Towards Higher Maturity for Machine Learning: A Conceptual Modelling Approach

Rohith Sothilingam, Eric Yu, Arik Senderovich

Abstract

Integrating machine learning (ML) applications into business settings presents challenges for many organizations despite rapid advances in ML technologies. There is a lack of systematic guidance in integrating ML into business applications. Conceptual modelling techniques have been used widely to analyze and enhance information systems. This paper outlines how conceptual modelling can be used to help organizations design their processes in order to fit their different needs when integrating ML into business applications. We identify the characteristics and challenges of ML as well as demonstrate how conceptual modelling can be applied to enhance ML integration processes and assist them in meeting these challenges.

Keywords: Enterprise Machine Learning, Conceptual Modelling, Process Modelling

1. Introduction

Machine learning (ML) is emerging as an important part of today's information systems and is a technology many are seeking to integrate into business settings. Designing processes for ML presents new challenges and differs from traditional business processes, with different requirements which current information systems analysis and design methods are not able to address. ML integration processes need to be well-designed and expressible to develop a systematic approach for expressing the complex process characteristics of ML integration.

This paper presents conceptual modelling as an approach to systematically design ML processes. The sub-field of conceptual modelling within computing and information systems is focused on developing modeling techniques for improving the level of understanding and communication among stakeholders and to define requirements for system development (Wand, Monarchi, Parsons & Woo, 1995). Conceptual modelling has been widely used for designing and expressing processes in information systems methods across many applications with known benefits. For example, business process models are recognized to improve processes, facilitate communication and support process design (Indulska, Green, Recker, & Rosemann, 2009). This paper outlines how conceptual modelling methods, with a focus on business process modelling techniques, can be used to improve ML processes and express their unique underlying characteristics, and thus to ease their integration into business settings.

Given the unique problems and challenges posed by ML processes, we outline the requirements for a conceptual modelling notation to be able to express these characteristics. The ML workflow involves a variety of roles, qualities and goals which are complex and may not be well-understood by all team members and stakeholders. It would be beneficial to improve current processes and visualize the details, connections and feedback cycles between their different elements. This can potentially help organizations to improve their maturity when integrating ML applications. Project managers and business analysts will be able to design processes specific to their organization by using these conceptual models.

In this paper we define the characteristics of ML and outline the need for systematic processes to meet the new challenges presented by these characteristics (Section 2). We express the benefits of conceptual modelling and apply models to demonstrate how current modelling methods can address new challenges presented by ML processes (Section 3). Based on the new challenges presented by ML, we define the requirements of a modelling notation that would

enable the unique characteristics of ML processes to be expressed (Section 4). We then conclude with related work and future work in Section 5.

2. Understanding Machine Learning development and deployment

There are a few major steps in the main flow of ML processes that are widely recognized for applying ML to business applications (CRISP-DM, 1999). However, describing the major steps of ML processes alone is insufficient for expressing and understanding their intricate details and nuances. As a solution, process modelling can be used to convey the flow of these ML processes and express the deeper details of their characteristics. We draw upon work by Akkiraju et al (2018) to present a high-level model for conveying the specific processes in the end-to-end ML workflow.

Defining the ML process. In Figure 1, we visualize a high-level model of the end-to-end ML workflow. This model serves as an example of how conceptual modelling can be used to express the characteristics of the major processes in the ML workflow at a high level. The process begins with a definition of the business need. Specifically, the analyst must have a clear picture of what added value(s) an ML application can bring into the organization. The technical ML model development begins with preparing and cleaning data to be ready for use in the training step. There is an iterative cycle between training and testing to attain the desired quality for the ML model. Once the ML model is developed, it is deployed into a business application and aligned with the business case, which was defined in the first step of the ML workflow. Once deployed, the ML model becomes accessible to the end user.

Figure 1: High-level ML

Characteristics of ML and technologies. ML processes involve multiple roles with different expertise that must collaborate iteratively throughout the ML workflow. Multiple teams such as DevOps engineers and

data scientists must collaborate throughout the process. Understanding the types of interactions between teams in the ML process can be complex, as roles, goals and responsibilities may be ambiguous. Additionally, teams need to collaborate in feedback cycles at various points throughout the process in order to achieve quality objectives. This presents challenges for conceptual modeling to be able to express complex iterative feedback cycles in different time frames.

The need for systematic processes in ML workflow. Conceptual modelling can be used to design systematic processes for expressing the ambiguities and challenges presented by the unique set of qualities of ML. We propose using conceptual modelling to define and contextualize the roles, responsibilities, goals and quality criteria required in the ML workflow. In order to understand how conceptual modelling can help advance the maturity of the ML processes in an organization, it is important to analyze the benefits of conceptual modelling and understand its drawbacks.

3. Conceptual Modelling

In this section, we outline the benefits of using conceptual modelling with a special focus on improving business processes. Conceptual modelling is well-established in expressing traditional information systems methods. We discuss a type of conceptual modelling known as business process modelling. Process models have been a foundational technique for information systems analysis since the 1970s (Ross & Schoman, 1977; Yourdon, 1989). We outline the general benefits of conceptual modelling and its application in traditional information systems, and explain how these benefits can be applicable for ML process design.

3.1 Benefits of Modelling in Information Systems

Improved understanding of business processes. The benefits of conceptual modelling are primarily managerial and organizational, where research points toward an improved and consistent understanding of business processes (Indulska et al., 2009). Many non-experts don't understand ML well enough to apply it usefully. We aim to show how conceptual modelling can improve the level of understanding within organizations and enhance their ability to apply ML processes strategically.

Process improvement. Process modelling provides greater ability to improve business processes, by facilitating cost reduction, cycle time reduction, and quality improvement (Indulska et al., 2009). For example, in the ML workflow, clearly defined interactions, roles

and responsibilities can greatly enhance the quality of interactions occurring between DevOps engineers, data scientists and business domain experts at multiple points in time throughout the process.

Process automation. Process modelling offers the ability to support process automation (Indulska et al., 2009). Process automation can be applied to automate sets of data processing elements by writing code, which can be handled by the machine more efficiently than a human. For example, infrastructure deployment in ML is often run using boilerplate code that can be automated for machine processing, rather than putting the onus on a human (Hummer, Muthusamy, Rausch, Dube, & El Maghraoui, 2019). Organizations can use conceptual modelling to design their own processes to automate this step. This can be done by visually expressing the processes and data flows that encapsulate infrastructure deployment. This also allows organizations to systematically analyze which roles are best performed by humans or by machines, according to the available resources at the organization.

Process interactions and integration. Process modelling can improve communication of business processes across different process stakeholders (Indulska et al., 2009). Improved communication can improve the maturity and value of ML processes. By nature, ML requires many points of contact and ongoing iterative feedback across its processes. In the rapidly advancing field of ML, new tools are constantly being developed and introduced into projects. Tool development and enhancements also take into account usage experiences and feedback, constituting tool development lifecycles. Having well-defined conceptual models of each of these processes will enable effective understanding, analysis, and design by clarifying how they interact and how they can work together over their respective lifecycles.

Facilitating and guiding projects. Conceptual modelling can help facilitate planning, scheduling, budgeting, and managing human and other resources, such as data sources and computational requirements (Indulska et al., 2009). Using the conceptual model as a guide, organizations can design their processes tailored to their own specific needs and available resources. The conceptual model can also serve as a basis for performance monitoring and evaluation, with key performance indicators (KPIs) associated with process stages and checkpoints to measure quality criteria. These benefits can be applied to ML projects, as they require careful coordination of interactions between many activities and team members, with some activities involving a degree of creativity and experimentation.

Governance. Having a well-defined process model provides the foundations for governance structures and procedures to be defined and integrated into the process. These may include audit and control, regulatory compliance, privacy and data usage restrictions, and fairness and ethical considerations. Process models contribute to transparency and accountability. This is applicable to ML processes and their outputs which are vulnerable to many risks, including bias and anomalies in source data and algorithms.

3.2 Process Modelling for ML

In this section, we consider applying business process modelling to ML processes. The types of processes in the end-to-end workflow of ML differ from traditional business processes, but many benefits of conceptual modelling are applicable to ML. Here we outline how some of these benefits contribute to improving the understanding of ML integration and development processes within enterprise settings. These potential benefits are objectives to strive for in creating and choosing modelling languages, methods, and techniques to support ML deployment. In Figure 2 and Figure 3, we present process models as examples of how process modelling can be used to express ML processes. These models are meant to facilitate discussion based on textual description (Brethenoux, Vashisth, & Hare, 2018; Bailis, Olukotun, Ré & Zaharia, 2017) and serve as a vehicle for conveying the potential benefits of modelling ML processes.

Customized process design. Conceptual modelling allows for the specifics of the ML process to be tailored to the needs of each project according to defined criteria, such as size, complexity, data sources, the nature of the technical challenges, type of ML, talent available, and organizational maturity. An explicit model of the ML process would clearly define the process as it is designed, so that it can be communicated unambiguously. During process design, alternative models can be presented so that pros and cons can be considered and discussed. Unlike in business process models of highly routinized work, conceptual models of ML need to allow for considerable flexibility due to the experimental and iterative nature of much ML work. Having a defined process through conceptual modelling is a first step toward advancing beyond an ad-hoc level of maturity.

Applying conceptual modelling methods to ML. Given the benefits of conceptual modelling presented in Section 3.1, we present conceptual models this section to express ML processes. There are multiple types of conceptual modelling, which offer different benefits and drawbacks. In this section we will focus on process modelling. Figure 1 and Figure 2 are

preliminary models used to convey how adjustments to existing modelling techniques can foster applicability toward ML.

We present a process model in Figure 2 as one type of method to express ML processes. This example of process modelling emphasizes the sequential flow of processes in the workflow. Figure 2 expresses the major activities of the continuous learning pipeline (Bailis et al., 2017). We used material from Akkiraju et al. (2018) to provide context and insight into the process model regarding necessary inputs and steps, particularly in planning for deployment. The continuous learning pipeline (Bailis et al., 2017) is one method of deploying ML. It involves a continuous development cycle between training the ML model and iteratively updating the ML application that the end user will access in the business setting. The roles involved include the data scientist, DevOps engineer and end user/domain expert. Using this modelling notation, we indicated roles involved by using a defined swimlane in the diagrams.

Figure 2: Continuous learning pipeline. Adapted from description by Bailis et al (2017) and Akkiraju et al (2018).

This process model expresses the collaboration between the data scientist and DevOps engineer to highlight the continuous iteration of ML in an enterprise setting. The process modelling notation used in this example is the BPMN notation which is widely used for business process modelling and management (OMG, 2006).

This modelling notation presents challenges in expressing all characteristics of ML processes. It is difficult to express the details of inputs, outputs and feedback cycles. As BPMN does not require inputs and outputs, it is difficult to clarify complex data flows. This is problematic because ML processes are highly dependent on the flows of data. It is also difficult to convey how roles need to be involved at multiple points in time due to the confinement of swimlanes. In the ML process, there are multiple feedback cycles where roles are introduced and re-introduced at numerous points in time.

For comparison, we present a different modelling notation in Figure 3 as another option for expressing ML processes, based on descriptions by Brethenoux et al. (2019) and Akkiraju et al. (2018). This example emphasizes data flows as well as the inputs and outputs required at each process in the workflow. Figure 3 shows a process model covering the major activities of the operationalization cycle (Brethenoux et al., 2019). The operationalization cycle presents another method for deploying ML. We included components of work conducted by Akkiraju et al. (2018) to contextualize particular inputs and planning for deployment such as in step 3. The numbering convention used in step 3 is to represent the added context provided by material from Akkiraju et al. (2018). This contextualization allowed us to align this process model in Figure 3 with the process model in Figure 2 because Figure 3 provides a more detailed representation of what is conveyed in Figure 2. Figure 3 expresses the process of the ML model transitioning to be deployed into the business setting for an end user to access. The modelling notation used in Figure 3 is the Data Flow Diagram (DFD), which is one of the simplest for process modelling and was developed to model the flow of data through an information system (Stevens, Myers & Constantine, 1974).

We have chosen to use call-out annotations in yellow to indicate examples of the need to represent the following in a modelling notation for ML processes: automation, abstraction, system dependencies to meet quality criteria, and the user as the human in the loop. In particular, examples of system dependencies where specified quality criteria must be met can be found in process steps 2 and 3.2. In process step 2, the quality of training data must be tested by the data science team. In process step 3.2, the integration approach must be tested by the DevOps team before governance rules can be applied. The objective of this process step is to prevent the ML model from producing inaccurate results while it is being applied in the business setting. Automation systems included in the workflow were also highlighted using yellow call-out annotations where we assigned roles to the developer of the automation system. For example, step 7 conveys that the DevOps engineer is responsible for configuring an API endpoint to deliver automated analytical insights to the end user. The human in the loop is included as a call-

out annotation to highlight where a non-expert in ML can be involved in the overall workflow. For example, in step 8, the business domain expert provides domain expertise to help determine how the ML model's output provides business contributions in the business setting. Another important aspect of the textual description is the indication of team roles and expertise that are required at each step. We annotated the DFD with braces covering a series of activities associated with a role in red.

Figure 3: Operationalization cycle

3.3 Identifying Gaps in Modelling ML: Other types of conceptual modelling

Many of the issues we mentioned in this section cannot be adequately expressed using process modelling. However, there are other types of conceptual modelling methods which offer other benefits and drawbacks, such as agent-oriented modelling and goal modelling. Recently, the hiBPM process modelling notation (Lapouchnian, Yu, & Sturm, 2015) was developed to model relationships among processes, such as processes that produce plans that are executed by other processes, and processes that produce designed artifacts used by other processes. The i* framework (Yu, Giorgini, Maiden, Mylopoulos & Fickas, 2011) is a form of agent-oriented

modelling used to model dependency relationships among participating stakeholders. BIM (Horkoff et al., 2014) offers a comprehensive goal modelling framework for strategic business modelling, including quantitative indicators (KPIs).

Gaps in Process modelling. By using process models, we observed some of the general benefits of conceptual modelling discussed earlier in Section 3. These examples include uncovering ambiguities, feedback cycles, multi-team collaboration, quality criteria and systematic detection of underlying problematic phenomena such as model drift, which is not easily detected. However, we demonstrated that current methods of process modelling are not able to encapsulate all characteristics of ML processes, which were outlined in the process models and discussion in Section 3.2. In examining the characteristics of enterprise ML processes, we recognize there are new challenges that may not be easily handled by existing conceptual modelling languages.

Extending current modelling methods. Various existing conceptual modelling methods have been discussed, each offering different benefits and drawbacks. Now that we have established that different kinds of conceptual modelling present different advantages and drawbacks, we are faced with an open research question: can we determine the appropriate mix of features from existing modelling languages to address the challenges of modelling ML processes? We aim to find a suitable set of complementary modelling formalisms that would present additional perspectives that are not revealed in the perspective of the process of ML integration.

4. Conceptual Modelling for Enterprise Machine Learning

In this section we discuss the nuances of ML that current conceptual modelling methods cannot represent as easily as other traditional IT systems. We apply the general processes mentioned in Section 3 to ML and suggest solutions using conceptual modelling to bridge the gap of what current methods are missing. Specifically, we look to apply the benefits of conceptual modelling in Section 3.1 to outline how conceptual modelling can improve ML processes. A new modelling notation must be developed to allow for a more systematic method to be formalized in the modelling notation. To support this, we outline the issues and considerations in seeking appropriate language features and modelling capabilities for supporting enterprise ML.

4.1 Collaboration between roles in the ML Process

Team collaboration and diversity of roles and responsibilities. The successful deployment of ML applications is dependent on the understanding of roles and responsibilities of multiple teams such as DevOps, data science, and enterprise architecture. DevOps engineering and data science teams must collaborate in feedback cycles throughout the process of operationalizing an ML model. Understanding the types of interactions requires effective collaboration across the organization with participation from business, data science specialties, and technology services (Sculley et al., 2015). As organizations acquire and develop their capabilities in ML, they must define roles and responsibilities to coordinate activities covering end-to-end ML processes. The conceptual modelling notation will need to be able to express the collaboration between several roles with diverse disciplinary backgrounds primarily between DevOps, data science, business domain expertise and requirements engineering.

Emerging roles with ambiguous expertise. Developing and using ML algorithms often requires deep expertise in the area of data science. Compared to technical roles in traditional IT, the role of the data scientist is not yet well understood and expected responsibilities are not systematically defined to the same level of precision. In many organizations, the data scientist is expected to perform the role of understanding data requirements, which is typically conducted by the business analyst in traditional IT (Akkiraju et al., 2019). There is existing research that maps roles to responsibilities in ML, such as in work by Linden, Idoine, Hare & Brethenoux (2018). In Table 1, we adapt work done by Linden et al. (2018), Brethenuox et al. (2018), and Tapadinhas, Idoine & Kronz (2018) to present a list of various types of roles, responsibilities and expertise in the ML workflow. We do not include specific mapping between each because the names of many roles involved in ML are constantly evolving and ambiguous. For example, the Citizen Data Scientist (Tapadinhas et al., 2018) is an emerging role where the individual holds both data science expertise and business domain expertise, enabling them to translate business needs while also understanding technical data science aspects. However, this role is not highly trained to have the same skills as a data scientist, but requires enough skill to translate domain expertise in both data science and business (Tapadinhas et al., 2018). This creates a high level of complexity when trying to define roles in ML, highlighting why ML poses unique problems and deserves heightened attention compared to traditional information systems.

Using textual description alone cannot capture this level of complexity. Conceptual modelling is needed to clarify this complex mapping of roles and responsibilities to understand

how organizations should design their teams and processes when implementing ML. In order to express this variety of roles involved, we used blue curly braces in our DFD model (Figure 3) to represent the role responsible for each activity. As a more formalized solution, a new modelling notation is needed.

Roles	Responsibilities	Skillset/qualifications
Business analyst	Requirements	People skills that
Data engineer	gathering	foster collaboration
DevOps Engineer	ML model	Leadership
Data Scientist	developing	Communication
Enterprise	Data preparation	and teamwork
Architect	Model integration	Industry and
Systems	Model monitoring	business function
administrator	Quality validation	knowledge
Application		Quantitative skills
developer		General technology
Software engineer		skills such as
Business domain		infrastructure and
expert		analytics
Citizen data		
scientist		
Business translator		
Software engineer		

Table 1: Breakdown of roles and responsibilities

Dependencies during collaboration. There are dependencies which exist between roles involved with diverse expertise. Each activity in the overall workflow involves a specific role responsible for the activity and has specific dependencies on the activities preceding it. For example, as shown in Figure 3, data quality must be tested during ML deployment. Testing is needed during the approach of integrating the ML model into a usable application for end users using quality criteria before proceeding to the next step (Brethenoux et al., 2018). This level of complexity between activities cannot be expressed clearly using existing modelling notations. A systematic method must be formalized in an extended modelling notation to provide the ability to express dependencies for each role and activity.

4.2 Continuous deployment

End-to-end workflow of ML from conception to deployment. For an ML application or solution to succeed, it must integrate into many processes of the enterprise, not least the end-user business process. Furthermore, it must also be incorporated into tool building cycles and business experimentation and innovation processes. These processes feed each other and have dependencies through multiple feedback loops, even though they operate on different timing scales. The conceptual modelling notation should be able to model a process architecture, consisting of multiple intertwined processes, and the types of interactions and relationships among them.

Systematic integration of ML. In ad-hoc development of ML, models may be used for specific use cases and may be meant for limited usage. However, in order to use ML in applications within enterprise settings they must be reliable, durable, scalable and re-usable for long-term usage by non-experts in ML. To achieve production quality and scale, ML applications require development and deployment processes that are substantially different from conventional systems, including, typically: data sourcing, data and feature preparation, algorithm selection and model training, testing and benchmarking, deployment and operations management, continuous monitoring and retraining, and quality assurance and governance throughout (Akkiraju et al., 2018) To express continuous deployment, the modelling notation will need to represent the goal of being able to systematically scale the ML model repository and monitor the organization's capacity to adapt and move at the pace of the ML industry over time.

Cumulative effects over time. There are different types of cumulative effects which do not exist in traditional IT systems but must be mitigated when considering ML processes. One example is model drift: once the ML model is deployed into a business application, continuous monitoring is needed as the accuracy of training data can deteriorate over time, resulting in inaccurate ML predictions. The conceptual modelling notation must be able to acknowledge the phenomena of cumulative effects and formalize systematic processes to express how the risk may occur and what processes or controls can be put in place.

4.3 Quality criteria

Quality criteria and metrics are important to include, to be associated with specific activities or checkpoints in the process. In particular, during the operationalization cycle of ML

deployment (Brethenoux et al., 2018), there are multiple points where various types of quality criteria must be met: testing the training data quality, testing the integration approach, validating business KPIs, measuring model performance using metrics and alerts, and measuring business contribution of the model. Each of these quality criteria checkpoints require different types of inputs and roles with diverse expertise and process strategies.

4.4 Engaging with Subject Matter Experts

Human in the loop. Involving the business domain expert of the end user in the loop allows for detection measures to be put in place such as bias detection. Data scientists alone cannot develop accurate unbiased ML applications, and they need the encoding of domain knowledge (Bailis et al., 2017). Encoding domain knowledge will provide context and expertise related to the business to ensure quality criteria can be met. The conceptual modelling notation will need to be able to express a feedback cycle with a domain expert in the deployment loop offering domain expertise as feedback.

Human-computer interactions. When considering depictions of processes involving interaction with automated activities, we should ask the following questions: (1) Who owns what? (2) How do we hold machines accountable? (3) How do we set metrics for quality assurance of activities which are conducted and automated by machines? A modelling notation needs to improve capabilities for expressing accountability in automated processes, which may be difficult to track. Applying ML algorithms to applications differs from traditional IT systems in that the functionality of the algorithm itself must be abstracted from other stakeholders of the system. This is because other IT, DevOps and business domain roles must be able to interact with the system, but typically lack the knowledge in data science. As the ML model is deployed into an application, domain experts and end users should have a simplified interface to interact with the model without requiring knowledge about the technical details of the workflow (Bailis et al., 2017). The end user should have abstracted access to the ML model through an endpoint, such as an automated API (Bailis et al., 2017). Designing this level of abstraction requires an integration strategy and appropriate infrastructure. For example, the operationalization cycle requires the configuration of decision points and an integration approach to be used to develop endpoints for the end user to access the ML model using an API call (Brethenoux et al., 2018).

4.5 The need for a new modelling notation: where current methods do not work

In Section 3 we presented process models expressing ML. We demonstrated how we needed to improvise to capture the specific characteristics of ML processes but were not able to capture all elements discussed in earlier sub-sections of Section 4.

Limitations of expressing dependencies. In Figure 3, we attempted to model multiple examples of dependencies in the ML process. These dependencies included dependencies between the end user and the ML model, as well as quality dependencies. We attempted to express abstracted human-computer interactions by conveying the interaction between the end user role and the ML model. However, this existing method does not make it obvious how the abstraction has occurred for the end user, or what specific interface the end user is using to access the ML model. We were also not able to express characteristics relating to quality aspects such as quality checkpoints. We improvised by using annotations, but the DFD modelling notation needs to be extended to allow a more systematic way of expressing quality aspects. A new formalism is required in a modelling notation to be able to provide a higher level of clarity and better express dependencies.

Limitations in defining roles, responsibilities and expertise. We used curly braces in Figure 3 to represent roles to clearly define who own each process. Using curly braces was an informal modification of the original DFD modelling notation. In addition, we were not able to express the types of expertise required at each step in the overall process using both modelling notations in Figure 2 and Figure 3. In Figure 2, we were able to use the BPMN swimlanes but this rigid definition of roles limited our ability to re-introduce roles at multiple points in time during the process. In addition, current processes did not sufficiently express the type of expertise needed at each point of the overall process. Defining roles alone is not sufficient, so current process modelling methods need to be extended with systematic formalisms to clearly express roles and expertise in ML.

Limitations in capturing varying points of time. Using current process modelling methods, it proved difficult to express the concept of time. In Figure 2 and Figure 3, we were limited in our ability to express the existence of complex phenomena such as model drift and technical debt. In addition, in Figure 3 we attempted to use feedback loops to express reiterations occurring between development and deployment which would occur at multiple points in time. Using current methods, these processes are not obvious in the process models and it is difficult to analyze. Conceptual modelling must be improved with a more systematic method to formalize how this can be expressed.

Research challenges. We recognize that past research has been done regarding designing processes for the ML workflow. For example, a goal-guided approach to ML solution development was proposed in earlier work by Nalchigar and Yu (2018). However, previous work such as this is not adequate for enterprise ML because of the lack of coverage for continuous deployment and application of ML in business settings. ML must offer business value for organizations and this cannot be done without applying and validating how ML aligns with metrics to measure business contribution such as KPIs. There is a lack of research in designing processes to express deploying ML for business applications within an enterprise setting. This creates research challenges because applying conceptual modelling to ML is different from traditional information systems, creating difficulty for conducting research toward ML.

5. Conclusions and Future Work

ML is a rapidly growing technology yet the actual deployment of ML models into usable applications has yet to reach mass adoption. There has been considerable work done regarding processes expressing the development of ML models, however, there are a lack of systematic processes for deploying ML into enterprise settings. This paper discusses the model expressiveness and analytical needs that are specific to ML deployment.

In this paper we draw on literature sources to construct sample models of ML, consisting of the operationalization cycle (Brethenoux et al., 2019) and the continuous learning pipeline (Bailis et al., 2017) using the DFD and BPMN modelling notations. We explored the new processes of ML deployment and new challenges presented which differentiate ML from traditional IT systems, thus highlighting the emerging need to design new types of processes. We proposed exploring how conceptual modelling can serve as a vehicle to integrate ML in business applications.

Given these findings, future work will consist of exploring how various conceptual modelling notations compare in their ability to successfully express these needs. There is opportunity to explore other aspects of ML processes, such as goals, which require further investigation. There is an opportunity to extend and combine current conceptual modelling techniques to systematically encapsulate the end-to-end workflow of ML from conception to deployment. Future work will include modelling language development using empirical studies on industry practice.

References

- Akkiraju, R., Sinha, V., Xu, A., Mahmud, J., Gundecha, P., Liu, Z., & Schumacher, J. (2018). Characterizing machine learning process: A maturity framework. *arXiv preprint arXiv:1811.04871*.
- Bailis, P., Olukotun, K., Ré, C., & Zaharia, M. (2017). Infrastructure for Usable Machine Learning: The Stanford DAWN Project. *arXiv preprint arXiv:1705.07538*.
- Brethenoux, E., Vashisth, S., & Hare, J. (2018, July 3). How to Operationalize Machine Learning and Data Science Projects. Retrieved October 15, 2019, from <https://www.gartner.com/document/3880054?ref=TypeAheadSearch&qid=9ab75b7e9b153d3bd0def4a&q=How to Operationalize Machine Learning and Data Science Projects>.
- Horkoff, J., Barone, D., Jiang, L., Yu, E., Amyot, D., Borgida, A., & Mylopoulos, J. (2014). Strategic business modelling: representation and reasoning. *Software & Systems Modelling*, 13(3), 1015-1041.
- Hummer, W., Muthusamy, V., Rausch, T., Dube, P., & El Maghraoui, K. (2019, June). Modelops: Cloud-based lifecycle management for reliable and trusted ai. In *2019 IEEE International Conference on Cloud Engineering (IC2E'19)*.
- Indulska, M., Green, P., Recker, J., & Rosemann, M. (2009, November). Business process modelling: Perceived benefits. In *International Conference on Conceptual Modelling* (pp. 458-471). Springer, Berlin, Heidelberg.
- Lapouchnian, A., Yu, E., & Sturm, A. (2015, May). Re-designing process architectures towards a framework of design dimensions. In *2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)* (pp. 205-210). IEEE.
- Linden, A., Idoine, C., Hare, J., & Brethenoux, E. (2018, August 27). Staffing Data Science Teams: Map Capabilities to Key Roles. Retrieved October 15, 2019, from <https://www.gartner.com/document/3888468?ref=TypeAheadSearch&qid=d8a9c1b58baa53d2ab0&q=Staffing Data Science Teams: Map Capabilities to Key Roles>.

- Lukyanenko, R., Castellanos, A., Parsons, J., Tremblay, M. C., & Storey, V. C. (2019, June). Using Conceptual Modelling to Support Machine Learning. In *International Conference on Advanced Information Systems Engineering* (pp. 170-181). Springer, Cham.
- Nalchigar, S., & Yu, E. Designing Business Analytics Solutions. *Business & Information Systems Engineering*, 1-15.
- OMG, B. P. M. N. (2006). Version 1.0. *OMG Final Adopted Specification, Object Management Group*, 190.
- Ross, D. T. (1977). Structured analysis (SA): A language for communicating ideas. *IEEE Transactions on software engineering*, (1), 16-34.
- Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., ... & Dennison, D. (2015). Hidden technical debt in machine learning systems. In *Advances in neural information processing systems* (pp. 2503-2511).
- Stevens, W. P., Myers, G. J., & Constantine, L. L. (1974). Structured design. *IBM Systems Journal*, 13(2), 115-139.
- Tapadinhas, J., Idoine, C., & Kronz, A. (2018, September 21). How to become a data-driven enterprise? In a new report, Accenture Labs details a five-stage journey to data industrialization and unlocking new insights and opportunities. Retrieved October 15, 2019, from www.gartner.com/document/3890468?ref=ddrec&refval=3901863
- The CRISP-DM process model. (1999). Retrieved October 15, 2019, from <http://www.crisp-dm.org>.
- Wand, Y., Monarchi, D. E., Parsons, J., & Woo, C. C. (1995). Theoretical foundations for conceptual modelling in information systems development. *Decision Support Systems*, 15(4), 285-304.
- Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J., & Fickas, S. (2011). Social Modelling for Requirement Engineering: An Introduction. *Social Modelling for Requirements Engineering*.