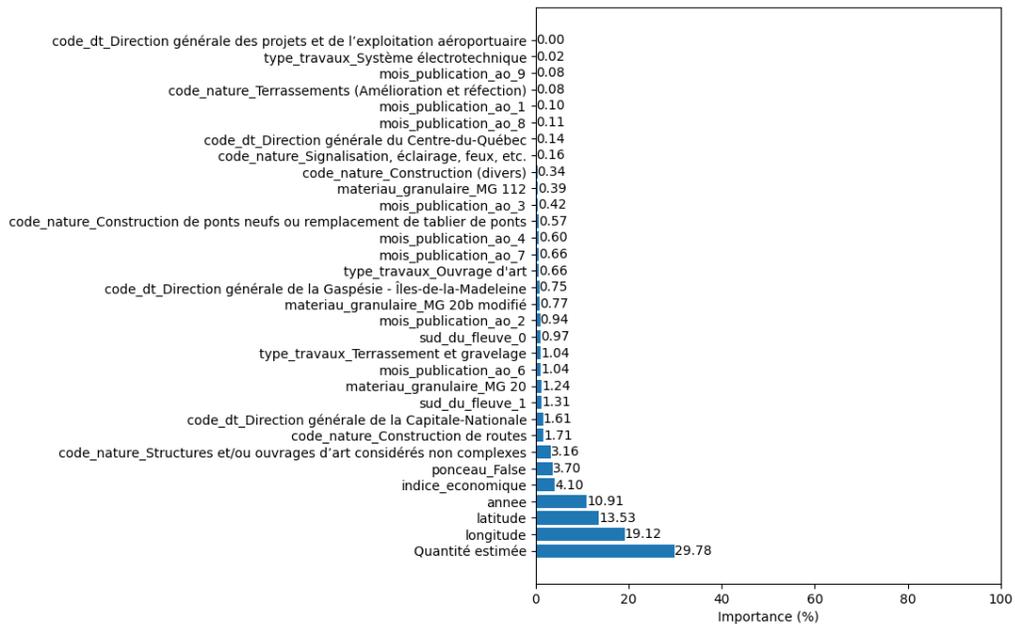


INTÉGRATION DE L'INTELLIGENCE ARTIFICIELLE (IA) À LA BASE DE DONNÉES BORDEREAU ET DEMANDE DE PAIEMENT (BDP)

Sous-fondation de chaussé

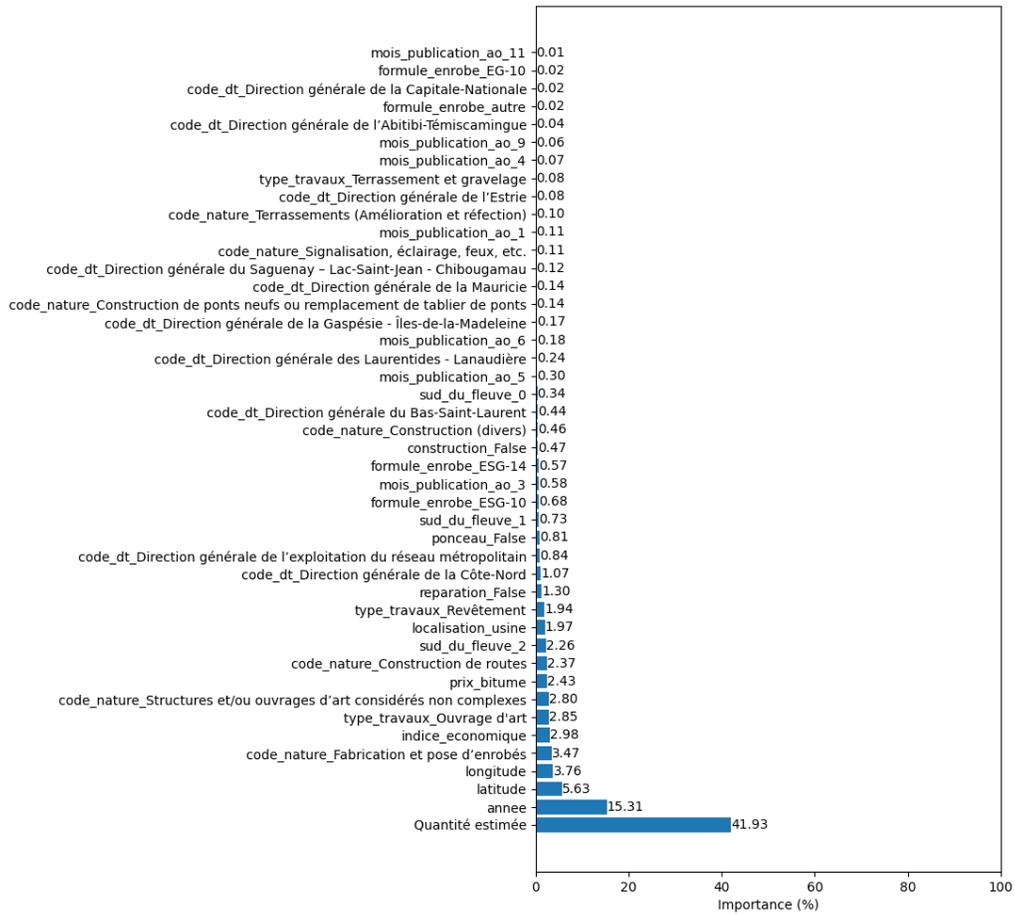


Fondation de chaussée



INTÉGRATION DE L'INTELLIGENCE ARTIFICIELLE (IA) À LA BASE DE DONNÉES
BORDEREAU ET DEMANDE DE PAIEMENT (BDP)

Couche de surface, enrobé



RÉFÉRENCES BIBLIOGRAPHIQUES

- [1]
V. Kouskoulas et E. Koehn, « Predesign Cost-Estimation Function for Buildings », *Journal of the Construction Division*, vol. 100, n° 4, p. 589-604, déc. 1974, doi: 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, déc. 1985, doi: 10.1080/01446198500000015.
- [3]
F. Khosrowshahi et A. Kaka, « Estimation of project total cost and duration for housing projects in the U.K. », 1996, doi: 10.1016/0360-1323(96)00003-0.
- [4]
R.A. McKim, « Neural network applications to cost engineering », *American Association of Cost Engineers*, vol. 35, p. 31, 1993, Consulté le: 16 mai 2022. [En ligne]. Disponible sur: <https://www.proquest.com/scholarly-journals/neural-network-applications-cost-engineering/docview/220434454/se-2?accountid=40695>
- [5]
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 sur: <https://www.proquest.com/scholarly-journals/neural-networks-versus-parameter-based/docview/220470091/se-2?accountid=40695>
- [6]
H. Li, « Neural networks for construction cost estimation », *Building Research & Information*, vol. 23, n° 5, p. 279-284, oct. 1995, doi: 10.1080/09613219508727476.
- [7]
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, doi: 10.1016/j.buildenv.2004.02.013.
- [8]
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, janv. 2017, doi: 10.1016/j.proeng.2017.07.199.
- [9]
G. Mahalakshmi et C. Rajasekaran, « Early Cost Estimation of Highway Projects in India Using Artificial Neural Network », in *Sustainable Construction and Building Materials*, Singapore, 2019, p. 659-672. doi: 10.1007/978-981-13-3317-0_59.
- [10]
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° 1, p. 41-64, janv. 2019, doi: 10.1504/IJPOM.2019.098712.

- [11]
C. L. C. Roxas, N. R. Roxas, J. Cristobal, S. E. Hao, R. Marie Rabino, et F. Revalde, « Modeling Road Construction Project Cost in the Philippines Using the Artificial Neural Network Approach », in *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, nov. 2019, p. 1-5. doi: 10.1109/HNICEM48295.2019.9072820.
- [12]
K. Tijanić, D. Car-Pušić, et M. Šperac, « Cost estimation in road construction using artificial neural network », *Neural Comput & Applic*, vol. 32, n° 13, p. 9343-9355, juill. 2020, doi: 10.1007/s00521-019-04443-y.
- [13]
S. Kim, C.-Y. Choi, M. Shahandashti, et K. R. Ryu, « Improving Accuracy in Predicting City-Level Construction Cost Indices by Combining Linear ARIMA and Nonlinear ANNs », *Journal of Management in Engineering*, vol. 38, n° 2, p. 04021093, mars 2022, doi: 10.1061/(ASCE)ME.1943-5479.0001008.
- [14]
B. Xiong, S. Newton, V. Li, M. Skitmore, et B. Xia, « Hybrid approach to reducing estimating overfitting and collinearity », *Engineering, Construction and Architectural Management*, vol. 26, n° 10, p. 2170-2185, janv. 2019, doi: 10.1108/ECAM-08-2018-0353.
- [15]
S. Petrusheva, D. Car-Pušić, et V. Zileska-Pancovska, « Support Vector Machine Based Hybrid Model for Prediction of Road Structures Construction Costs », *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 222, p. 012010, janv. 2019, doi: 10.1088/1755-1315/222/1/012010.
- [16]
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° 1, p. 03119008, janv. 2020, doi: 10.1061/(ASCE)CO.1943-7862.0001678.
- [17]
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, août 2016, doi: 10.1145/2939672.2939785.
- [18]
L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, et A. Gulin, « CatBoost: unbiased boosting with categorical features », *arXiv:1706.09516 [cs]*, janv. 2019, Consulté le: 16 mai 2022. [En ligne]. Disponible sur: <http://arxiv.org/abs/1706.09516>
- [19]
S. B. Jabeur, C. Gharib, S. Mefteh-Wali, et W. B. Arfi, « CatBoost model and artificial intelligence techniques for corporate failure prediction », *Technological Forecasting and Social Change*, vol. 166, p. 120658, mai 2021, doi: 10.1016/j.techfore.2021.120658.
- [20]

A. Mahdavian, A. Shojaei, M. Salem, J. S. Yuan, et A. A. Oloufa, « Data-Driven Predictive Modeling of Highway Construction Cost Items », *Journal of Construction Engineering and Management*, vol. 147, n° 3, p. 04020180, mars 2021, doi: 10.1061/(ASCE)CO.1943-7862.0001991.

[21]

N. Frosst et G. Hinton, « Distilling a Neural Network Into a Soft Decision Tree », *arXiv:1711.09784 [cs, stat]*, nov. 2017, Consulté le: 16 mai 2022. [En ligne]. Disponible sur: <http://arxiv.org/abs/1711.09784>

[22]

T. Vidal, T. Pacheco, et M. Schiffer, « Born-Again Tree Ensembles », *arXiv:2003.11132 [cs, stat]*, août 2020, Consulté le: 16 mai 2022. [En ligne]. Disponible sur: <http://arxiv.org/abs/2003.11132>

[23]

S. Lundberg et S.-I. Lee, « A Unified Approach to Interpreting Model Predictions », *arXiv:1705.07874 [cs, stat]*, nov. 2017, Consulté le: 16 mai 2022. [En ligne]. Disponible sur: <http://arxiv.org/abs/1705.07874>

[24]

C. Rudin, « Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead », *Nat Mach Intell*, vol. 1, n° 5, p. 206-215, mai 2019, doi: 10.1038/s42256-019-0048-x.

[25]

« IQ30 : tout savoir sur l'indice boursier du Québec ». <https://www.fondsftq.com/fr-ca/particuliers/epargne-positive/iq30-indice-boursier-quebec> (consulté le 16 mai 2022).

[26]

I. Guyon, J. Weston, S. Barnhill, et V. Vapnik, « Gene Selection for Cancer Classification using Support Vector Machines », *Machine Learning*, vol. 46, n° 1, p. 389-422, janv. 2002, doi: 10.1023/A:1012487302797.

[27]

« scikit-learn: machine learning in Python — scikit-learn 1.1.0 documentation ». <https://scikit-learn.org/stable/> (consulté le 17 mai 2022).

[28]

F. T. Liu, K. M. Ting, et Z.-H. Zhou, « Isolation Forest », in *2008 Eighth IEEE International Conference on Data Mining*, déc. 2008, p. 413-422. doi: 10.1109/ICDM.2008.17.

[29]

M. M. Breunig, H.-P. Kriegel, R. T. Ng, et J. Sander, « LOF: identifying density-based local outliers », *SIGMOD Rec.*, vol. 29, n° 2, p. 93-104, mai 2000, doi: 10.1145/335191.335388.

[30]

R. Zhang, S. Zhang, S. Muthuraman, et J. Jiang, « One class support vector machine for anomaly detection in the communication network performance data », in *Proceedings of the 5th conference on Applied electromagnetics, wireless and optical communications*, Stevens Point, Wisconsin, USA, déc. 2007, p. 31-37.

[31]

D. Slack, S. Hilgard, E. Jia, S. Singh, et H. Lakkaraju, « Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods », *arXiv:1911.02508 [cs, stat]*, févr. 2020, Consulté le: 20 mai 2022. [En ligne]. Disponible sur: <http://arxiv.org/abs/1911.02508>