Submitted to *INFORMS Journal on Applied Analytics*
manuscript OM-05-2023-0027

Estimating Road Construction Costs with Explainable Machine Learning

Rosanne Larocque
Polytechnique Montréal, Montreal, Canada
rosanne.larocque@polymtl.ca

Anne-Marie Boulé
Ministère des Transports et de la Mobilité durable, Quebec, Canada
anne-marie.boule@transports.gouv.qc.ca

Quentin Cappart
Polytechnique Montréal, Montreal, Canada
quentin.cappart@polymtl.ca

A preliminary estimation of construction costs is a crucial operation of any project related to civil engineering. An accurate estimation ensures a proper management of the available funds and helps the project managers in their decision-making processes. For instance, it is common that specific subtasks of the project are delegated to private subcontractors. Through a call for tenders, each eventual subcontractor has the opportunity to propose a bid with a price for supplying the service. As the call is generally public, a competition may arise between subcontractors. This impacts the price proposed by the competitors to get the contract. In order to select a subcontractor, the project manager needs to have an accurate idea of a reasonable price for the subtask given. A price higher than expected is undesirable, but a price significantly lower than expected may also result in a bad quality of service. The project manager must also be able to explain to stakeholders why a price is suited and justify why a specific subcontractor has been selected. Providing an estimation both accurate and transparent is a hard problem for the project manager. A growing trend is to leverage machine learning for this estimation, but designing a model that is both accurate and explainable is still a challenge. Another difficulty is that an approach which is accurate for estimating the cost of a subtask may not be efficient for another one. Based on this context, this paper introduces a framework for estimating construction costs while tackling both challenges. It is based on six machine learning models, and on Shapley Additive explanations. This project has been commissioned by the Ministry of Transport and Sustainable Mobility a public agency responsible for transport infrastructure in Quebec, Canada. Experiments have been carried out on real data, covering historical road construction costs of 11,646 contracts and 8 subtasks from 2014 to 2021. Results show that the framework is able to surpass the accuracy of human estimations up to 31.56%, while being able to adequately explain how the estimations have been obtained.

Introduction

Early stage estimation of costs plays an important role in any construction project, and has a significant impact on the decision-making process carried out by project managers (Kouskoulas and Koehn 1974, Bowen and Edwards 1985). Provided with accurate estimates, they will be able to perform better decisions, and avoid out-of-budget solutions. Large-scale road construction projects (e.g., building a bridge or a new highway) include many tasks and generally involve experts in several domains, such as civil engineers, architects, or soil scientists. Given the variety of tasks that must be carried out and the different expertise required for performing them, it is common to outsource some of them to private subcontractors (Hui et al. 2008). Through a call for tenders (i.e., a formal invitation to make an offer for the supply of a service), each eventual subcontractor has the opportunity to propose a bid with a price for carrying out the task. As the call is generally public, a competition may arise between subcontractors. This impacts the price proposed by the competitors to get the contract. In order to select a subcontractor, the project manager needs to have an accurate idea of a reasonable price for the task given. On the one hand, a price higher than expected is undesirable, but on the other hand a price significantly lower than expected may also result in a bad quality of service. Ideally, the price should accurately mirror the actual cost of the task, grounded in quantifiable factors like raw material costs, task complexity, or required workforce. Initially, this estimation was commonly conducted by experts, coupled with an analysis of historical costs for similar tasks. Although such estimations by experts are still commonly used, a growing trend is to leverage the abilities of machine learning for performing this estimation (McKim 1993, Arafa and Alqedra 2011). The goal is to obtain cost estimations that as accurate as possible.

However, relying on an estimation, albeit very accurate, is not enough for a practical use. The project manager must also be able to explain to stakeholders why a price is suited for a task and justify why a specific subcontractor has been selected. This transparency is paramount to avoid illegal collusion, which is still a critical concern nowadays (Shan et al. 2018). This consideration is commonly referred to as *explainable machine learning* (Gade et al. 2019). Briefly, a model is *explainable* if we are able to easily comprehend why specific decisions or predictions have been made. Unfortunately, the explainability level of a machine learning model is usually inverse to its prediction accuracy. In other words, the higher the prediction accuracy, the lower the model explainability. Depending on how

much they want the model to be accurate or explainable, practitioners must carefully select which kind of learning approaches to use. For instance, decision trees (Quinlan 1986) are simple approaches that are easy to explain, unlike deep neural networks (LeCun et al. 2015) that are known to suffer from a lack of explainability.

Another difficulty is that an accurate approach for estimating the cost related to a specific task may not be efficient for another one. There are many reasons explaining this behaviour, such as the number of available data to train the models. Some tasks may have a large number of historical data whereas they can be scarce for others. Besides, learning approaches do not have the same sensitivity to the number of data. Deep neural networks require many samples for getting a good accuracy, whereas random forests (Breiman 2001) can handle efficiently situations with only few training data.

Coming up with an approach to estimate the cost of tasks related to a construction project is then a challenge that many industries or government agencies are faced with. It is for instance the case of the *Ministry of Transport and Sustainable Mobility*, a public agency responsible for transport infrastructure in Quebec, Canada. Every year, the Ministry must carry out on average more than 700 contracts involving 40,000 different tasks that are planned to be outsourced. The estimation is directly used to select general contractors for the task. Currently, the relevance of an estimation for a task is done by field experts from the Ministry and is based on the historical costs of previous related tasks.

In 2017, the *Auditor General of Quebec*, an office dedicated to promoting, through audit, parliamentary control over public funds and other public property, realized an audit and highlighted the need for the Ministry to improve the cost estimation of project related to civil engineering. In this context, obtaining predictions that are both accurate and explainable is critical. The contributions proposed in this paper have been dedicated to this purpose.

Given the success of machine learning in diverse industrial applications (Rai et al. 2021) such as in transportation (Arıkan et al. 2023), supply chain (Dodin et al. 2023), fraud detection (Nanduri et al. 2020), or public reporting (Muir and Reich 2021), its application for performing cost estimations in an automated fashion is a promising and profitable direction. Three requirements must be fulfilled. First, the estimation obtained must be accurate and at least better than the ones that are obtained with human experts. Second, it must be explainable, meaning that stakeholders must be able to understand what are

the root causes of a prediction. Third, it must be flexible to accommodate estimations on various tasks and on different levels of data proliferation. Based on this context, this paper presents an end-to-end framework based on machine learning, which is dedicated to estimate the cost of tasks related to road construction projects while fulfilling these three requirements. Specifically, the contributions are as follows.

1. A prediction module composed of six supervised machine learning approaches, namely, decision tree (Quinlan 1986), random forests (Breiman 2001), XGBoost (Chen and Guestrin 2016), CatBoost (Prokhorenkova et al. 2018), neural networks (LeCun et al. 2015), and ensemble methods (Dietterich 2000). Depending on the task, the model with the best accuracy is selected.

2. An explanation module based on a game theoretic approach, *Shapley Additive explanations* (Lundberg and Lee 2017). An important feature of this module is its independence with the underlying learning approach. It can then directly be used to obtain explanations for the six previous models without any modification.

3. Experiments on real data issued by the Ministry of Transport and Sustainable Mobility of Quebec covering historical road construction costs of 11,646 contracts and 8 tasks from 2014 to 2021. The results obtained show that the framework is able to surpass the accuracy of human estimations by 3.42% and up to 31.56% on new submissions while being able to adequately explain how the estimation has been obtained.

This project has been commissioned by the Ministry of Transport and Sustainable Mobility of Quebec and the relevance of the predictions have been validated by experts from the Ministry both in terms of accuracy and explainability. The paper is structured as follows. The next section introduces related approaches to estimate construction costs and on providing explanation of machine learning approaches. Then, the prediction problem considered in this paper is formalized together with a description of the dataset available. The six learning approaches we consider are then described, and our core contribution, referred to as the *data-driven prediction pipeline*, is presented subsequently. Finally, we present an experimental analysis that includes a survey conducted among experts and a discussion about the prospective utilization of the pipeline for the commissioner. The objective is to evaluate the prediction quality and validate the framework's interpretability level.

Literature Review

The literature review is divided into two parts. First, we present and analyze related works in the field of construction cost estimation. Second, we describe commonly used recent approaches for explaining the behaviour of machine learning models. We also discuss recent applications of explainable machine learning for industrial problems.

Estimating Construction Cost

Delivering precise and transparent cost estimations has consistently been a fundamental factor in the decision-making process. Since the 1970s, linear regression has been widely adopted for such estimations in numerous projects (Kouskoulas and Koehn 1974, Bowen and Edwards 1985, Khosrowshahi and Kaka 1996). In the nineties, methods based on artificial intelligence were progressively considered. For instance, McKim (1993) proposed a method based on *case-based reasoning* (CBR) and leveraged data from previous projects to forecast costs of similar subsequent projects. Interestingly, this method aims to mimic the reasoning process carried out by humans to obtain this estimation. Practitioners also considered *neural networks* (NN) as a black-box tool for this prediction (McKim 1993, De la Garza and Rouhana 1995, Li 1995). Even at that time, concerns about the lack of transparency were already pointed out. Kim et al. (2004) conducted an analysis to compare performances of a CBR-based method, a linear regression and another one neural network predictor. Although the results showed that the neural network achieved the best overall performances, they concluded that the CBR-based method was more suitable for a practical use, since the performance was not significantly lower and has the benefit to be transparent and easier to use.

Notwithstanding this concern, neural networks have been identified by the survey of Barakchi et al. (2017) as the second most popular tool for cost estimation methods in transport infrastructure. The first position is held by parametric methods based on regression analysis. Nowadays, neural networks are still one of the favourite choices for cost estimations (Mahalakshmi and Rajasekaran 2019, Aretoulis 2019, Roxas et al. 2019, Tijanić et al. 2020). It is important to mention that the focus of these analyses was to obtain accurate estimations while saving time related to preprocess the data and without paying attention to the transparency of the predictions.

Besides, the drawbacks of neural networks for this task were also highlighted in recent studies (Kim et al. 2022). One of them is the tendency of deep neural networks to overfit

the data used to train the model. Briefly, the model is able to provide good accuracy on the training data but struggles to generalize to new situations. Overfitting is a well-known issue in the machine learning community and is likely to happen when large models (such as neural networks) are trained on a small dataset. This situation is common in the construction industry, where only limited historical data are available. To tackle this issue, hybrid methods, combining both neural networks and lighter models, were proposed (Xiong et al. 2019, Petrusheva et al. 2019). As a concrete example, a linear regression can be combined with a neural network before predicting the final cost (Kim et al. 2022). The linear regression is dedicated to limit overfitting as it is a simple model, whereas the neural network is dedicated to improve the accuracy.

There exist also in the literature other approaches for cost prediction. Elmousalami (2020) has listed roughly 20 different tools to carry out such an estimation in construction projects. One conclusion was that *XGBoost* (Chen and Guestrin 2016) is identified as the approach giving the best performances in realistic settings. A similar approach, *CatBoost* (Prokhorenkova et al. 2018), has also obtained excellent performances in estimation related to the field of finance (Jabeur et al. 2021).

Other key aspects of this problem are the data management and the evaluation process of the models. Most of the related works proceed as follows. First, the dataset containing all the historical data is split into two sets, a *training set*, used to train the model, and a *test set*, used to evaluate the final performances. There is also eventually a third set, the *validation set*, used to compare different models and to tune hyperparameters. The division is usually carried out randomly. However, an important aspect of data related to construction projects is that they are often collected at different days, weeks, months, or even years. A direct implication is that a model trained with data collected in 2023, may not be relevant to predict the cost of projects that will occur few years later. This issue is commonly known as a *distributional shift*, data used for the training becomes progressively obsolete. To address this concern, Kim et al. (2022) and Mahdavian et al. (2021) recommend considering the year in which the data was collected as a factor for appropriately partitioning the dataset into training and validation sets. Instead of a random splitting, they use the oldest data to train the model, and the newest data to evaluate it. This is done in a cross-validation fashion. By doing so, the final performance reflects more accurately the fact that the model will be evaluated for future data, which is what

is intended in practice. Interestingly, Mahdavian et al. (2021) observed that linear models performed better than neural networks within this setting.

In summary, our analysis of related works indicates that no single model consistently delivers the best performance across all scenarios, despite all being geared towards construction cost estimation. Models commonly used are either linear regressions, neural networks, XGBoost, CatBoost, or a combination of them. In the machine learning community, *decision trees* (Quinlan 1986) and *random forests* (Breiman 2001) are two other popular choices. This motives our contribution to propose an approach based on all of these models in order to obtain the best estimation as possible. Nevertheless, a direct drawback of such an architecture is its substantial reduction in transparency and explainability of predictions, given the involvement of multiple models. Addressing this transparency deficit is essential in our scenario. The following paragraphs explore recent approaches aimed at enhancing explainability in machine learning.

Explainable Machine Learning

Most of the models introduced earlier were not primarily designed to be transparent. Hence, it is advisable to explore supplementary strategies for addressing this deficiency. Following the artificial intelligence terminology, the term *explainability* refers to the capacity that an algorithm has to describe its logic in a way that is understandable by a human person (Xu et al. 2019). Intuitively, some approaches are, by design, explainable. For instance, decision trees have a simple structure, on which a prediction corresponds to a sequence of simple rules. However, it is not the case of most approaches, and it is why the approach is often improved by a module dedicated to improve the explainability level. There exist two main families of methods for that.

The first category gathers methods that are *model-specific*. They are entirely related to a specific learning approach and cannot be used for another one. For instance, Frosst and Hinton (2017) introduced a principle known as *distillation*, which aims to replicate the behaviour of a deep neural network using a simpler and more explainable model. This process, however, often results in a loss of accuracy. Similarly, Vidal and Schiffer (2020) proposed a method to convert random forests into a single decision tree. Interestingly, this approach does not incur a loss of accuracy, but it is specifically applicable only to random forests.

The second category encompasses *model-agnostic* methods, which, by definition, can be applied independently of the specific model used. These methods typically analyze the model's output directly. Among them, *Shapley Additive Explanations* (Lundberg and Lee 2017) have garnered significant interest within the research community. This approach unified existing methods, such as LIME (Ribeiro et al. 2016) and DeepLift (Shrikumar et al. 2017), towards a unique framework for interpreting predictions. The key idea is to measure accurately the importance of each feature and evaluate their impact on the predictions obtained. Melançon et al. (2021) have successfully used this technique to explain and predict service-level failures in supply chains. However, concerns and limitations have also been raised about this method. The explanation provided relates to the predictions, but not to the model itself, which remains a black-box. Rudin (2019) has a clear opinion on this and advocates stopping explaining black-box machine learning models and use interpretable models instead when high-stakes decisions are involved. Well-known examples of self-explanatory models are decision trees, as presented previously. However, such simple models often do not compete with expressive models in terms of prediction accuracy.

As our approach to estimate construction cost will be structured about several models, a model-agnostic explanation method is required. We propose to use *Shapley additive explanations*, following the excellent explanations that have been obtained with this method. To address the concerns highlighted by Rudin (2019), we also validate the results in collaboration with field experts.

In an industrial context, questions of trust in machine learning approaches are becoming increasingly important as these tools are planned to be used for high-stakes decisions. For such a reason many industrial practitioners are looking for models that are not only accurate, but also trustworthy. Being able to explain the behaviour of a model is going one step closer to this requirement. For instance, Rudin and Ustun (2018), motivated by healthcare and criminal justice applications, highlighted that there exist technologies based on mixed-integer and nonlinear programming to build transparent machine-learning models that are often as accurate as black-box machine-learning models. In the metallurgy field, Salles Melo Lima et al. (2021) leveraged machine learning algorithms to predict the production levels of pig irons on a charcoal blast furnaces at different configurations. Explanations are obtained thanks to a sensitivity analysis. Explainable approaches have been also considered for maritime event recognition (Yeo et al. 2019) and numerous other applications (Carletti et al. 2019).

Problem Definition and Related Data

This paper proposes an end-to-end framework to estimate the cost related to different tasks of a construction project. The estimation should be both accurate and transparent. A *construction task* is defined by a set of *features*. Let $x = \{x_1, \ldots, x_n\}$ be a set of n features associated to a specific task, and let $y \in \mathbb{R}$ be the real cost associated to this task. Formally, the goal is to build a function $f : x \to \mathbb{R}$ giving an estimate $\hat{y}$ of the cost. The *accuracy* for a task x is a measure of how $\hat{y}$ is close to the real cost y, and the *transparency* relates to the capacity to link which features $\{x_1, \ldots, x_n\}$ have contributed the most to the prediction $\hat{y}$.

Construction Tasks Considered

Eight construction tasks, carried out in a regular basis by the Ministry of Transport and Sustainable Mobility are considered in this paper. All of them relate to civil engineering projects and have a specific amount of historical data, corresponding to a specific call for tenders. They are summarized in Table 1. For instance, Task 8 relates to the general organization of the construction site. Each construction task t has its own dataset ($\mathcal{D}_t$) and is composed of a set of historical data $D_t : \{(x^{(i)}, y^{(i)}), \ldots, (x^{(n)}, y^{(n)})\}$ collected from 2014 to 2021. The features and the cost associated to each task are described subsequently. As we can see, the number of data highly depends of the specific task, from 675 to 2596 samples. This motivates the need to consider several models as the performance of a machine learning algorithm depends on the number of data available. Specific prediction models will be trained for each construction task.

Identifier	Construction task	Number of historical data
1	Surface course	2596
2	Base course	1819
3	Armature	936
4	First class excavation	675
5	Second class excavation	1733
6	Pavement sub-base	1029
7	Pavement base	1191
8	Construction site organization	1667

Table 1 Summary of the construction tasks considered in this paper, with the number of historical data.

Feature Description

Each historical data is represented as a set of features. An exhaustive list of the features considered is proposed in Table 2. We can observe that the collected features encompass technical aspects of the task, information about the time period, and details pertaining to the economic situation. Data has been either collected from the call for tenders, from the submissions of contractors, or from external sources such as the *Centre d'Analyse et de Suivi de l'Indice Québec* (CASIQ) database to obtain the economic index of Quebec. Integrating these features and analyzing their relevance for the cost estimation have been done in collaboration with field experts. Intuitively, some features are not relevant for specific tasks. For instance, there is no need to consider the distance to the asphalt plant if there is no asphalt involved in the task. For such a reason, each task from Table 1 only used a subset of the features from Table 2. Features in the table that are present in each task are annotated with a star (*).

Computing Historical Costs

The construction cost $y^{(i)}$ serving as ground truth for a sample $x^{(i)}$ is an information which is not trivial to obtain. On one hand, the construction cost is directly associated with the price quoted by the contractor chosen to provide the service. However, there is no guarantee that the lowest bid accurately reflects the true cost of construction. This discrepancy poses a risk that the model may learn from potentially misleading information if it is trained solely on data from the lowest bids. To mitigate this risk, we propose using information from the three lowest bids instead of relying exclusively on the lowest bidder. This approach helps provide a more robust and representative dataset to train the model.

Each time a call for tenders has been issued, several contractors have the opportunity to propose a bid with a price for supplying the service. Let C be the set of contractors applying for a specific call for tenders. For each call, the Ministry ranks the different contractors following the price they propose. Let $\langle p_1, \ldots, p_C \rangle$ be the sequence of all the prices sorted increasingly by their price. The contractor with the lowest bid is ranked first, and so on. The construction cost $y^{(i)}$ used as reference to train the models is defined as the average unit price proposed by the three lowest bidders, according to the selection policy of the Ministry. It is formalized as $y^{(i)} = (p_1^{(i)} + p_2^{(i)} + p_3^{(i)})/3$. Provided with this information, we have 8 datasets that can be used to train the models.

Feature	Domain type	Description
Quantity	integer	Quantity of material estimated in the submission
Subtask type*	categorical	Specific subtask (reparation, construction, etc.)
Year*	categorical	Year when the call for tenders has been issued
Month*	categorical	Month when the call for tenders has been issued
District*	categorical	District responsible for the project
River side*	categorical	North/South/Junction of St. Lawrence River
Latitude*	integer	Latitude of the construction site
Longitude*	integer	Longitude of the construction site
Asphalt type	categorical	Type of the asphalt used (ESG-10, ESG-14, etc.)
Galvanized	binary	Has galvanized steel been used or not
Granular material	categorical	Type of granular material used (gravel, etc.)
Culvert	binary	Need of culvert, or not, for the task
Reparation	binary	Presence of keywords related to *reparation*
Construction	binary	Presence of keywords related to *construction*
River	binary	Presence of keywords related to a *river*
Coating	binary	Presence of keywords related to *coating*
Security	binary	Presence of keywords related to *security*
Coating	binary	Presence of keywords related to *coating*
Slope	binary	Presence of keywords related to *slope*
Bitumen cost	real	Cost of the bitumen
Employee wages*	real	Average wages of the employees
Economic index*	integer	Value of the IQ-30 index
Asphalt plant	real	Time to reach the closest asphalt plant

Table 2 Summary of all the features considered for the different tasks.

The rationale behind considering the three lowest bidders is to mitigate the potential for abnormally low prices on a particular pay item by leveraging competitive advantages that a bidder might possess. This approach is aimed at estimating prices that are equitable for both the contractor and the public organization. In some cases, the lowest bidder may not reflect a fair price. Additionally, the lowest bidder might have lowered their prices to secure a particular job for various reasons, rendering their prices unrepresentative of

the true cost or market value. The practice of using the three lowest bidders to assess the estimate's performance aligns with common industry standards and is employed by various organizations, including the *American Association of State Highway and Transportation Officials* (AASHTO) and the *US Department of Transportation-Federal Highway Administration* (FHA).

Learning Algorithms

Our contribution leverages the respective strengths of six different training algorithms. As highlighted in the literature review, there is no consensus about a model providing the best performances for any situation. Besides, considering different models strengthens the robustness against the diversity we have on the number of historical data (Table 1). Brief descriptions of the algorithms we considered are provided below. For a more comprehensive understanding, including mathematical formalizations, please refer to the detailed information available in the online supplementary materials.

Decision Tree

Introduced by Quinlan (1986), a *decision tree* carries out predictions by assessing the value of a specific feature sequentially. An illustration of the decision process for a toy sample is proposed in Figure 1. Assuming a call for tenders represented by the features $\{x_{\mathsf{quantity}} = 20, x_{\mathsf{type}} = \mathsf{reparation}, x_{\mathsf{year}} = 2018\}$, the unit construction cost $\hat{y}$ is estimated at \$8. This corresponds to the second leaf in the tree, obtained as the quantity is lower than 50, and as a reparation is involved in the task. There exist several algorithms to compute such a tree from historical data, such as C4.5 (Quinlan 1993), or, more recently, DL8.5 (Aglin et al. 2020).

Random Forests

Random forests (Breiman 2001) are a combination of several decision trees that were independently trained on various sub-samples of the initial dataset. By design, random forests are able to learn more complex relationships in the data, but are harder to explain. It is also more robust against outliers. The training process is explained in detail in the initial paper (Breiman 2001).

Gradient Boosting: XGBoost and CatBoost

Like random forests, *gradient tree boosting* methods (Friedman 2001) rely on a collection of decision trees, albeit with a distinct approach. Instead of taking the average prediction

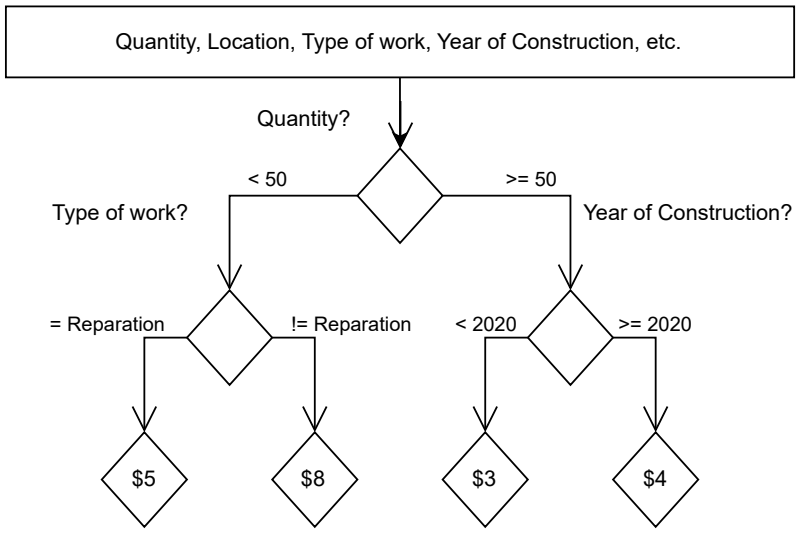

Figure 1 **A simplified representation of a decision tree, applied to our problem.**

of complete decision trees, gradient tree boosting takes the sum of a set of partial decision trees. The first tree gives a rough estimation of the cost, and the subsequent trees increase or decrease the latter prediction in order to improve its accuracy. This process is illustrated in Figure 2. Assuming a call for tender represented by the features $\{x_{\text{quantity}} = 20, x_{\text{type}} = \text{art}, x_{\text{year}} = 2021\}$, the unit construction cost $\hat{y}$ is estimated at \$3. From the first tree, the first estimation is $-$\$2 (quantity greater than 20, and year after 2020), which is then corrected by $+$\$5 from the second tree (no reparation in the task).

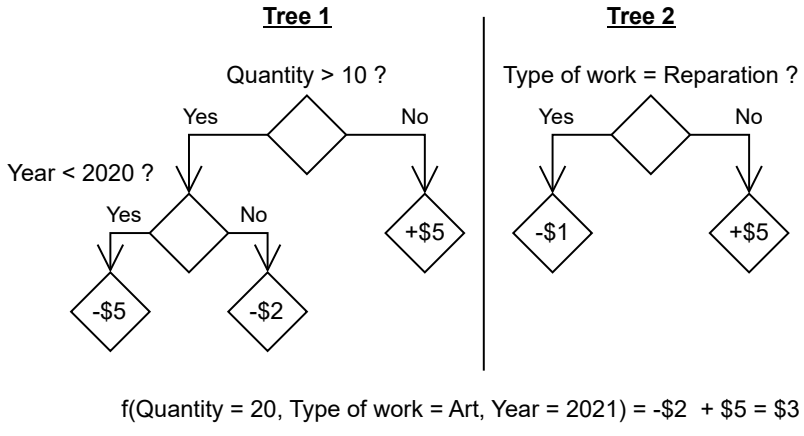

Figure 2 **A simplified and fictive representation of a gradient tree boosting model.**

XGBoost (Chen and Guestrin 2016) and *CatBoost* (Prokhorenkova et al. 2018) are two popular gradient tree boosting algorithms and mainly differ on how the training is carried out. For instance, CatBoost only builds symmetric trees, where XGBoost leverages asymmetric trees. Also, CatBoost supports categorical data, whereas a transformation to numerical data is required for XGBoost. A review of the differences and similarities between both algorithms is proposed by Bentejac et al. (2021). Finally, there also exists other similar algorithms such as *Lightgbm* (Ke et al. 2017).

Neural Network

Popularized by deep learning (LeCun et al. 2015), a *neural network* is a non-linear function, parameterized by a set of *neurons* which are organized in a sequence of *layers*. Briefly, each neuron computes its own prediction, and these are then combined to form the final prediction. This process is illustrated in Figure 3 for a simple neural network of 2 layers.

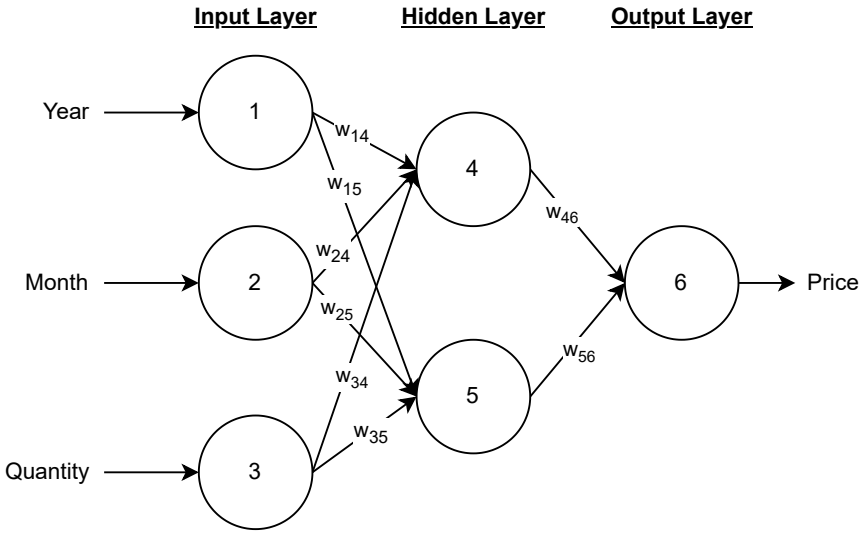

Figure 3 A simplified representation of a 2-layers neural network containing three features.

The input layer correspond to the features given as input. In this example, the neural network takes an input $\{x^{(i)}_{\mathsf{year}}, x^{(i)}_{\mathsf{month}}, x^{(i)}_{\mathsf{quantity}}\}$, and gives as output the estimated construction cost $\hat{y}^{(i)}$. Mathematically, each hidden neuron consists in a linear combination of its input, transformed by a non-linear function. The rectified linear unit ($\mathsf{ReLU}(x) = \max(x, 0)$) is often considered for that (Glorot et al. 2011). Training such a network is commonly carried out with back-propagation (Rumelhart et al. 1986) together with a stochastic gradient

descent algorithm, such as Adam (Kingma and Ba 2014). Additional mechanisms such as weight initialization (He et al. 2015) or dropout (Srivastava et al. 2014) are used in practice to improve the performance of the learning.

Ensemble Voters

Finally, our last model employs a technique known as *ensemble voting*, commonly recognized for its efficacy in boosting predictive accuracy (Dietterich 2000). The underlying principle involves aggregating the outputs of the five preceding models to enhance the overall quality of the predictions. Typically, this aggregation is performed by calculating the median of the predictions from each model, which helps to achieve a more robust and reliable final prediction.

Data-driven Prediction Pipeline

This section presents the data-driven prediction pipeline that we have designed to train models dedicated to estimate the unit cost of construction tasks. An overview of the architecture is proposed in Figure 4. The pipeline is divided into six steps. The first step must be carried only once for each dataset whereas the other steps are performed individually for each model. Additionally, we propose a maintenance step to ensure the model is systematically updated over the years.

Step 1: Data Preparation

This step is carried out for the 8 datasets identified in Table 1 (i.e., a specific construction task) and mainly consists in preparing the data prior its use for training a model and evaluating them. Only two operations are performed: (1) each data sample with missing values is removed, (2) each categorical features in Table 2 is encoded with a one-hot representation. For instance, assuming there are three types of subtasks (reparation, construction, demolition), the representation of each subtask is as follows: reparation $\rightarrow [1,0,0]$, construction $\rightarrow [0,1,0]$, and demolition $\rightarrow [0,0,1]$. All the models are then trained and evaluated from these datasets.

Step 2: Data Scaling

This step involves scaling data values to facilitate the training process and enhance model performance. It is important to note that unlike the first step, a specific step (and all the subsequent ones) must be carried out for each model type. The reason is that each learning algorithm has a different sensitivity to this operation. As a concrete example,

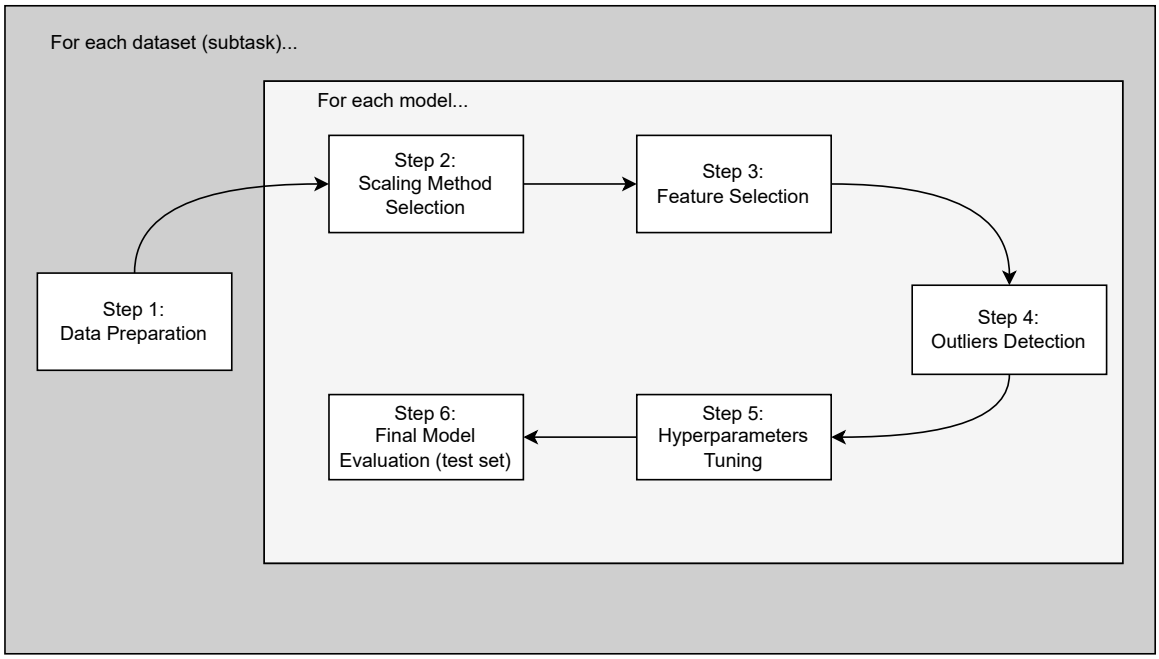

Figure 4 Overview of the data-driven prediction pipeline (6 main steps).

the training phase of a neural network often requires data normalization, which is not the case of decision trees. Scaling is applied exclusively to numerical features (integers or real numbers). We explore several scaling options, with comprehensive details provided in the online supplement.

1. NoScaling: the values of each feature are not modified.
2. Normalization: this operation consists in shrinking the domain of each value between 0 and 1.
3. Standardization: this operation consists in modifying the values of each feature in order to have them centered in zero and with a standard deviation of 1.

The selection of the scaling scheme is carried out in Step 6 through a cross-validation procedure. For each trained model, the scheme yielding the best performances is then selected.

Step 3: Feature Selection

The quality of a model highly depends on the input features. Whereas the common trend is to add as many features as possible and let the model discover by itself what are the most relevant features, having too many features may negatively impact the performances (e.g., when two features are colinear or are irrelevant to the prediction task). For such a reason

we added a feature selection procedure to the pipeline. Concretely, only a subset of the features identified in Table 2 is used for training a model. A popular choice is the *recursive feature elimination* (RFE) algorithm (Guyon et al. 2002). Briefly, this algorithm works as follows: (1) a model is trained using all the features, (2) a measure of the importance of each feature is computed and the least relevant is eliminated, (3) the procedure is repeated with one feature less, until a predetermined minimal number of features is reached (8 in our implementation). The exact number of features to keep is determined by the cross-validation process. For this task, we used the implementation provided by scikit-learn (Pedregosa et al. 2011). We highlight that the main goal behind this feature selection was to improve accuracy. However, as a beneficial side-effect, removing features from the input also tend to increase the explainability level of the models.

Step 4: Outliers Detection

When dealing with real data, it is possible that some samples are affected by experimental errors, such as erroneously encoding handwritten numbers in a spreadsheet. While modern learning algorithms exhibit a degree of robustness to outliers, these misleading samples can have a detrimental effect on performance, especially when the dataset is limited. To address this concern, we mitigate the issue by incorporating an *outliers detection* phase into our pipeline. Three automatic options to detect outliers are considered (plus the option to remove nothing) and used thanks to the implementation provided by scikit-learn (Pedregosa et al. 2011).

1. IsolationForest (IF): the idea is to isolate data points by removing few features of the dataset. Then, data points that are easy to isolate are categorized as outliers (Liu et al. 2008).

2. LocalOutlierFactor (LOF): each data point is characterized by its neighborhood (i.e., a set of other data points in the dataset with relatively close values). Provided with this information, a measure of how isolated a data point is with respect to its neighbourhood is computed. Data points with a high isolation score are then removed (Breunig et al. 2000).

3. OneClassSVM (OCSVM): this method is a variation of *support vector machines* (SVM) (Hearst et al. 1998) that can be used in an unsupervised setting for novelty detection, and by extension, for outliers detection (Zhang et al. 2007).

4. None: no outlier has been removed from the data.

Step 5: Hyperparameters Tuning

This step is dedicated to tune the hyperparameters of each model (e.g., the number of layers of a neural network) to find an efficient combination among the different alternatives. It has been done first by taking directly the default values for some hyperparameters, then by a grid-search for a restricted set of remaining hyperparameters. We would like to emphasize that one of the hyperparameters under consideration is the choice of loss function, which can either be the mean absolute percentage error or the mean squared error.

Step 6: Final Model Evaluation

Previous steps generated different options for each learning algorithm. This final step is now dedicated to select the most efficient option for each of them. It is done through a k-folds cross-validation procedure, with the specificity to take into account the temporal nature of the data, as proposed by Mahdavian et al. (2021). As a reminder, our historical data has been collected from 2014 to 2021. The validation procedure consists in splitting the dataset into 7 subsets (one per year). Each fold (i.e., a pair including a training and validation set) is built by taking the subset of a specific year for the validation, and the subsets of all the previous years for the training. Finally, the training subsets must have at least three years in order to have enough data to train the models. In our case, this procedure yields 4 folds and is illustrated in Figure 5. The folds obtained are formalized in the online supplement. The model having the best average performance on the four folds is then selected for the test phase. Year 2021 is kept outside this procedure and is used only as a test set for the final performances.

Finally, the metric used to evaluate performance is the *mean absolute percentage error* (MAPE). This metric is known to be simple to interpret and has been considered in related works (Mahalakshmi and Rajasekaran 2019, Xiong et al. 2019, Tijanić et al. 2020). It is formalized in the online supplement.

Maintenance Step: Updating the Model

Consider the application of our model for the year t. For training, we use data collected up to year $t-2$, while data from year $t-1$ is solely for validation purposes, as depicted in Figure 5. As we transition to year $t+1$, we propose to update the training dataset to include data from year $t-1$ and use the data from year t for validation. To ensure the relevance of the training data, we may also set a historical limit, beyond which data is

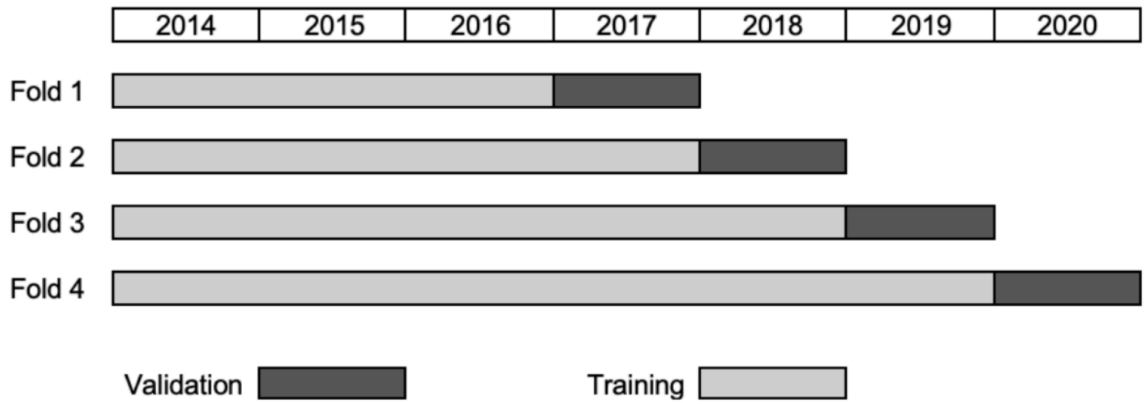

Figure 5 **Illustration of the temporal cross-validation for our dataset.**

considered outdated and thus excluded from the training process. This iterative updating methodology ensures that our model adapts to change and aims for robustness over time.

Evaluation of Performance

The goal of this section is to empirically validate the efficiency of our approach on new data. Two experiments are proposed: (1) an evaluation of the six learning algorithms described previously and tuned thanks to our data-driven prediction pipeline, and (2) a comparison with the performance of a human estimation, as currently done by specialists from the Ministry. Prior to this analysis, the experimental protocol and information about the training phase is proposed.

Experimental Settings

Eight tasks have been identified in Table 1 and six models (decision tree (DT), random forests (RF), XGBoost (XGB), CatBoost (CatB), neural network (NN), and ensemble voters (EV)) are considered for predicting the cost of each task. This yields a total of 48 models to consider, with several possible configurations for each of them. The data-driven prediction pipeline has been executed for all the options and a final model with its own configuration is obtained for each task. Such models are the ones having the best performance during the cross-validation phase. They are summarized in Table 3. We note that because of the one-hot encoding, the number of features is higher than the ones provided in Table 2.

Interestingly, our observations indicate that no single algorithm or configuration (aside from scaling, which appears unnecessary) consistently yields the best results across all

Construction Task	**Model**	**Scaling**	**# Features**	**Outliers detection**
Surface Course	EV	NoScaling	30/74	IF
Base course	CatB	NoScaling	18/71	None
Armature	RF	NoScaling	40/60	LOF
First Class Excavation	XGB	NoScaling	47/61	LOF
Second Class Excavation	XGB	NoScaling	50/67	OCSVM
Pavement Sub-base	EV	NoScaling	19/62	LOF
Pavement Base	EV	NoScaling	31/64	LOF
Construction Site Organization	NN	NoScaling	42/67	OCSVM

Table 3 Best model with its configuration for each construction task. The information x/y indicates that x features among y are kept.

subtasks. This finding supports the widely recognized principle that model performance is highly dependent on the specific task and the volume of available data. It also underscores the value of our pipeline in identifying the optimal configuration for each scenario. Each configuration has been trained on a computer provided with a 2.6 GHz Intel Core i5 processor and 8 GB of RAM, and without GPUs. Table 4 recaps the typical training time (in seconds) of a model for each task. We can observe that training a model roughly takes between 1 second for the simplest models (decision tree) and 30 minutes for the most complex (neural network). This is relatively fast and it explains why we can allow the pipeline to test a large number of configurations.

Construction Task	DT	RF	XGB	NN	CatB	EV
Surface Course	<1	1	<1	2	230	513
Base Course	<1	1	1	549	2	394
Armature	<1	<1	<1	1840	2	570
First Class Excavation	<1	<1	<1	87	4	507
Second Class Excavation	<1	<1	<1	370	10	430
Pavement Sub-base	<1	<1	<1	762	1	297
Pavement Base	<1	<1	<1	563	2	834
Construction Site Organization	<1	<1	<1	290	1	220

Table 4 Time (seconds) required for training each selected model.

Experiment: Performance of the Best Models

In this first experiment, the best configurations for each training algorithm are compared on the data related to year 2021. This situation corresponds to a realistic scenario because such data was not seen during the training nor the validation. The results are summarized in Table 5 using the mean absolute percentage error (MAPE) as evaluation metric. The best results are highlighted and correspond to the models recorded in Table 3. Further to the observation that no single model consistently outperforms others across all tasks, it is notable that neural networks generally show poor performance, except in the task of *construction site organization* on which all methods tested yield similarly inaccurate results. Field specialists suggest that the challenge of this task stems from its broad impact on the entire civil engineering project, making it more difficult to estimate accurately compared to other tasks that focus on more localized aspects of a project.

Construction Task	DT	RF	XGB	NN	CatB	EV
Surface Course	22.37	22.64	22.60	30.91	29.69	**20.63**
Base Course	25.68	19.62	20.87	40.59	**19.57**	20.83
Armature	32.10	**26.69**	30.76	40.98	27.81	31.22
First Class Excavation	77.10	60.53	**52.20**	339.58	72.64	55.13
Second Class Excavation	44.63	39.38	**37.04**	39.64	37.33	38.89
Pavement Sub-base	26.00	21.02	20.37	55.07	20.21	**19.96**
Pavement Base	20.36	17.76	17.15	47.89	17.14	**16.96**
Construction Site Organization	103.23	113.82	83.34	**82.15**	132.06	109.07
Average performance	43.93	40.18	**35.54**	84.60	44.55	39.09
Median performance	29.05	**24.67**	26.68	44.44	28.75	26.03

Table 5 MAPE (in percentage) computed on data from year 2021 for each model and for each construction task. Best results are highlighted.

Experiment: Comparison with Expert Estimations

The next step involves validating the predictions made by our models against the estimations provided by experienced experts from the Ministry of Transport and Sustainable Mobility with previous calls for tenders. Before a call for tenders is launched, these specialists estimate a cost for each subtask requested in the call for tenders. The experts

have information about the construction project concerned by the call. They also have access to descriptive statistics about the cost of subtasks as billed in previous contracts, filtered to be relevant to the contract at hand. The results are summarized in Table 6. Except for the *construction site organization* which has proven to be a complex task for estimation, the machine learning models provide more accurate estimates of the costs of construction projects compared to human estimations. This empirically validates the benefits of employing machine learning approaches for preliminary cost estimations in civil engineering projects.

Construction Task	**Model**	**Performance**	**Expert Performance**
Surface Course	EV	**20.63**	22.63
Base course	CatB	**19.57**	21.47
Armature	RF	**26.69**	39.00
First Class Excavation	XGB	**52.20**	64.79
Second Class Excavation	XGB	**37.04**	38.35
Pavement Sub-base	EV	**19.96**	28.25
Pavement Base	EV	**16.96**	22.08
Construction Site Organization	NN	82.15	**43.35**

Table 6 MAPE (in percentage) computed on data from year 2021 for each best model and for the estimation provided by field experts.

Explaining Predictions with Shapley Additive Explanations

The previous section showed that accurate estimations could be obtained for different construction tasks. However, the requirement of *explanability*, i.e., to be able to understand what are the root causes of a prediction, is still unaddressed. As no unique model has been identified, the method used to obtain explanations must be able to accommodate to different learning algorithms. We propose to tackle this challenge thanks to *Shapley Additive Explanations* (SHAP), introduced by Lundberg and Lee (2017). Briefly, Shapley values have been initially developed by Shapley (1953) for cooperative game theory. The idea is to measure the contribution of each player for the accomplishment of a goal and to reward them accordingly to their own contribution. A Shapley value represents the *marginal contribution* of a player. Lundberg and Lee (2017) proposed to leverage Shapley

values to interpret predictions of machine learning algorithms. In this context, the goal is to compute a Shapley value for each *feature* and to obtain a measure of their contribution for the prediction obtained by the model. A feature with a high score could then be interpreted as paramount for the prediction. Their computation is formalized in the online supplements.

In our case, the features in Table 2 will be associated with a Shapley value for each construction task. The final prediction can be expressed by means of Shapley values. In other words, a prediction for a sample i corresponds to the average prediction plus the contribution of each feature, weighted by their Shapley value. It is a linear function, which is easier to explain. An efficient way to compute the Shapley value has been provided by Lundberg and Lee (2017) and is open-sourced[1]. We used this implementation for generating the figures presented in this section.

Analysis: Local Explanations

This first analysis aims to analyze the prediction carried out for a specific sample related to a construction task. It is commonly referred to as a *local explanation*. One example is illustrated in Figure 6 for an arbitrary sample related to *armature* construction task.

Such a plot is better analyzed from the bottom to the top. First, we observe that the average prediction of f for this task is a unitary cost of \$6.797. Then, each row shows the marginal contribution of each feature on the average cost, either as a decreasing factor or as an increasing factor. For instance, we can observe that this specific longitude (corresponding to the west of Montreal Metropolitan network) has significantly increased the cost, such as the fact that a reparation task is involved. On the other hand, the fact that the project has been issued in 2016 (least recent year in our dataset) decreases the average cost, which makes sense in light of the annual inflation. Such an analysis has a direct interest for the project managers to understand what are the root causes of a specific prediction. A similar plot can be obtained for each other combination of tasks and samples.

Analysis: Global Explanations

A drawback of the previous representation is, as it considers only one sample, it does not help for comparing the set of values of a single feature. For instance, considering the *quantity* feature, we are unable to know if increasing the quantity will tend to increase or

[1] `https://github.com/slundberg/shap`

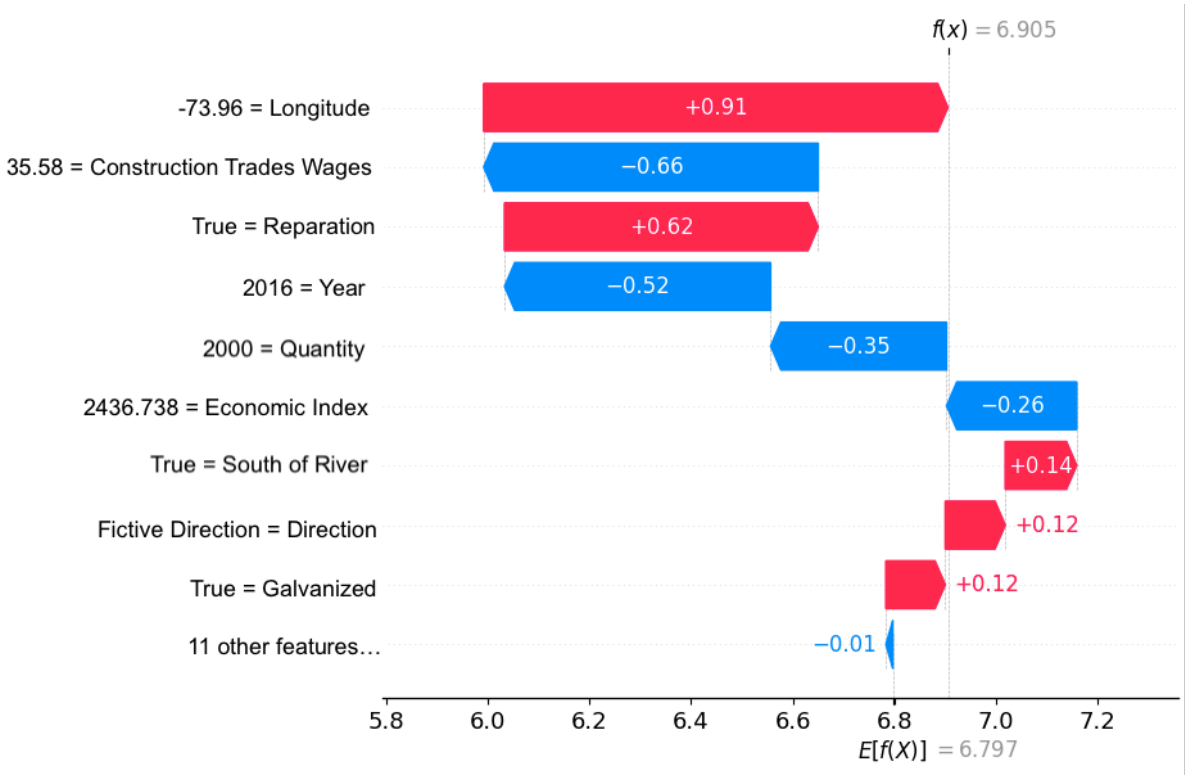

Figure 6 **Local explanation of a specific sample for armature task. The model is based on random forests.**

decrease the construction cost. This issue can be mitigated by *global explanations*, as the one provided in Figure 7 for the *armature* task.

Each row corresponds to a specific feature, and each dot represents a specific sample. Its color indicates the value of the feature (blue being low, and red being high). The x-axis corresponds to the impact of the feature value from the sample on the prediction cost. For instance, the first row (*quantity*) shows that large quantities (red color) tend to decrease the construction cost (on the left of the x-axis). The more the quantity is, the cheaper is the cost (switching to blue values on the right). This directly makes sense in the industrial context, as the predicted cost is *unitary*, which enables economies of scale. Another (not surprizing) information is that the more the employee wage is (*construction trades wages*), the higher is the cost. Such a global explanation can be obtained for each construction task. We finally note that there are other visualization tools that can be leveraged.

Analysis: Insights Provided by Shapley Values

The objective of this section is to illustrate the practical application of insights obtained from Shapley values within actual business operations, particularly in terms of validating

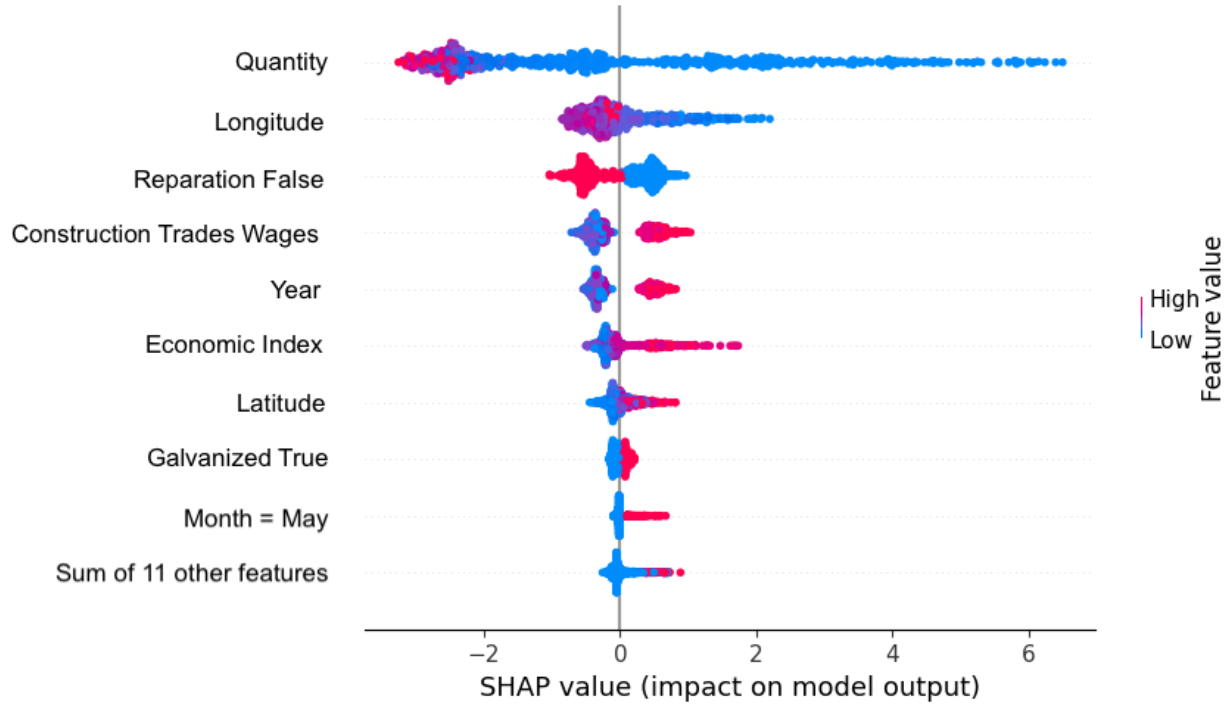

Figure 7 **Global explanation of a specific sample for armature task. The model is based on random forests.**

(or challenging) the intuitions of experts. For this purpose, we analyzed all explanations derived from Shapley values, as depicted in Figure 7, specifically focusing on the armature task. We then asked a field expert to review each explanation individually and evaluate their significance. A summary of the expert's feedback is presented below.

1. Larger Quantities Reduce Costs. The observation that larger quantities tend to decrease costs is attributed to economies of scale in material ordering. We point out that the costs reported are unitary.

2. East-most Longitudes and Cost Reduction. Contrary to expert intuition, it was found that projects in east-most longitudes tend to have lower costs, despite iron providers being predominantly located in the south-west. This intriguing finding encourages further investigation to understand the underlying reasons.

3. Repairs Increase Costs. Including repair work in a task naturally leads to higher costs due to the increased workload.

4. Impact of Higher Wages. As expected, higher wages for employees contribute to increased project costs.

5. Recent Years Show Higher Costs. The trend of rising costs in more recent years can be explained by annual inflation.

6. IQ-30 Index and Cost Correlation. A high index, indicating a robust economy in the province, correlates with increased costs, which aligns with expectations.

7. North-most Latitudes and Higher Costs. Projects located in north-most latitudes tend to be more expensive, which is consistent with expert intuition given that iron suppliers are mostly in the south-west, making transportation longer and more costly.

8. Galvanized Steel and Cost Increase. The use of galvanized steel, being a pricier material, understandably increases the cost of tasks.

9. May Issuance and Cost Increase. Projects initiated in May tend to be costlier, explained by the timing of construction activities peaking in summer. This requires months of advance planning that typically begins in May.

An analogous analysis was conducted for the pavement sub-base category and is provided in the online supplements. Interestingly, this examination of Shapley values yielded two non-trivial insights, highlighting the nuanced impact of certain factors within this specific category.

1. West-most Longitudes and Increased Costs. Regions of such longitudes are characterized by either a lack of competition, resembling oligopolistic markets, or high road traffic areas like large cities, which hampers construction site productivity. Both scenarios naturally lead to elevated costs.

2. North-most Latitudes and Decreased Costs. Conversely, north-most latitudes are linked to lower project costs. These areas often enjoy healthier competition among service providers, which serves to drive costs down.

Additionally, observations similar to those made for the previous task have also been noted, indicating consistent patterns across different project categories. We are confident that this analysis provides experts with novel insights, revealing information not readily discernible through conventional methods like decision trees or linear regression. Furthermore, this analysis is valuable to project managers by enabling them to justify task pricing to stakeholders, thereby enhancing the transparency of cost estimations.

Analysis: Feedback from the Field Experts

To better assess the practical implications of Shapley values, we conducted a survey involving experts who regularly perform cost estimations for the Ministry. This section outlines

the methodology used for the survey and summarizes the key findings. We randomly selected two historical scenarios for each construction task as outlined in Table 1. For each of these scenarios, we developed three explanation tools: one based on decision trees, one on linear regression, and the last one on Shapley values. Examples of these explanations are provided in the online supplements. The survey encompassed a set of questions, which were asked to the participating experts:

1. Did the explanation provided by Shapley values align with your expectations regarding the predicted cost? If not, please explain why. A similar question was asked for the decision tree and the linear regression explanations.

2. Did the explanation provided by Shapley values lead you to discover any surprising insights regarding the predicted cost? If so, please elaborate on the findings. A similar question was asked for the decision tree and the linear regression.

3. Overall, which method do you find most effective in explaining the predicted cost, and with which of these methods would you prefer to work in the future?

In addition to these questions, general comments and feedback regarding the methods were also gathered. We received responses from five field experts, and the primary insights gained from this survey are detailed as follows. To prevent bias in the assessments made by experts, it is important to note that our pipeline's foundation in Shapley values was deliberately undisclosed in the survey.

Decision Trees The experts emphasized the presence of discrepancies between their expectations and the explanations offered by the method. For example, there were cases where the quantity was historically recognized to have a substantial impact on the cost, but the decision trees did not adequately reflect this relationship. One notable advantage that emerged from the survey was that the visualization provided by this method was considered intuitive. However, it was acknowledged that this intuitiveness might become more challenging to grasp when dealing with larger decision trees. Among the five experts, one expressed a preference for working with this method.

Linear Regression The experts noted that this method struggled to capture non-linear dependencies between the features. Additionally, they observed that the predictions were primarily influenced by only a few features with substantial coefficients (as exemplified by the *year* in the example provided in the online supplements), obscuring the significance of other features. None of the experts expressed a preference for working with this method.

Shapley Values One expert highlighted that this tool is non-standard and may entail a learning curve to fully comprehend. It was observed that the quantity appeared to have a lesser impact than expected, while the site localization had a more significant impact. The ability to visualize the marginal impact of each feature on the cost was appreciated by one expert. Notably, two experts expressed a preference for working only with this method.

General Comments In addition to the specific insights, two experts pointed out that adding other input features could enhance the model's performance. In response to the last question, one expert emphasized that the three methods provide complementary information and expressed a desire to work with all of them. However, the last expert chose not to answer this question, as they think that crucial input features were missing, making it challenging to draw a definitive conclusion. In summary, the survey indicates a favourable reception of Shapley values as an explanation tool. It is worth noting that, in terms of prediction accuracy, Shapley values offer greater flexibility in terms of model choice and have the potential to unlock improved performance.

Discussion: Vision for the Next Steps

Over the past five years, authorities in Quebec have shown an increased interest in the quality of cost estimates for construction projects. The results obtained from this research project have prompted the Ministry of Transport and Sustainable Mobility to explore ways to implement the tools developed within this project into the current infrastructure. This tool can provide estimators with a baseline for the various pay items of a project. Estimators will be able to better assess historical costs thanks to the explicability models that were developed.

With around 700 construction projects occurring annually, the integration of machine learning approaches to aid in cost estimation could significantly enhance the efficiency of estimators. This possible integration has generated considerable interest among field experts. This, in turn, would allow them to focus on significant lump-sum items that require more time and experience to estimate accurately. As demonstrated in the study, the model faced challenges with lump-sum items, and further testing appears necessary for the model to produce reliable cost estimates for those items. While the Ministry does not anticipate relying solely on machine learning for future cost estimates, this technology is set to become an additional tool available to estimators to produce fair and reliable estimates, hence ensuring that public funds are allocated appropriately.

To this end, the envisioned contractor selection process is structured as follows: (1) the model generates a cost estimate for a specified task, (2) Shapley values are used to determine which features significantly influenced the cost estimation, and (3) experts leverage this insight, alongside their own expertise, to predict the costs submitted in the upcoming call for tenders. The lowest bid should be selected in the call for tenders, unless that the bid is too different to the predicted costs. The primary purpose of the model is to illuminate features and to augment their decision-making by providing pertinent information that aids in identifying anomalies, such as unusually low costs.

Conclusions

Early stage estimation of costs plays an important role in any construction project and has a significant impact on the decision-making process carried out by project managers, when subtracting subtasks through call for tenders. Ideally, the price put forth by general contractors should precisely represent the actual cost of the task, considering measurable factors like the cost of raw materials. This estimation is typically accomplished through the expertise of professionals and an analysis of historical costs for similar tasks. Based on this context, this paper proposes a data-driven prediction pipeline dedicated for estimating construction costs. The pipeline is based on six standard machine learning algorithms, on several data preprocessing steps, and on Shapley additive explanations. Three requirements were considered: the prediction must be accurate (at least of a same level of quality than a human estimation), the tool must be able to accommodate with different kinds of tasks, and finally the predictions obtained must be explainable. Experimental results and analyses carried out on eight construction tasks showed that these requirements were successfully fulfilled for seven of them. The main results showed that better prediction than the expert estimation could be reached (up to 31.56%) on new submissions, but also that there was no learning algorithm which provides the best results for all the tasks. Finally, the estimations obtained have been validated by specialists from the Ministry.

References

Aglin G, Nijssen S, Schaus P (2020) Learning optimal decision trees using caching branch-and-bound search. *Proceedings of the AAAI Conference on Artificial Intelligence* 34(04):3146–3153.

Arafa M, Alqedra M (2011) Early stage cost estimation of buildings construction projects using artificial neural networks. *Journal of Artificial Intelligence* 4(1).

Aretoulis GN (2019) Neural network models for actual cost prediction in Greek public highway projects. *International Journal of Project Organisation and Management* 11(1):41–64.

Arıkan U, Kranz T, Sal BC, Schmitt S, Witt J (2023) Human-centric parcel delivery at Deutsche Post with operations research and machine learning. *INFORMS Journal on Applied Analytics* 53(5):359–371.

Barakchi M, Torp O, Belay AM (2017) Cost estimation methods for transport infrastructure: a systematic literature review. *Procedia engineering* 196:270–277.

Bentejac C, Csorgo A, Martinez-Munoz G (2021) A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review* 54:1937–1967.

Bowen P, Edwards P (1985) Cost modelling and price forecasting: practice and theory in perspective. *Construction Management and Economics* 3(3):199–215.

Breiman L (2001) Random forests. *Machine learning* 45:5–32.

Breunig MM, Kriegel HP, Ng RT, Sander J (2000) LOF: identifying density-based local outliers. *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 93–104.

Carletti M, Masiero C, Beghi A, Susto GA (2019) Explainable machine learning in industry 4.0: Evaluating feature importance in anomaly detection to enable root cause analysis. *2019 IEEE international conference on systems, man and cybernetics (SMC)*, 21–26 (IEEE).

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

De la Garza JM, Rouhana KG (1995) Neural networks versus parameter-based applications in cost. *Cost Engineering* 37(2):14.

Dietterich TG (2000) Ensemble methods in machine learning. *International workshop on multiple classifier systems*, 1–15 (Springer).

Dodin P, Xiao J, Adulyasak Y, Alamdari NE, Gauthier L, Grangier P, Lemaitre P, Hamilton WL (2023) Bombardier aftermarket demand forecast with machine learning. *INFORMS Journal on Applied Analytics* 0(0).

Elmousalami HH (2020) Artificial intelligence and parametric construction cost estimate modeling: State-of-the-art review. *Journal of Construction Engineering and Management* 146(1).

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Annals of statistics* 1189–1232.

Frosst N, Hinton G (2017) Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784* 0(0).

Gade K, Geyik SC, Kenthapadi K, Mithal V, Taly A (2019) Explainable AI in industry. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 3203–3204.

Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. Gordon G, Dunson D, Dudík M, eds., *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, 315–323 (PMLR).

Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. *Machine learning* 46:389–422.

He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B (1998) Support vector machines. *IEEE Intelligent Systems and their applications* 13(4):18–28.

Hui PP, Davis-Blake A, Broschak JP (2008) Managing interdependence: The effects of outsourcing structure on the performance of complex projects. *Decision Sciences* 39(1):5–31.

Jabeur SB, Gharib C, Mefteh-Wali S, Arfi WB (2021) CatBoost model and artificial intelligence techniques for corporate failure prediction. *Technological Forecasting and Social Change* 166:120658.

Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY (2017) LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems* 30.

Khosrowshahi F, Kaka AP (1996) Estimation of project total cost and duration for housing projects in the UK. *Building and Environment* 31(4):375–383.

Kim GH, An SH, Kang KI (2004) Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning. *Building and environment* 39(10):1235–1242.

Kim S, Choi CY, Shahandashti M, Ryu KR (2022) Improving accuracy in predicting city-level construction cost indices by combining linear ARIMA and nonlinear ANNs. *Journal of Management in Engineering* 38(2):04021093.

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* 0(0).

Kouskoulas V, Koehn E (1974) Predesign cost-estimation function for buildings. *Journal of the Construction Division* 100(4):589–604.

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444.

Li H (1995) Neural networks for construction cost estimation. *Building Research and Information* 23(5):279–284.

Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, 413–422 (IEEE).

Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems* 30.

Mahalakshmi G, Rajasekaran C (2019) Early cost estimation of highway projects in India using artificial neural network. *Sustainable Construction and Building Materials*, 659–672 (Springer).

Mahdavian A, Shojaei A, Salem M, Yuan JS, Oloufa AA (2021) Data-driven predictive modeling of highway construction cost items. *Journal of Construction Engineering and Management* 147(3):04020180.

McKim RA (1993) Neural network applications to cost engineering. *Cost Engineering* 35(7):31.

Melançon GG, Grangier P, Prescott-Gagnon E, Sabourin E, Rousseau LM (2021) A machine learning-based system for predicting service-level failures in supply chains. *INFORMS Journal on Applied Analytics* 51(3):200–212.

Muir WA, Reich D (2021) Using machine learning to improve public reporting on US government contracts. *INFORMS Journal on Applied Analytics* 51(6):463–479.

Nanduri J, Jia Y, Oka A, Beaver J, Liu YW (2020) Microsoft uses machine learning and optimization to reduce e-commerce fraud. *INFORMS Journal on Applied Analytics* 50(1):64–79.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Édouard Duchesnay (2011) Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* 12(85):2825–2830.

Petrusheva S, Car-Pušić D, Zileska-Pancovska V (2019) Support vector machine based hybrid model for prediction of road structures construction costs. *IOP Conference Series: Earth and Environmental Science* 222(1):012010.

Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A (2018) CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems* 31.

Quinlan JR (1986) Induction of decision trees. *Machine learning* 1(1):81–106.

Quinlan JR (1993) C 4.5: Programs for machine learning. *The Morgan Kaufmann Series in Machine Learning* 0(0).

Rai R, Tiwari MK, Ivanov D, Dolgui A (2021) Machine learning in manufacturing and industry 4.0 applications. *International Journal of Production Research* 59(16):4773–4778.

Ribeiro MT, Singh S, Guestrin C (2016) "Why should I trust you?" explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.

Roxas CLC, Roxas NR, Cristobal J, Hao SE, Rabino RM, Revalde F (2019) Modeling road construction project cost in the philippines using the artificial neural network approach. *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 1–5 (IEEE).

Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1(5):206–215.

Rudin C, Ustun B (2018) Optimized scoring systems: Toward trust in machine learning for healthcare and criminal justice. *Interfaces* 48(5):449–466.

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323(6088):533–536.

Salles Melo Lima M, Eryarsoy E, Delen D (2021) Predicting and explaining pig iron production on charcoal blast furnaces: A machine learning approach. *INFORMS Journal on Applied Analytics* 51(3):213–235.

Shan M, Le Y, Yiu KT, Chan AP, Hu Y, Zhou Y (2018) Assessing collusion risks in managing construction projects using artificial neural network. *Technological and Economic Development of Economy* 24(5):2003–2025.

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

Shrikumar A, Greenside P, Kundaje A (2017) Learning important features through propagating activation differences. *International Conference on Machine Learning*, 3145–3153 (PMLR).

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(56):1929–1958.

Tijanić K, Car-Pušić D, Šperac M (2020) Cost estimation in road construction using artificial neural network. *Neural Computing and Applications* 32:9343–9355.

Vidal T, Schiffer M (2020) Born-again tree ensembles. III HD, Singh A, eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 9743–9753 (PMLR).

Xiong B, Newton S, Li V, Skitmore M, Xia B (2019) Hybrid approach to reducing estimating overfitting and collinearity. *Engineering, Construction and Architectural Management* 26(10):2170–2185.

Xu F, Uszkoreit H, Du Y, Fan W, Zhao D, Zhu J (2019) Explainable AI: A brief survey on history, research areas, approaches and challenges. *Natural Language Processing and Chinese Computing: 8th CCF International Conference*, 563–574 (Springer).

Yeo G, Lim SH, Wynter L, Hassan H (2019) MPA-IBM project SAFER: sense-making analytics for maritime event recognition. *INFORMS Journal on Applied Analytics* 49(4):269–280.

Zhang R, Zhang S, Muthuraman S, Jiang J (2007) One class support vector machine for anomaly detection in the communication network performance data. *Proceedings of the 5th Conference on Applied Electromagnetics, Wireless and Optical Communications*, 31–37.