Imperial College Business School

Machine Learning in Construction:

Conceptual and Practical Examination for Data Driven Future

MSc Business Analytics
Imperial College Business School

Robert Arnason CID: 01592337

September 2, 2019

Word Count: 5,500

Abstract

The construction industry is one of the largest economic sectors in the world, employing about 7% of the global working population. Despite this large global impact, the industry has experienced very low productivity growth rate in recent years. This report will address what is causing this lag and present technological solutions to catch up with other industries. Research into recent advancements along with emerging ideas was conducted to gain overview of possibilities. Machine learning methods were implemented to evaluate the feasibility of a few ideas. The results are promising and some methods can greatly improve future industry safety and productivity. There is however a large hurdle to deal with, the complete lack of data collection culture is something that has to be changed. The only way to do that is to make managers and employees within the industry realise what can be gained.

Contents

1 Introduction

The possible applications of machine learning in all industries are bountiful and there is great value to be gained. However, there are some industries who are more open to implementing these new technologies than others. This can be caused by an easier access to data, more financial gain or higher openness for new technologies. The construction industry has many ways to gain from machine learning technologies, through planning, safety and management, among others. There has however been small incentive from construction managers to take advantage of this situation. This can be observed in the fact that the construction industry has had an abnormally low labour-productivity growth over the past two decades. With an average annual growth rate of only 1%, the industry is lagging well behind the global economic growth rate of 2.8%. If this gap is to be closed, the yearly industry value can be raised from an estimated $10 trillion to $11.6 trillion. Barbosa et al. (2017) This promise of immense gains is what makes this a worthy research topic and why the construction industry has caught the attention of many technology companies in recent years.

The main cause of this low productivity is how under-digitised the industry is and how slowly it adopts new technologies. The biggest obstacle is the management mentality. This is however changing with every generation, but seems to be happening at a slower pace than in other industries. Therefore, the first step to implement any new technology would have to be either changing the company culture or developing products which can overcome this obstacle. For external developers, the latter would be the easier path to take. But there one promising way to get management and employees on board, by focusing on safety. The construction industry is one of the most dangerous, which is why many recent applications are focused on assessing safety conditions in the workplace.

This report will start with an overview of the literature used throughout the report. Following that, is a section that covers the possibilities for machine learning in construction. The focus will be on three major managerial aspects, project planning, safety management and project management. These solutions will attempt to accommodate for current industry mentality and technological feasibility. The report will then move on to present working solutions for two of the three previously mentioned managerial aspects, project planning and safety management. Finally, there will be discussions on limitations and further analysis along with concluding remarks.

2 Literature Review

The purpose of this report is to identify operational areas where machine learning can transform the modern construction industry. The focus will be on implementing methods from three separate fields to increase the construction industry's productivity. These methods come from time estimation approaches, natural language processing and computer vision. To improve knowledge in these fields, research into pre-existing solutions or similar implementations was conducted. Furthermore, research into effective and commonly used machine learning algorithms was conducted so they could be used to develop a working time estimation solution. This includes gradient boosting, random forest and Bayesian optimisation.

2.1 Gradient Boosting and Random Forest

Random forest is based on the idea of bagging, where individual trees are built on a bootstrap sample of the data. Randomness is then introduced into each tree by only allowing a random subset of features to split on. In the end predictions are made based on all trees, either by majority vote for classification or averaging for regression. This method is very good at controlling overfitting and reducing variance. It can also be grown in parallel since none of the trees are connected until they are used for prediction.

Another very popular tree-based algorithm is called gradient boosting. Different from random forest, gradient boosting method builds each tree sequentially where each tree is simple, often referred to as a weak learner. The errors from each weak learner is used when growing the next tree to increase emphasis on the training data that was difficult to estimate. The term boosting actually refers to a family of algorithms that convert many weak learners to a single strong learner. In gradient boosting each tree has the aim to gradually minimise the loss function, such as squared error or absolute error, of the whole system. The loss function measures how good the model is at fitting the data. The gradient boosting algorithm uses a two-step procedure when creating each weak learner. First step is to fit a weak learner with the negative gradient of the loss function by using least-squares. Then a weighted value for each weak learner is determined by minimising the overall loss function using gradient descent. Guelman (2012)

Bayesian optimisation is a very effective method to find the global optimum of noisy black-box functions, where the optimisation problem aims to minimise a given function by changing select variables. Bayesian optimisation is therefore often used to tune model hyperparameters, parameters which are fed into the model to control bias and variance. The optimisation problem treats them as the variables and the objective function is to minimise error. This method has been gaining popularity in machine learning to automate the hyperparameters tuning and deployment process. Previous methods such as grid search and random cross validation search can be very expensive to carry out when there are many hyperparameters with a wide range of possibilities. Hua & Jie (2019)
Bayesian optimisation works by construction a posterior distribution function that can produce the best approximation of the optimised function. With each new observation, this posterior distribution improves and the algorithm becomes more aware of what hyperparameter space should be explored. This exploration is iterative, with the algorithm using its knowledge of the target function to fit a new posterior distribution and use it, with an exploration strategy, to determine where to explore next. With this process, the number of steps needed to approximate the optimal combination of hyperparameters are minimised. Adams et al. (2012)

2.2 Time Estimation Methods

Time scheduling is a fundamental step for project planning but also the most prone to errors or miscalculations. Which can be caused by logical errors or unexpected task complications. The latter can sometimes be reduced by using distributions with a long right tail to emulate how task duration can sometimes extend unexpectedly.

One example of a widely used project management tool that implements a right-skewed distribution to estimate task duration is the program evaluation and review technique or PERT. The PERT-beta distribution is a transformation of the four-parameter β distribution. Where the only data points required are the optimistic, most likely and pessimistic time estimates. However, to get these parameters one would need historical data to get reliable numbers. Researchers have long debated that other distributions can outperform the PERT-beta in a practical environment when the three previously mentioned parameters are inaccurate.Hajdua & Bokor (2014) There have been numerous research papers that support this statement by showing superior distribution performance over the PERT distribution, these include the log-normal Mohan et al. (2007) and Weibull McCombs et al. (2009) distributions. Those who defend the PERT, state that choosing another distribution would be irrelevant since the estimates are always imprecise. This is a valid point and a random sampling from any of these distributions will not likely prove to be an effective time estimation method, so other methods were investigated.

A collection of statistical procedures called survival analysis have been used to produce more robust time estimates. For example, both in recent and past studies, it has been used to estimate task duration for crowdsourced tasks Wang et al. (2011) and to compare variable importance and response based on survival behaviour. Melnyk et al. (1995) The reason why survival analysis relates well to time estimation is because it uses data to analyse how much time will pass until an event occurs, usually referred to as the survival time. In time estimation this could be interpreted as completion time. Kleinbaum & Klein (2012) To fit task distribution, survival analysis can replicate the previously mentioned probabilistic distributions, using specific survival and hazard functions. Bispo et al. (2013)

However, survival analysis is not a very recent methodology and this was only used as a foundation to identify more advanced time estimation methods. Alternative survival analysis methods, who can deliver better accuracy in some cases, are so called tree-structured survival models. Cappelli & Zhang (2007) This could indicate that a more advanced tree-structured machine learning algorithm might also be good to accurately predict task duration. Solution for travel time estimation has been tried by using gradient boosted decision trees Zhang & Haghani (2015) and in a more recent study, gradient boosting decision trees was used to estimate case-time in surgical operation rooms with good results. Bartek et al. (2019) The variables used in the model are a mixture of patient, surgeon and procedure related data. This method outperformed other machine learning algorithms and the commonly used historical averages. The latter is most often the main factor used for time scheduling in construction industry. Based on these experiments, the method might provide some significant opportunities to improve scheduling in construction.

2.3 Natural Language Processing

Free text fields are a constant problem for clean databases across all industries. The main problems arise from human errors when writing in the information or difficulty in extracting relevant information without manual labour. Natural language processing (NLP) is a fast-evolving research field that gives computers the ability to read, understand and analyse text written by humans. The rapid recent improvements are mostly thanks to increased data and computational power. This technology could be extremely useful in construction when dealing with progress, safety and inspection reports. For example, a recent paper attempted to extract precursors and outcomes from unstructured injury reports with exceptional accuracy. Tixier et al. (2016) The core of this paper was to use NLP along with manually defined keyword dictionaries to classify injury reports.

2.4 Computer Vision

Computer vision is another field that has taken great strides in recent years with the increased computing power. This more advanced technology has been able to increase accuracy and open the doors to some deep learning implementations. Current research is looking at a wide range of solutions for construction. For example, automatic identification of personal protective equipment (PPE) on construction sites. Mneymneh et al. (2017) Where pattern recognition is used to identify high visibility clothing and hardhats. More complicated solutions attempt to solve crucial health and safety issues by using object detection, object tracking and action recognition. Seo et al. (2015) Finally, a difficult and important problem that has been researched is to inspect and assess the condition of civil infrastructure by using deep learning. Spencer et al. (2019) The possibilities here are limitless but can stretch far into the future. Some years in the future, earthmoving equipment might be fully automated. There are already some experiments that have been conducted using GPS technology. Bonchis et al. (2011) By combining this with the computer vision technology currently used in self driving cars, every earthmoving project might become automated in the future.

Computer vision is a fast-growing field, but it can be extremely complicated and there are many limitations that currently follow it. This includes the difficulty of scene understanding and to comprehend the three-dimensional structure of images. However, experimentation on simple approaches that can use object detection, visible background estimation and shape priors to label the visible and obstructed portions of image background. This technology extended to identify three dimensional surfaces. Guo & Hoiem (2015)

3 Possibilities in Construction

This section will present implementation ideas for machine learning in the construction industry. It will be split into three major aspects that every construction project is highly reliant on, planning, management and safety. Each section could benefit from fields within computer science and artificial intelligence who vary greatly in complexity. How these fields can be utilised will both be conceptual and supported by current research and technology covered in the theory review section.

3.1 Project Planning

Project planning is a fundamental step in every industry and the construction industry is no exception. A well-structured progress schedule is required to gain trust from investors and lenders who will fund the project. Because of this, managers and directors spend many hours analysing all tasks and estimating completion time. The most common method in project planning is using historical data to estimate this task duration. A technique that does exactly this is the previously mentioned PERT technique. Where the task duration is estimated with a three-parameter distribution, the optimistic, most likely and pessimistic time estimates. Historical data and professional experience is most often the main factors used to estimate each of these three parameters. However, years of experience is needed to identify the complicated design aspects which cause longer completion time, this is also often highly subjective and prone to human error.

A more robust approach would be to use a machine learning algorithm to estimate the duration of each work breakdown structure (WBS) item. Work breakdown structure is a common method that breaks a single large project into multiple smaller work items. In most cases, historical and characteristics data, is currently being used to estimate task duration. If the algorithm would be able to easily access these characteristics, it could identify patterns in minutes that a human would take years to learn. As previously mentioned, this was tried in a recent research project that used gradient boosting to estimate case-time in surgical operation rooms. Bartek et al. (2019) The results managed to show the advantages of using machine learning for time estimations. However, the conclusion was that to be able to gain from this, trained models would be needed for every surgeon. This would most likely be the case as well for each work breakdown structure or WBS item in a construction project. Therefore, training and development time would have to be taken into consideration when observing the feasibility of this solution.

3.2 Safety Management

As previously mentioned, the construction industry is one of the largest in the world. In the United Kingdom it accounts for about 6.5% of the country's GDP (gross domestic product), according to numbers from 2014. Infrastructure & Authority (2016) However, the industry's total number of fatal injuries in 2018/19 was 30, which accounts for 20% of the country's total workplace fatalities. The second highest of all major industries behind the combined number of 32 in agriculture, forestry and fishing. Unfortunately, these are not surprising statistics because globally, construction is one of the most dangerous profession.

With this knowledge, it is clear that plenty can be gained from improved technology for safety related intervention and assistance. Safety is something that everyone can gain from, both individual employees and managers. Since this is a high-risk industry, there already is a well-established mentality for incident documentation across the industry. Most often in free text and image format, collected through a structured report or just a single photograph.

This data collection method means that any data analysis is challenging. Currently, most companies rely on every text-based report to be read and every image to be visually inspected by managers to gain any information. Therefore, a logical next step would be to automate this by using NLP or computer vision. There are already solutions emerging that are identifying repeating patterns in written reports to give a quick overview of common bad practices. As can be seen in the previously mentioned paper. Where precursors and outcomes were extracted from

unstructured injury reports. Tixier et al. (2016) More advanced solutions would be able to use computer vision to identify the bad practices seen in a photograph. This algorithm would either be allowed to automatically identify bad practices and log them or give text-based suggestions that have to be accepted. Even more complicated possibilities would rely on scene understanding to map three-dimensional information from images. This would greatly increase the information extracted from each inspection photo. Being able to understand heights and ledges would give the algorithm multiple new features to observe. Furthermore, this could be used on aerial photos, more commonly taken on large construction sites. These photographs could identify areas on the construction site where safety practices are not acceptable. For example, if there is a lack of railings where there is risk of falling from heights.

3.3 Project Management

Daily management at a construction site shares many aspects with a large-scale manufacturing business. There is a product delivery date that has to be met, resources and manpower are limited and product quality has to follow expectations. However, in construction, the last factor is of the highest importance. The lives of all occupant depend on it and a single structural failure can have severe consequences. To ensure that all work is up to standard, regular inspections are carried out where countless inspection documents and photographs are generated.

To go through all this documentation, some companies are developing software to remove manual work. For example, computer vision can assist by organising all photographs based on the WBS item or location. So, if someone photographs the final reinforced steel placement before it is covered in concrete. The general contractor might need to access this for future reference or quality assurance. In that case, the automatic organisation would save hours of either looking through unstructured folders or manual organisation. Further in the future these photographs could even be used by machine learning technology to identify faulty construction practices through abnormal bends or cracks.

Another inspection related solution would implement speech recognition to automatically log the inspector's remarks. Speech recognition technology has shown incredible progress in recent years, with some companies claiming over 90% accuracy. Huang (2017) Having these remarks on text formats is often necessary for documentation. It would also enable content analysis similar to the one mentioned in the safety management section. By doing that, recurring comments from quality inspectors could help identifying where bad practices have to be addressed.

4 Practical Cases

The previously mentioned concepts and solutions vary greatly in difficulty. Both when it comes to computational power and theoretical difficulty. This section will introduce some working models that were created to demonstrate their feasibility and usefulness. The main experimentation was focused on developing a time estimation model along with some experimentation with NLP. Knowledge and time limitations were the main drivers behind these two choices, computer vision is extremely complication and could therefore not be attempted.

4.1 Project Estimation Testing

The first and most important step to be able to test the previously mentioned concepts, is to acquire data. Unsurprisingly, this proved to be very challenging since the main obstacle of machine learning in construction is the lack of data collection. After reaching out to several construction companies in Iceland, the responses were always positive when it came to providing data. But the problem was always the same, data collection was not done on enough granularity. Most often, the only numbers were the initial estimate and the final completion time of the entire project. These two data points alone are not likely to result in an accurate model. Eventually, one company was found that could provide detailed data for two projects. The company asked however not to be named in the report. So, from now on the projects will be referred to as project 1 and project 2 and the data from both projects was combined into a single data set.

At first look at this data it became clear that number of observations was very limited. The data set only consisted of 9 WBS items, some not even available for both projects. For these WBS items, project 1 had at most one observation and project 2 had around four observations for different project sections such as basement, first floor and others. Because of this data structure, it was decided to first use the observations from project 1 to estimate completion time for each corresponding WBS item in project 2. This would by done by finding the ratio of man-hours per unit and use that ratio to estimate the WBS item duration in project 1 based on number of units. This method attempts to replicate the current method that most companies use. It is worth mentioning the companies also use domain specific knowledge to tune these rough estimates, but this will not be attempted here. The results from these estimates can be seen in Table 1. Comparisons are done using mean absolute percentage error (MAPE), a useful method when comparing model performance on different data sets, a necessary element in this report. The MAPE is calculated with the following equation.

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{A_i - F_i}{A_i} \right|$$

Here A denotes the actual values while F are the forecast values. The MAPE can exceed 100% because it shows percentage deviation, for example if all prediction are three times higher than the real value then the resulting MAPE would be 200%.

WBS Item	**Mean Absolute Percentage Error**
Ceilings	50.9 %
Concrete Flooring	43.73 %
Doors, Inside	778.86 %
Floors, Finishing	85.56 %
Inner Walls	112.53 %
Paint, Inside	79.59 %
Overall	**101.28 %**

Table 1: Mean absolute percentage error based on the performance of project 1 ratio-based predictions for project 2 done for every WBS item

It can be seen from this table that this ratio-based method is failing overall, with about 100% MAPE. There are moments where the results are acceptable, for example in the concrete flooring. But also cases where the estimates are drastically off, as in the inside door estimates. This method is therefore in no way good enough to use for actual estimation. The lack of data and the roughness of the approach results in very poor performance. If the objective is to automate the process, better methods have to be implemented.

To examine the performance of machine learning algorithms on construction data, the observations from project 2 could be split up to predict time duration for its WBS items. In other words, a part of project 2 would be used to predict on a separate part of project 2. This evaluation was done by using leave-one-out cross validation and by analysing the cross validation error. The small size of the data set is far from ideal but this might be able to hint at how machine learning can perform in this situation. However, for this to succeed, additional feature variables have to be added. Data regarding project specific information such as quantities and characteristics of each WBS item is not easily accessible since it is mostly kept on architectural drawings. An example of how such a drawing looks like can be seen in Figure 1. Design and drawing software might be able to extract these numbers. But for this report, manual work was needed to extract the information from the drawings. This is a highly inefficient and error prone data extraction method. Especially since the details on these drawings can easily cause errors when reading them for an extended period of time. Therefore, investigation into other, more automated, methods would be crucial for future growth.

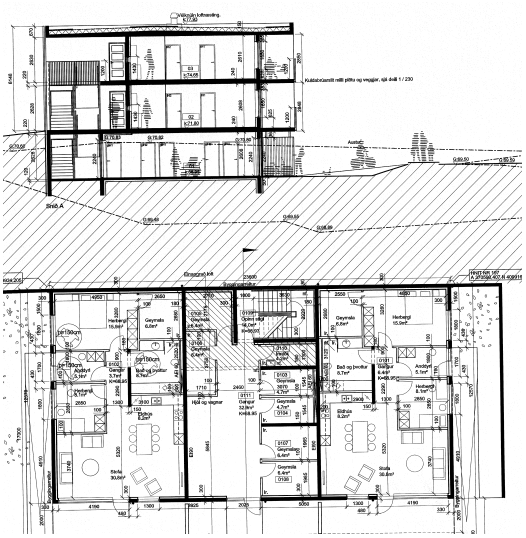

Figure 1: Sample image showing a drawing similar to the ones used in data extraction for this report, the drawings are very visually clustered

With the features created from the drawings, multiple models could be trained and fitted for each work item. Examples of these features are the number of holes and corners in the outer and inner walls. This information can raise prediction performance because holes and corners increase complexity and man-hours when setting up concrete moulds for outer walls and structural skeletons for inner walls. To explore various regression performances, three different algorithms were tested with the default hyperparameter settings, regression tree, random forest and XGBoost. The results from each model compared to the original project 1 (P1) ratio estimations can be seen in Table 2.

WBS Item	Regression Tree	Random Forest	XGBoost	P1 Estimations
Outer Walls	32.37 %	31.14 %	32.28 %	NaN
Floors, Finishing	38.01 %	42.6 %	37.84 %	85.56 %
Ceilings	43.61 %	36.6 %	43.42 %	50.9 %
Concrete Flooring	41.23 %	42.21 %	43.18 %	43.73 %
Inner Walls	97.48 %	92.31 %	126.32 %	112.53 %
Glass Walls	24.97 %	27.18 %	24.83 %	NaN
Paint, Inside	260.04 %	364.62 %	259.73 %	79.59 %
Doors, Inside	NaN	NaN	NaN	778.86 %
Overall	**78.27 %**	**92.99 %**	**82.73 %**	**101.28 %**

Table 2: Comparison table showing the cross validation MAPE for every WBS item and multiple models – NaN values in P1 estimations are caused by no data in project 1 and NaN values in other models was because of too few rows to perform cross validation

The table shows that in most cases some models are performing better than the initial P1 ratio estimates. The WBS items where most additional features were added are outer walls, inner walls, glass walls and inside paint. Outer walls and glass walls are experiencing good performance but lack of data means that there is no benchmark to compare to with the original estimation. Inner walls are performing equally bad as before while inside paint is seeing incredibly bad performance, maybe caused by faulty features. It can also be assumed that the size of the data set is having negative effect on model performance and confidence. The data points are also all from the same project so the correlation between the WBS items can be the only thing causing the occasional good performance. So, even though three of the models show lower overall mean absolute percentage error, it cannot be stated that they are superior estimation approaches. The performance quality is also far from a feasible product that creates progress schedules.

One problem with the earlier machine learning approach is that the data points are all from the same project which can cause the models to succeed just because of the similarity. However, it can be argued that with enough additional project specific data collection these similarity patterns would be accessible to the models. For example, when working on outer walls, increased height can cause problems. So, if this would be included as a defining feature, the models would be able to distinguish between projects based on that feature similarity.
To address all these data related problems, data similarity, lack of data points and feature scarcity, a pseudo data set was used to test the feasibility. Because of the overall lack of data collection that would take years to remedy, a non-construction related data set had to be used. The chosen data set consists of roughly 1.4 million New York City taxi journeys. Kaggle (2017) The data set has one column with trip duration along ten columns to use for predictions and use to engineer new features. This included trip ID, vendor ID, passenger count, and data log method along with the time and location of the pickup and drop-off. This data set was assumed to be similar enough to task duration information for a construction WBS item. The underlying concept is the same, to estimating the completion time of a single task that mainly depends on one fundamental unit and where multiple factors can cause complications or delays. In this case that unit is distance,

whereas in construction it might be the length of a concrete wall.

With this large and detailed data set, the possibilities of the whole model grew extensively. But like with most large data sets, there was quite a lot of pre-processing required to be able to use it for accurate predictions. The data looked to be relatively clean, but there were still a few notable errors. Some trips lasted for up to eight hours, some were located in a different state and some had coordinates that were located in the middle of the ocean. With the data cleaned, additional features could be added. This included extracting haversine distance, heading, borough information and routing distance, among others. The routing distance was extracted using an open source routing machine, or OSRM, project.

When all the data had been cleaned and new features prepared, the same models were tested to examine what model would be the strongest candidate for finer tuning. These models were, regression tree, random forest and XGboost. The performance was now evaluated using 5-fold cross validation, the results for each model can be seen in table 3.

Model	**Mean Absolute Percentage Error**	**Root Mean Squared Error**
Regression Tree	44.02 %	353.09
Random Forest	34.15 %	251.55
XGBoost	40.05 %	276.02

Table 3: Comparison of cross validation MAPE for multiple regression models

As the table shows, all models are showing promising results, with random forest performing best. Because all models are tree based this might support the previous claims that tree-based algorithms are powerful for time estimations. Before moving forward, two additional things were now considered. First of all, accuracy had to be good for the model to be of any use. Secondly, the model training time had to be considered. Especially since that for the time estimation, multiple models might have to be trained for each WBS item. Random forest was performing better but it was computationally expensive, taking about twice as long as the XGBoost. It can be run in parallel on multiple CPUs for faster training but the same can be said about the XGBoost algorithm.

Based on this information, the XGBoost algorithm was chosen for further tuning and analysis. The hyperparameter tuning was done using Bayesian optimisation and ran on an elastic computing instance at Amazon Web Services. With all hyperparameters tuned, the final model could be retrained once more on the entire training set and predictions could be made on the test set. The accuracy for the XGBoost model and the initial random forest model, trained and tested in the same way, can be seen in Table 4. A benchmark estimation created from the time per unit ratio, the same method used for the project 1 construction estimation, is also included.

Model	Mean Absolute Percentage Error	Root Mean Squared Error
Benchmark Estimate	131.96 %	2114.88
Random Forest	33.68 %	248.84
XGBoost	**31.83 %**	**239.75**

Table 4: Comparison of the report's core estimation methods when trained on the entire training data and tested on the test data

As the table shows, the final XGBoost model now slightly outperformed the initial random forest model and performed way better than a benchmark created from the time per unit estimation. The faster training and tuning times are what initially made XGBoost preferable and now the performance is also better. With these results, the time estimation of WBS items, using XGBoost, should be well within reach if the data collection would be better.

4.2 Automated Content Analysis for Incident Reports

There was also a desire to create a simple working example implementing NLP to identify risk factors. To do this a data set containing OSHA accident and injury reports for construction workers from 2015-2017 was used. Shipwrekt (2018) OSHA stands for the Occupational Safety and Health Administration and is an agency of the United States Department of Labour.

The solution takes two free text descriptions of the accident and finds repeating words within them. One text is a long abstract version and the second is a short description containing key details. The approach breaks each word down to their root form and counts the frequency in the both texts. To evaluate the accuracy of this method, a keyword column included in the data set, most likely recorded along with the other texts, could be used. The initial results from this experiment were not very good, with only about 28 % of the keywords being correctly identified in the text.
The main cause of this were found to be numerous noise words in both texts, words related to personality descriptions, dates and counting, among others. These words would not provide any valuable information when trying to identify causes of accidents. Therefore, it was decided to filter out majority of this noise, a similar solution used in the previously mentioned research where precursors and outcomes were extracted from unstructured injury reports. Tixier et al. (2016) The weight on the event description words were also increased because these descriptions are often more to the points. These fixes did however not deliver any spectacular improvements, with the accuracy only increasing up to about 37%.

The real gains from this is hard to quantify, but this could serve as an instructive tool for managers. It would provide oversight and identify repeating causes of injury. An example of how this information could be conveyed to a construction manager can be seen in the mock-up in Figure 2.

Figure 2: Application mock-up showing a conceptional representation of a health and safety information screen, information is based on the NLP solution

The figure shows the most frequent words from both texts, based on results from the NLP experimentation. This would allow the health and safety manager to respond to dangers from falling and being struck by something. To the left an example of overall risk factors can be seen, some are related to external data such as weather while some could be linked to these repeating report words. This information can help with decision making that affects the most important aspect of the workplace, employee safety. The experimental technique tried here needs a lot of work to be a feasible product to increase safety. But when there is this much to be gained, the incentives should be clear.

5 Discussion

5.1 Limitations and Further Analysis

The limitation which had the most effect on this report is the lack of data and data collection culture within the industry. One initial objective of this report was to create a simple working solution to this, a data collection application. The concept was that if the fundamental use of the application would be a time clock to track work hours for each employee. It could assist with payroll while collecting the WBS item data on the side. However, this application could not be developed because of access restrictions in the Imperial College Office package.
The material covered in this report stretches over multiple applications and technological fields. There are plenty of opportunities to tune the NLP solution into a working safety management product. There are also many possibilities when it comes to the image processing solutions, these could not be attempted in this report because of time and difficulty. Finally, a machine learning product for the construction industry has long been on the author's mind. This report will serve as excellent foundation for any product development attempts in the future.

5.2 Conclusions

This report has covered what are the next logical steps for machine learning implementation in the construction industry. As previously mentioned, the solutions vary greatly in complexity and technological fields. Nevertheless, some of these concepts were tested by using both real and pseudo data sets, with very promising results. The initial tests with construction data resulted in MAPE of about 80%, a rather poor performance. However, the pseudo taxi data set was able to lower the percentage to 32%, a very promising result. Based on these conceptional ideas, who were backed by recent research, there is clearly a lot to be gained. Both when it comes to financial gain for owners, but more importantly, invaluable gains from personnel safety. There are however difficult hurdles that have to be dealt with first. Mainly the lack of data collection and technological mentality within the industry. This is most likely causing the lack of machine learning solutions and slow growth rate that plagues the industry. Some ideas were also covered which might ease with the technological transformation while the solutions are still proving their worth. These ideas would be the next logical steps so machine learning could transform the industry. Setting up data collection systems that can deliver instant gains in daily management should be the first priority. This could get management on board and show owners the financial gains, paving the way for technological breakthroughs.

6 Appendices

6.1 Python Notebooks and Scripts

This section outlines the purpose of all the Jupyter notebooks and Python scripts.

6.1.1 Construction Notebooks

Construction_Time_Estimation_Models.ipynb: Takes in the summary construction data. Calculates some rough ratio-based time estimation and experiments with a few regression based algorithms.

6.1.2 Taxi Notebooks and Scripts

Geocoding.ipynb: Takes in the taxi data and predefined coordinate polygons, these include borough and airport borders. The pickup and drop off coordinates are then segmented into boroughs based on the polygons. Furthermore, a binary column is created that identifies if the pickup or drop off was near an airport.

OSRM_Route_Scraper.ipynb: Takes in the taxi data and uses the pickup and drop off coordinates to identify the best route between the two points. The route is supplied by a python client for the open source routing machine, OSRM, project API.

Data_Wrangling_NYC_Taxi.ipynb: Takes in the taxi, geocoding and OSRM routing data. Does all required cleaning and feature engineering for the final time estimation models.

Taxi_Time_Estimation_Models.ipynb: Takes in the cleaned data and runs multiple models to identify the best solution. When the best model had been determined, it trains and tests a fine tuned version of that model.

XGB_Tuning.py: A separate Python script that takes care of hyperparameter tuning for the XGBoost model. This had to be in a separate file so the tuning could be done with Amazon Web Services.

6.1.3 Natural Language Processing Notebook

NLP_Analysis.ipynb: Takes in accident and injury reports and does some experimental natural language processing on that data. Both by checking accuracy based on predefined keywords and by extracting the most common words to identify repeating causes.

6.2 Amazon Web Service: Elastic Computing Instance

This section shows the Bayesian optimisation iterations from the XGBoost hyperparameter tuning. This was all done by using a AWS EC2 instance.

iter	target	colsam...	eta	gamma	max_depth	min_ch...
1	-174.2	0.5494	0.2579	0.3999	20.46	17.52
2	-171.8	0.3638	0.1123	0.6345	10.54	19.65
3	-168.4	0.6947	0.07682	0.4113	12.44	22.29
4	-172.3	0.6044	0.2823	0.5144	7.805	16.87
5	-174.5	0.5548	0.2725	0.2082	18.29	12.0
6	-176.0	0.7738	0.1686	0.426	28.51	6.904
7	-170.0	0.3414	0.1226	0.6886	13.96	10.89
8	-174.0	0.6812	0.259	0.5403	18.3	10.61
9	-173.4	0.4554	0.2738	0.4464	20.88	29.25
10	-178.1	0.5295	0.2622	0.4121	5.781	12.67
11	-168.9	0.6524	0.2481	0.483	11.46	17.43
12	-167.9	0.6673	0.09775	0.6388	25.93	22.25
13	-169.1	0.7935	0.2778	0.4824	12.17	16.53
14	-171.3	0.7923	0.2574	0.576	18.81	23.25
15	-175.5	0.4996	0.2381	0.5245	23.89	17.74
16	-178.3	0.7805	0.2426	0.4038	25.25	10.75
17	-172.8	0.4838	0.213	0.5205	25.2	28.09
18	-171.5	0.4961	0.2068	0.5172	8.1	23.36
19	-171.0	0.487	0.2132	0.3326	17.69	13.38
20	-177.2	0.3043	0.1649	0.4053	22.33	8.784
21	-167.4	0.7029	0.07244	0.6618	26.23	22.94
22	-166.9	0.7062	0.05	0.6752	26.91	22.65
23	-183.6	0.3	0.3	0.2	26.95	22.76
24	-167.6	0.7234	0.1552	0.5512	13.75	13.36
25	-173.3	0.4645	0.1897	0.349	25.38	20.53
26	-168.7	0.4394	0.1599	0.2304	11.02	20.2
27	-174.2	0.4231	0.1911	0.6997	7.599	8.845
28	-168.3	0.5377	0.05244	0.4664	26.83	22.03
29	-169.6	0.7726	0.1566	0.4786	9.94	17.37
30	-170.7	0.4553	0.08736	0.5532	24.2	10.26
31	-169.1	0.5476	0.2419	0.5841	14.6	24.68
32	-169.0	0.452	0.2052	0.3378	13.87	12.18
33	-167.8	0.5443	0.137	0.4462	12.35	22.21
34	-166.8	0.7442	0.05	0.7	26.84	22.6
35	-166.8	0.7392	0.05	0.7	26.94	22.55
36	-166.8	0.8	0.05	0.6582	26.92	22.62
37	-166.8	0.7437	0.05	0.6071	26.88	22.55
38	-166.8	0.8	0.05	0.6648	26.87	22.49
39	-167.0	0.696	0.05	0.6628	26.84	22.41
40	-166.8	0.7689	0.05	0.5967	26.93	22.36
41	-166.7	0.7851	0.05	0.5536	26.8	22.28
42	-170.2	0.7168	0.1416	0.4751	9.907	17.4
43	-166.7	0.7515	0.05	0.6411	26.87	22.19
44	-170.9	0.7897	0.1729	0.6263	26.85	22.28
45	-167.0	0.6811	0.05	0.5512	26.86	22.24
46	-166.8	0.7497	0.05	0.5531	26.84	22.39
47	-167.1	0.7136	0.06259	0.6683	26.3	22.86
48	-166.7	0.7506	0.05	0.5578	26.81	22.14
49	-168.4	0.6104	0.1051	0.6417	26.28	22.89
50	-166.7	0.7817	0.05	0.6256	26.83	22.66

Figure 3: Sample image showing a drawing similar to the ones used in data extraction for this report

References

Adams, R. P., Snoek, J. & Larochelle, H. (2012), 'Practical bayesian optimization of machine learning algorithms'.

Barbosa, F., Woetzel, J., Mischke, J., Ribeirinho, M. J., Sridhar, M., Parsons, M., Bertram, N. & Brown, S. (2017), Reinventing construction through a productivity revolution, Technical report, McKinsey Global Institute.

Bartek, M. A., Saxena, R. C., Solomon, S., Fong, C. T., Behara, L. D., Venigandla, R., Velagapudi, K., Lang, J. D. & Nair, B. G. (2019), 'Improving operating room efficiency: A machine learning approach to predict case-time duration', *Journal of the American College of Surgeons* .
URL: *https://doi.org/10.1016/j.jamcollsurg.2019.05.029*

Bispo, R., Bernardino, J., Marques, T. A. & Pestana, D. (2013), *Discrimination Between Parametric Survival Models for Removal Times of Bird Carcasses in Scavenger Removal Trials at Wind Turbines Sites*, Advances in Regression, Survival Analysis, Extreme Values, Markov Processes and Other Statistical Applications, Springer-Verlag Berlin Heidelberg, Berlin, pp. 65–72.

Bonchis, A., Hillier, N., Ryde, J., Duff, E. & Pradalier, C. (2011), 'Experiments in autonomous earth moving', *IFAC Proceedings Volumes* **44**(1), 11588–11593.
URL: *https://doi.org/10.3182/20110828-6-IT-1002.00536*

Cappelli, C. & Zhang, H. (2007), *Survival Trees*, Statistical Methods for Biostatistics and Related Fields, Springer-Verlag, Berlin, pp. 167–179.

Guelman, L. (2012), 'Gradient boosting trees for auto insurance loss cost modeling and prediction', *Expert Systems with Applications* **39**(3), 3659–3667.
URL: *https://doi.org/10.1016/j.eswa.2011.09.058*

Guo, R. & Hoiem, D. (2015), 'Labeling complete surfaces in scene understanding', *International Journal of Computer Vision* **112**(2), 172–187.
URL: *https://doi.org/10.1007/s11263-014-0776-7*

Hajdua, M. & Bokor, O. (2014), 'The effects of different activity distributions on project duration in pert networks', *Procedia - Social and Behavioral Sciences* **119**, 766–775.
URL: *https://doi.org/10.1016/j.sbspro.2014.03.086*

Hua, C. & Jie, B. (2019), 'A new hyperparameters optimization method for convolutional neural networks', *Pattern Recognition Letters* **125**, 828–834.
URL: *https://doi.org/10.1016/j.patrec.2019.02.009*

Huang, X. (2017), 'Microsoft researchers achieve new conversational speech recognition milestone'. Accessed on 12th August 2019.
URL: *https://www.microsoft.com/en-us/research/blog/microsoft-researchers-achieve-new-conversational-speech-recognition-milestone/*

Infrastructure & Authority, P. (2016), Government construction strategy: 2016 - 2020, Technical report, Great Britain.

Kleinbaum, D. G. & Klein, M. (2012), *Survival Analysis: A Self-Learning Text, Third Edition*, 3rd edn, New York, NY.

McCombs, E. L., Elam, M. E. & Pratt, D. (2009), 'Estimating task duration in pert using the weibull probability distribution', *Journal of Modern Applied Statistical Methods* **8**(1), 282–288.
URL: *https://doi.org/10.22237/jmasm/1241137500*

Melnyk, S. A., Pagell, M., Jorae, G. & Sharpe, A. S. (1995), 'Applying survival analysis to operations management: Analyzing the differences in donor classes in the blood donation process', *Journal of Operations Management* **13**(4), 339–356.
URL: *https://doi.org/10.1016/0272-6963(95)00031-3*

Mneymneh, B. E., Abbas, M. & Khoury, H. (2017), 'Automated hardhat detection for construction safety applications', *Procedia Engineering* **196**, 895–902.
URL: *https://doi.org/10.1016/j.proeng.2017.08.022*

Mohan, S., Gopalakrishnan, M., Balasubramanian, H. & Chandrashekar, A. (2007), 'A lognormal approximation of activity duration in pert using two time estimates', *Journal of the Operational Research Society* **58**(6), 827–831.
URL: *https://doi.org/10.1057/palgrave.jors.2602204*

Seo, J., Han, S., Lee, S. & Kim, H. (2015), 'Computer vision techniques for construction safety and health monitoring', *Advanced Engineering Informatics* **29**(2), 239–251.
URL: *https://doi.org/10.1016/j.aei.2015.02.001*

Spencer, B. F., Hoskerea, V. & Narazakia, Y. (2019), 'Advances in computer vision-based civil infrastructure inspection and monitoring', *Engineering* **5**(2), 199–222.
URL: *https://doi.org/10.1016/j.eng.2018.11.030*

Tixier, A. J. P., Hallowell, M. R., Rajagopalan, B. & Bowman, D. (2016), 'Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports', *Automation in Construction* **62**, 45–56.
URL: *https://doi.org/10.1016/j.autcon.2015.11.001*

Wang, J., Faridani, S. & Ipeirotis, P. G. (2011), Estimating the completion time of crowdsourced tasks using survival analysis models, *in* 'Crowdsourcing for Search and Data Mining (CSDM 2011) Workshop of the Fourth ACM International Conference on Web Search and Data Mining (WSDM 2011)', Vol. 31, pp. 31–34.

Zhang, Y. & Haghani, A. (2015), 'A gradient boosting method to improve travel time prediction', *Transportation Research Part C: Emerging Technologies* **58**(Part B), 308–324.
URL: *https://doi.org/10.1016/j.trc.2015.02.019*

Data Set Sources

Kaggle (2017), 'New york city taxi trip duration [csv files]'. Accessed on 13th August 2019.
URL: *https://www.kaggle.com/c/nyc-taxi-trip-duration*

Shipwrekt, K. (2018), 'Osha accident and injury data injury records for 2015-2017 [csv file]'. Accessed on 13th August 2019.
URL: *https://www.kaggle.com/ruqaiyaship/osha-accident-and-injury-data-1517*