WHITEPAPER

McGill

Is Machine Learning suitable to improve my process?

A guide to assess the applicability of machine learning algorithms in the manufacturing industry

Manuel Sage & Yaoyao Fiona Zhao

Department of Mechanical Engineering

McGill University, Montreal

November 2020

Version 1.0

Preface

This whitepaper proposes a guide for the work on machine learning projects in the manufacturing industry. The objective of the guide is to improve the collaboration between machine learning experts and decision makers in companies. The guide and its action steps are conceptualized to lead to early decisions regarding the applicability of machine learning for projects in manufacturing and design processes.

The whitepaper is an extract of the research conducted in the Master thesis of M. Sage. More details, in particular on the three analyzed case studies, can be found in Sage (2021) after publication.

This is the first version of the document. We intend to expand and refine the proposed guide on a continuous basis with experiences from additional case studies. For questions, comments, or suggestions, please contact us at *yaoyao.zhao@mcgill.ca*.

1. Chances and Challenges of Machine Learning in the Manufacturing Industry

Throughout the last years, machine learning (ML) has attracted the interest of decision makers in the manufacturing industry (Diez-Olivan et al., 2019), (Ge et al., 2017). Within the vague boundaries of Industry 4.0, companies seek to further improve their processes to remain competitive in challenging market conditions. A key challenge thereby is to efficiently handle the enormous amounts of data that modern companies generate. From design to production to service, all steps of an enterprise's value chain produce an increasing quantity of data. However, according to Iafrate (2018), on average only 10% of a company's data is analyzed or further processed, a phenomenon he calls "analytical rupture". The insufficient usage of data is a main motivation for the application of ML. ML algorithms are promising tools that can acquire useful knowledge from historical data in order to allow informed decisions for future data.

Despite the popularity of ML, there exist various uncertainties and difficulties business decision-makers without ML expertise experience when considering learning algorithms for an application. ML is a relatively new field with a lack of theoretical grounding and experiences on its applicability. Given an application scenario, the choice of algorithms for data analysis and predictions is rather driven by trial and error and the expert's intuition, than by common rules in literature (Wuest et al., 2016). In addition, the research on ML is quickly evolving. New techniques and potential use cases are proposed in short intervals, challenging companies to stay on the ball. Moreover, ML experts are rare and expensive. Acquiring the necessary knowledge to understand the application domain and preparing the data can take significant amount of an expert's time (Grzegorzewski and Kochanski, 2019). Those steps happen before even knowing if the outcome is an improvement compared to the existing situation or other solutions.

A ML project in industry consists of two main parties: the industrial client, host of the process in question, and the ML expert, responsible for the application of the algorithms. The key challenge of ML projects is the knowledge gap between those two parties, as represented

Figure 1.1: Strengths and weaknesses of the main parties

in Figure 1.1. The industrial client is expert on the application domain and thus able to interpret the data but has no deeper knowledge of ML. The ML expert on the other hand can apply data handling and machine learning algorithms but is not familiar with the application domain. It is striking that the strengths of one party resemble the weaknesses of the other party. Hence, a close collaboration between industrial client and ML expert are crucial for successful projects.

This whitepaper seeks to reduce the knowledge gap by proposing a guide for ML projects in the manufacturing industry. The guide consists of action steps, rules, and guidelines and is designed to lead to initial decisions on the applicability of ML for industrial processes. The guide was developed in the work of Sage (2021), summarizing the experiences made from three case studies in design and manufacturing operations at an original equipment manufacturer (OEM) and a small and mid-size enterprise (SME) in Canada. Before the action steps of the guide are presented in Chapter 3, the next Chapter introduces background information on ML and data handling for ML.

2. Background

2.1 Machine Learning

Machine learning is a subcategory within the broad field of artificial intelligence (AI) that intends to obtain *intelligence* by learning from experience. This experience is represented by the data, serving as input to ML algorithms. ML has three main subcategories with different working principles: supervised learning, unsupervised learning, and reinforcement learning. This work focuses on supervised learning, the most popular group of algorithms for manufacturing processes (Cadavid et al., 2019). For supervised learning, the training data consists out of input vectors x and their corresponding target vectors y. Together, they form labeled pairs (x, y) of training instances. Depending on the application area, x takes different forms and could represent, for example, the pixel values of an image, word counts in a document, or the measures of sensors on a production line. Regarding y, a continuous output space is called a regression problem. If the output space is discrete, the algorithm tries to learn a classification problem. Popular are binary classification tasks, such as *part okay* vs. *part not okay*, but multiclass (y contains more than two elements) and multilabel (multiple classes can be assigned to one instance) classifications are widely used as well. The target of a supervised task is learning a function $f : x \mapsto y$ that maps the input space into desired values of the output space and describes a relation between input and output. After learning, the mapping f outputs a prediction y^* for a query x^* (Jordan and Mitchell, 2015), (Bishop and Nasrabadi, 2007).

Supervised learning comprises various algorithms belonging to different technique families. In the work of Cadavid et al. (2019) literature on machine learning applications in production planning and control is reviewed. After analyzing 93 publications, the authors rank families of algorithms by their frequency. Of the eight most popular families, five belong to supervised learning. Table 2.1 briefly introduces an algorithm for each of the five technique families.

Table 2.1: Selected supervised learning algorithms

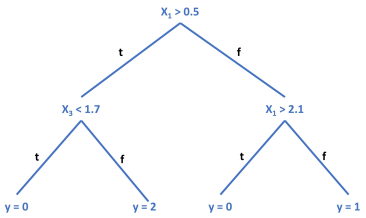
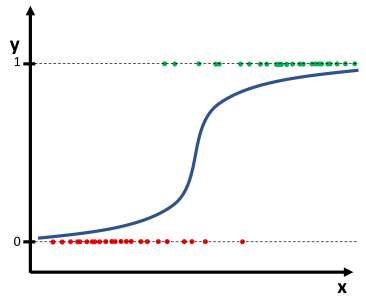
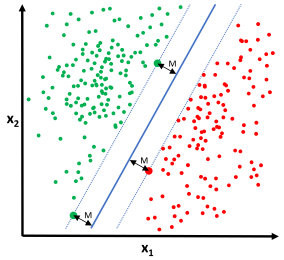
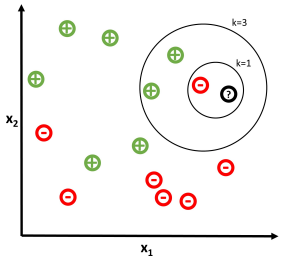
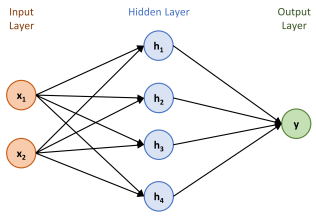
Decision Tree (DT) (Hastie et al., 2001)		Family: Tree-based Models
DTs are used for classification and regression tasks. With binary decisions on feature values on every node, they partition the input space into rectangles. Each rectangle (the "leaves" of the tree) represents a prediction.		+ Fast/easy to construct + Interpretable (set of if-then rules) + Good outlier handling – Not all functions can be expressed with DTs – Sensitive to small changes in data
Logistic Regression (Shalev-Shwartz and Ben-David, 2014)		**Family: Regression**
Discriminative approach for classification tasks. Inputs a linear regression of shape $\alpha = w_0 + w_1x_1 + \ldots + w_mx_m$ (with w_0 as bias term) into the logistic function $\sigma = 1/(1 + exp(-\alpha))$. $\alpha = 0$ represents the decision boundary. Learns the weights via likelihood maximization and gradient descent.		+ Interpretable (probabilistic output) + Good performance in many binary classification tasks – Discriminative models allow less structural assumptions about data – Poor performance on non-linear decision boundaries and multiclass classification
Support Vector Machines (SVM) (Bishop and Nasrabadi, 2007)		**Family: Support Vector Machines**
Discriminative approach, typically for classification. Divides the feature space into classes with a line (or hyperplane for more than 2 features) and maximizes the margin towards the closest datapoints, called support vectors.		+ Effective in high dimensional feature spaces + Convex optimization leads to unique solution – Low interpretability since there is no probabilistic explanation
k-Nearest Neighbor (kNN) (Bishop and Nasrabadi, 2007)		**Family: k-Nearest Neighbors**
Instance-based learning for regression and classification. Given a query, this approach uses a pre-defined distance metric (e.g. Euclidean) to identify the k-closest training instances by feature values, k is a tuneable parameter. Labels by majority vote for classification or averaging for regression.		+ High interpretability + No computational expenses for learning – Every query requires computation – Weak performance in high dimensional feature spaces – Performance highly depends on choice of distance metric
feed-forward Neural Network (FFNN) (Hastie et al., 2001)		**Family: Neural Networks**
Layers of neurons with nonlinear activation functions connect inputs and outputs. The size and number of layers are tuneable parameters. The weights of each neuron are determined via gradient descent using the backpropagation algorithm.		+ Ability to learn complex input-output relationships – Difficult training procedure (hyperparameter tuning) – High computational expenses for training

Table 2.2: Data handling for machine learning

Data preparation	Data preprocessing
- Collect data from historical database	- Remove outliers, anomalies & errors
- Sample & variable selection	- Handle missing values
- Explore linear & nonlinear relationships	- Apply scaling or normalization

2.2 Data for Machine Learning

ML algorithms learn from experience, represented by the data they are trained with. Consequently, techniques to analyze and set up data for ML are a crucial aspect of every use case. Landset et al. (2015) distinguish three possibilities to tune ML applications: the data source, the model, and the model's implementation. While according to Cognilytica Research (2020) over 80% of the overall project time is spent on the data, ML research often focuses on improving the model (Breck et al., 2019). On application level, the data-related work is highly project dependent. In literature, definitions and proposed methodologies on data handling vary. Ge et al. (2017), divide the data handling for ML into two areas: data preparation and data preprocessing. Data preparation summarizes the initial work to acquire the relevant data of an application and gain an overview of its characteristics. Data preprocessing describes techniques to improve data quality, such as data cleaning or scaling. Table 2.2 provides an overview of typical action steps for both terms (Ge et al., 2017).

At first, data preparation consists of collecting historical data that is often scattered across different databases of a company (Grzegorzewski and Kochanski, 2019). As not all the collected data might be relevant for the objective of the project, the next step is to eliminate irrelevant information by choosing meaningful subsets of the dataset and its features. Exploring linear and nonlinear correlations in the data helps to augment the understanding of the dataset and further reduce its size. Once the data matrix that most ML algorithms require is obtained, data preprocessing techniques allow to further improve data quality. Typical procedures include the removal of outliers or anomalies, the handling of missing values, and scaling or normalizing of features (Wujek et al., 2016), (Ge et al., 2017).

3. Recommended Approach to Adopt ML

A typical process flow for ML projects in industry is visualized in Figure 3.1. The first steps are an analysis of the current situation and the derivation of a target for the entire endeavor. Then, an overview of the available data should be obtained. This allows a first decision on the feasibility of the project. If continued, changes to the current process or the data acquisition system may be required. Otherwise, the actual ML work consisting of model selection, training, and hyperparameter tuning can be performed. At last, based on the performance of the developed model, the final decision is made whether to implement the ML model in the industrial environment or reject it and select either another solution or keep the process unchanged. Although both parties are ideally involved in all steps, the primary responsibility depends on the work stage and shifts between client and ML expert.

This works aims to facilitate the initial decision of machine learning projects by providing a methodology that is extended with lessons learned from three case studies. Figure 3.1 shows the three proposed action steps preceding Decision I:

1. Analysis of current situation
2. Definition of the target
3. Analysis of data

The following three sections will present each of the action steps and explain their importance in order to quickly reach a decision on the applicability of ML. In addition, the lessons that were learned from the conducted case studies are introduced.

3.1 Analysis of Current Situation

When considering ML as technique to optimize an existing process, analyzing the current situation is the first important action step. The work conducted by both parties should go beyond a brief description of the existing process and mainly focus on two aspects:

1. How does the flow of information look like?
2. What is the deficiency of the current process?

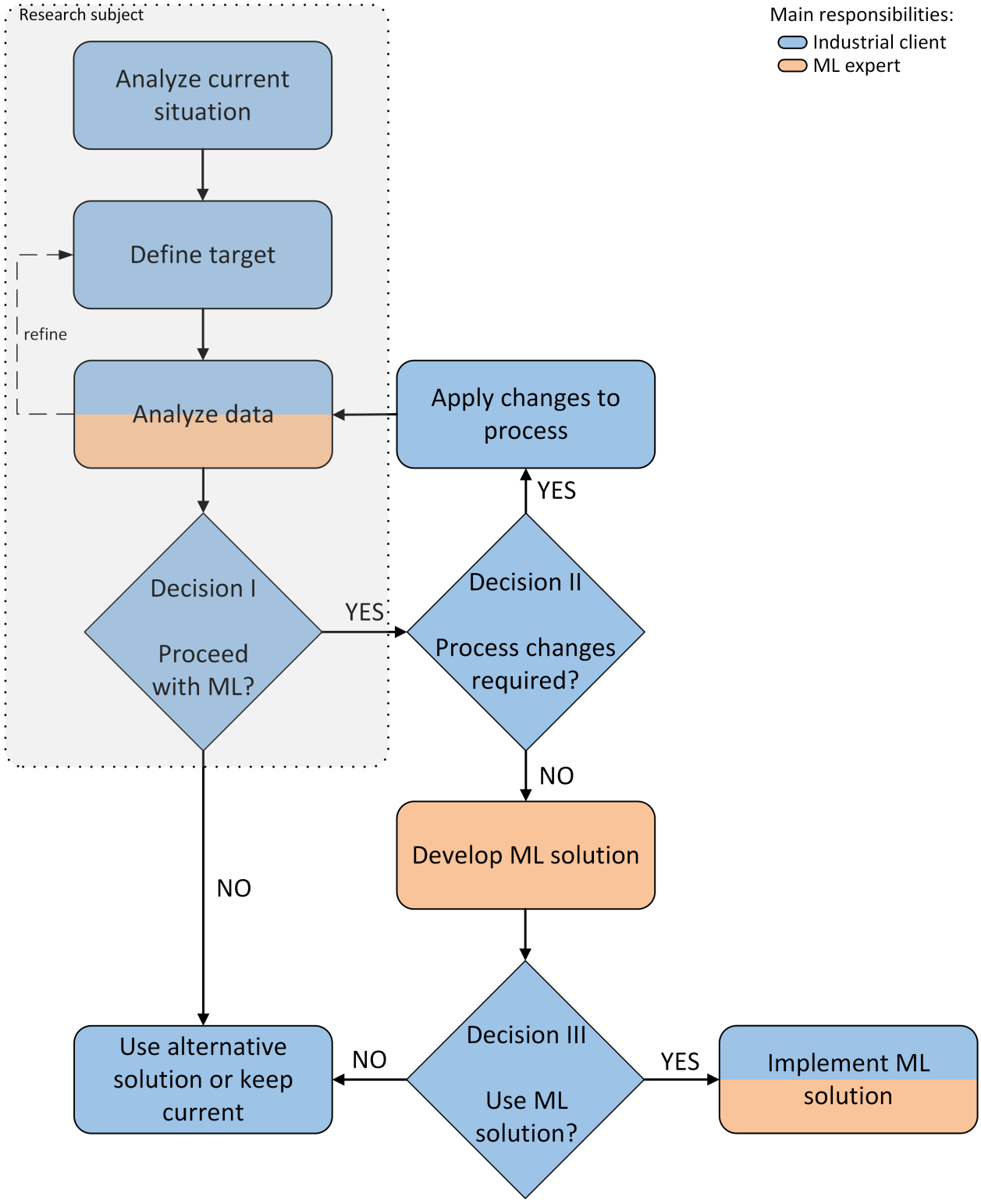


Figure 3.1: Process flow for ML applications

Due to the relevance of data for ML, precisely revealing the flow of information throughout a process will facilitate all subsequent work on the project. A suitable tool to visualize information flow are flow diagrams. The second aspect is to address the shortcomings of the current process that motivate a company to consider ML as new solution. For the industrial customer, this paves the way for the following definition of the project targets. From the perspective of the ML expert, acquiring a comprehensive understanding of the project's domain allows to interpret the results of ML algorithms and avoids mistakes during the development of a solution.

3.2 Target Definition

Deriving the target of a potential ML use case has two components: Defining the overall goal of the project and defining the resulting requirements on the learning algorithm. For the overall goal, a clear picture of how the new, modified process should look like is necessary. The target definition is supposed to describe the desired integration of ML into the current system. If there exists more than one objective, specifying an order of importance helps to set priorities.

For the second component, the requirements on the learning algorithm, the relevant questions are:

- What should be predicted and how should the output look like?
- What is the consequence of a wrong prediction?
- What are the requirements on prediction performance?

The answers to these questions have significant influence on the subsequent ML work. The desired output, e.g. a continuous number for a regression problem, determines an initial selection of candidate algorithms. Performance specifications can help to choose the right algorithm, define the desired quality and quantity of the training data, or simply serve as a reference for the ML expert during training. Generally, defining the target and analyzing the data might be mutually influential and not strictly sequential. Some researchers suggest

target definition after data analysis (Wuest et al., 2016). The reason to first define the objective is to utilize the project's goal as a guide throughout the data analysis. However, the results of the data analysis might influence the target definition and require a refinement. Understanding the intention of the project enables the ML expert to search for similar applications in literature. Moreover, upcoming work on learning algorithms can be grouped into tasks. In the survey of Cadavid et al. (2019) a framework is presented that categorizes ML-related work in industries into 11 activities. This framework can serve as a basis to decide what type of ML activity is required.

Lessons learned 1: A thorough definition of the objectives including a clear picture of the target process is crucial.

Although the target definition seems to be a logical step of every project, the experiences made from case studies in the manufacturing industry show that incompletely defined objectives negatively influence the work on ML application. As described above, ML projects requires a clear picture of the target process. Else, the initial decision on the applicability of learning algorithms will be delayed.

Lessons learned 2: For surrogate models, detailed performance requirements are essential and should be determined ahead of data generation.

This lesson learned originates from the developed of a surrogate model for computationally expensive simulations. The typical methodology for surrogate modeling is to first generate a dataset using the process one seeks to replace. Then, this dataset is used to train a ML algorithm. The utility of a surrogate model depends on its performance, the performance itself partially depends on the size of the dataset. Therefore, it is advisable for surrogate models to determine the required performance ahead of the data generation. On the one hand, this allows to generate more data if the model is underperforming and on the other hand, if all requirements are met, to stop data generation and save time and resources.

3.3 Analysis of Available Data

After familiarizing with the present situation and defining the project's objective, the data analysis begins. Exploring and evaluating the available data is the most time-consuming process step and requires a high expenditure from both involved parties. For the industrial customer, analyzing data extends the knowledge about the own process. Simultaneously, collaborating with the ML expert helps understanding the requirements of learning algorithms on data. This increased awareness of current and target state facilitates an early assessing of the chance of success of a project. The proposed methodology for data analysis is successively answering the questionnaire below. It consists of six questions pertaining to the data in general, its quantity, and its quality. The suggested methods for each question are described in the following.

- General: What data is available?
- General: Is the data generation accessible and modifiable?
- Quantity: How much data is available?
- Quantity: Is there other data that could be made available?
- Quality: Is the data consistent and balanced?
- Quality: What part of the data is relevant regarding the objective?

What data is available?

To answer this general question on the type of data, the work of two survey papers on ML applications in industry is used. Tao et al. (2018) classify data generated by manufacturing processes into five categories: management data, equipment data, user data, product data, and public data. Cadavid et al. (2019) expand this scope by adding a sixth class: artificial data. Figure 3.2 displays the whole framework including a brief description of each category. The data of a project can consist of multiple categories.

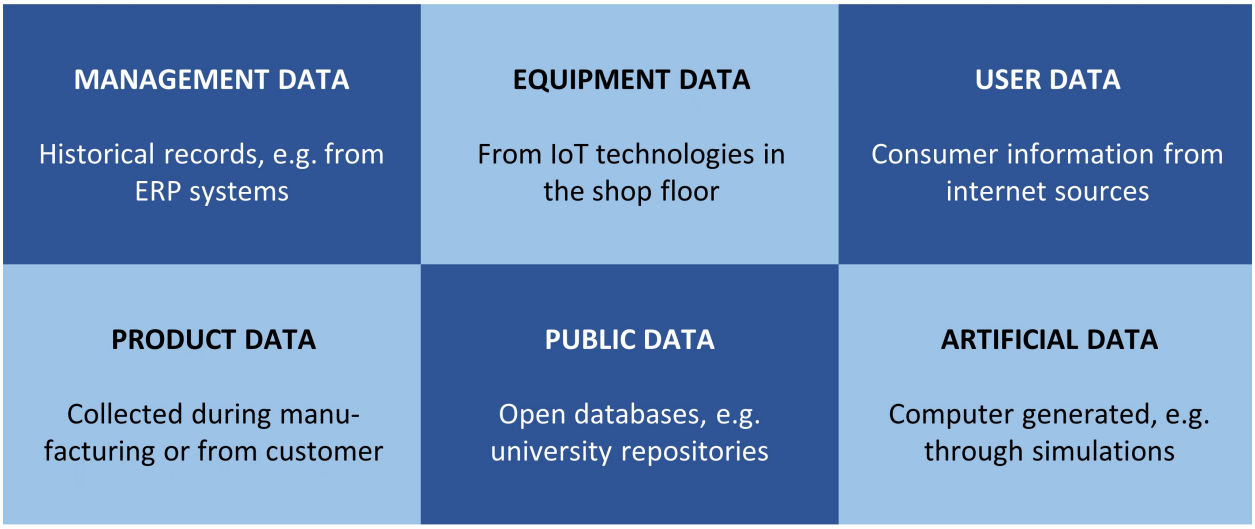


Figure 3.2: Categorization of data for ML (Tao et al., 2018), (Cadavid et al., 2019)

Is the data generation accessible and modifiable?

This question was added to the questionnaire due to the experience made from the conducted case studies. Two lessons learned pertain to the process of data generation and raising the question ensures that the developed guide takes the findings into account.

Lessons learned 3: The ability to access and influence the data generation process increases the prospect of successfully implementing a ML solution.
Some ML projects depend on fixed datasets, composed of historical instances from a data generation process that can not be altered for the sake of the project. In this case, deficiencies in the dataset quickly lead to an end of the project. Consequently, it is beneficial for ML applications to receive access to the data generation. Thereby, size, composition, and structure of the dataset can be designed matching the objectives of the project and the demands of ML algorithms at the same time. Especially the ability to generate more data if needed is an advantage that facilitates the development of an accurate ML model. As an example, a ML project that seeks to predict product failure on a production line (i.e. part okay vs. part not okay) could solely be based on existing data of the production line. Alternatively, to improve the chances of success for the project, the company could install additional sensors to obtain different data.

Lessons learned 4: If data is specifically created for the ML model, high attention must be paid to configuration of the data generation pipeline.
As described above, obtaining access to the data generation process is advantageous. However, deciding on the parameters that define data generation also adds an error possibility. This lesson learned is a consequence of a wrongly defined data generation pipeline in one of the conducted case studies that led to a significant delay of the project. Hence, to improve the flow of ML projects, data generation pipelines must be carefully configured.

How much data is available?

Quantifying a dataset has various aspects. Considering the common data matrix that ML algorithms take, the number of rows corresponds to the instances while the number of columns corresponds to the features. Both components are of interest for ML as well as the time span of data generation. For example, data acquired from a process during a few weeks might not be sufficient to represent seasonal effects and require a longer data collection.

Is there other data that could be made available?

The quantification of data potentially leads to the conclusion that the current dataset is not large enough for the intended experiments. Alternatively, despite being large enough, qualitative constraints in the data found at a later time might require more and other information. It is therefore advisable to consider collecting more data than initially available at an early stage of the project. The accessibility of data might impose another obstacle. In industry, security concerns or limitations in the IT infrastructure can restrict or delay the access to data (Wuest et al., 2016). Possible measures are requesting access to databases, cooperate with other parties involved in the process such as suppliers, or resume the data generation process.

Is the data consistent?

A simple yet effective tool to answer this question is plotting variables via bar charts, histograms or scatterplots. For continuous variables, scatterplots indicate the distribution of

instances over a discrete set of values. Alternatively, the possible values can be grouped into segments and plotted as categorical values, i.e. as frequency counts over classes. The visualization allows to reveal inconsistencies in both inputs and outputs of a dataset that could affect the performance of a ML algorithm. Examples for data inconsistencies are outliers or class imbalances. While outliers are a few datapoints with values far off the remaining instances, class imbalance refers to the situation in which significant differences between the frequencies of classes is observed. In fact, both problems describe data sparsity, i.e. a lack of data points in specific areas of the dataset that might influence predictions negatively.

Another potential technique to analyze the consistency of the dataset is the principal component analysis (PCA). PCA is a statistical process for dimensionality reduction that transforms correlated variables into a new, orthogonal basis (Jolliffe, 2002). While there exist multiple motivations to apply PCA, in terms of data analysis a reduction of the dimensionality enables data visualization. High-dimensional datasets (> 3 variables) cannot be graphically displayed as a whole. Analyzing variable by variable helps to identify inconsistencies as described above. However, this procedure is time consuming and does not reveal patterns in the data. Instead, performing a PCA with three or less dimensions allows to visualize the data in one figure. Data clusters, patterns, and outliers can thereby be detected at one glance.

Lessons learned 5: Significant changes to the process in question potentially cause unusable historical data.

This finding stems from two of the three conducted case studies in Sage (2021). In manufacturing companies, the data interesting for ML project is often generated by production processes, for example by sensors on an assembly line or by management data regarding the allocation of resources. To learn the underlying rules of a process, ML algorithms require considerable amounts of data. In most cases, the amount of available data is proportional to the duration of the data generation. However, industrial processes face constant changes and barely remain unaltered for longer periods. It is therefore advisable, to inspect if the whole dataset was generated under comparable circumstances as changes to the process might reflect on the data.

Lessons learned 6: Experiments on subsets of the whole dataset can alleviate data sparsity and reduce the time needed to reach a first conclusion on ML applicability.

Rolling out a large-scale project with many features, especially high-cardinality categorical features, complicates data handling. Additionally, data imbalance can increase the sparsity of the feature space and affect the performance of learning algorithms. Reducing the dimensionality by choosing a subset of the dataset with comparably many instances can thus serve as a workaround until the feasibility of ML is evaluated.

What part of the data is relevant?

Besides inconsistencies, data collected in industry many times includes noise and irrelevant information (Wuest et al., 2016). Consequently, determining the relevance of features regarding the project's objective is of great importance in order to improve predictions, accelerate model training, or gain insights into the process. There exist numerous approaches to quantify the relevance of data. Since the approaches differ in the type of analyzed correlations, as well as their strengths and weaknesses, it is not advisable to rely on the results of a single technique. Table 3.1 introduces three techniques that are commonly used to assess data relevance as well as Python libraries that enable their implementation.

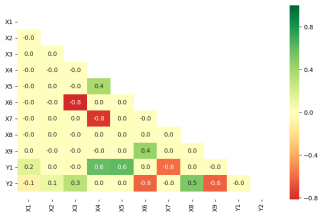
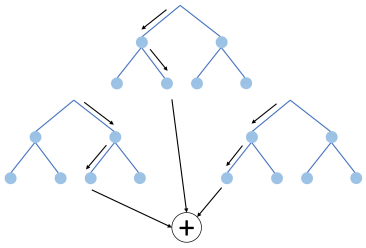
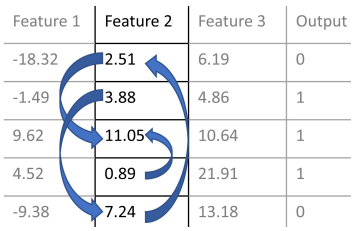
Additional Lessons Learned

Two additional findings impeding the application of ML were made that do not belong to a specific question of the data analysis:

Lessons learned 7: If the relation between input and output is known, the application of ML is not reasonable.

In industry, ML is frequently misunderstood as a tool primarily for automation. Instead, (supervised) learning algorithms should be seen as black box functions that take historical data to learn a relation between inputs and outputs. The trained algorithm can then indeed be deployed for automation purposes, but not in all automation projects ML is a reasonable

Table 3.1: Popular techniques to evaluate the relevance of data

<table>
<tr><td colspan="2">Correlation Heatmap</td><td>Libraries: Pandas / seaborn</td></tr>
<tr><td>Calculates linear correlations, e.g. using Pearson correlation coefficient, between selected variables in the dataset. Helps to find features that are (positively or negatively) correlated to each other or to the target variable.</td><td>  </td><td>+ Fast computation
+ Good overview over lin. correlations in whole dataset
– Captures only lin. correlations
– Requires encoding for categorical features</td></tr>
<tr><td colspan="2">Tree-based feature importance</td><td>Library: scikit-learn</td></tr>
<tr><td>Fits a tree-based model (e.g. random forest) on the dataset. Measures each features' ability to split the dataset by accumulating the improvement of the split-criterion for a feature over all trees in the forest (Hastie et al., 2001).</td><td>  </td><td>+ Easy to implement, fast computation
+ Captures non-linear relationships
– Struggles with high-cardinality categorical features
– Disregards interdependences between features</td></tr>
<tr><td colspan="2">Permutation importance</td><td>Library: scikit-learn</td></tr>
<tr><td>Takes a preselected ML model and trains it on the dataset. Randomly permutes the values of a feature, retrains the model and measures the effect on the performance. High decrease of performance indicates high feature importance. Repeats procedure multiple times to obtain more stable results.</td><td> 
</td><td>+ Captures non-linear relationships and interdependences between features
+ Takes account for one-hot-encoded categorical features
– Dependent on choice of ML model
– Longer computations for reliable results</td></tr>
</table>

choice. If the relation between inputs and outputs is well known and could be described e.g. via if-then statements, ML algorithms would seek to learn already known relations. In other words, the process would unnecessarily rely on predictions were clear answers are possible.

Lessons learned 8: If the data generation of a process comprises characteristics one seeks to replace, the application of ML is not reasonable.

ML is a popular approach to replace decisions made by humans. When multiple factors influence a decision, humans struggle to keep the overview or require time to evaluate the situation. Trained on historical data, a ML algorithm can output informed decisions in a fraction of the time a human needs. However, the quality of the algorithm's decision highly depends on the training data. The data used for training might comprise characteristics of

former decisions that are intended to be replaced by ML. In this case, instead of learning new rules for decision making, the algorithm learns to reproduce previous decisions. This finding is best illustrated with an example from tool life predictions. Instead of relying on human experience to change tools for milling and turning operations, a company wishes to implement a learning algorithm. The observed problem is that most tool changes in the provided dataset were conducted due to human experience. Hence, any algorithm trained on this dataset learns to reproduce human decision making instead of the actual goal, a better exploitation of tool life.

4. Conclusions

4.1 The Final Guide

The result of the research conducted in Sage (2021) is a proposed methodology for ML projects extended by the findings made on three case studies. The final guide that combines action steps and lessons learned is displayed in Figure 4.1.

The first step of the modified guide is the analysis of the current situation by visualizing the information flow and determining the shortcoming of the process in question. Thereby, the incentive to apply ML is revealed, which facilitates the definition of targets in the next step. The target definition has two aspects: specifying the goals of the overall project and those specifically related to ML. Defining ML goals comprises defining the required ML activities, prediction format, performance, as well as evaluating the consequences of wrong predictions. The experience from case studies showed that a thorough target definition is crucial to quickly assess the applicability of ML. Furthermore, if data is generated for a project, the performance requirements should be determined ahead. The third step, data analysis, is conducted by answering the six questions of the questionnaire. The first two questions are of general nature and target the available data types and the access to the data generating process. Question three and four address the quantitative aspects while question five and six focus on the quality of the data. Besides guidelines that serve as additional information for the questions, the data analysis also includes two important rules that describe situations in which the application of ML is not reasonable due to characteristics of the data. Finally, the guide concludes with the decision whether a project should go ahead with a ML solution or not.

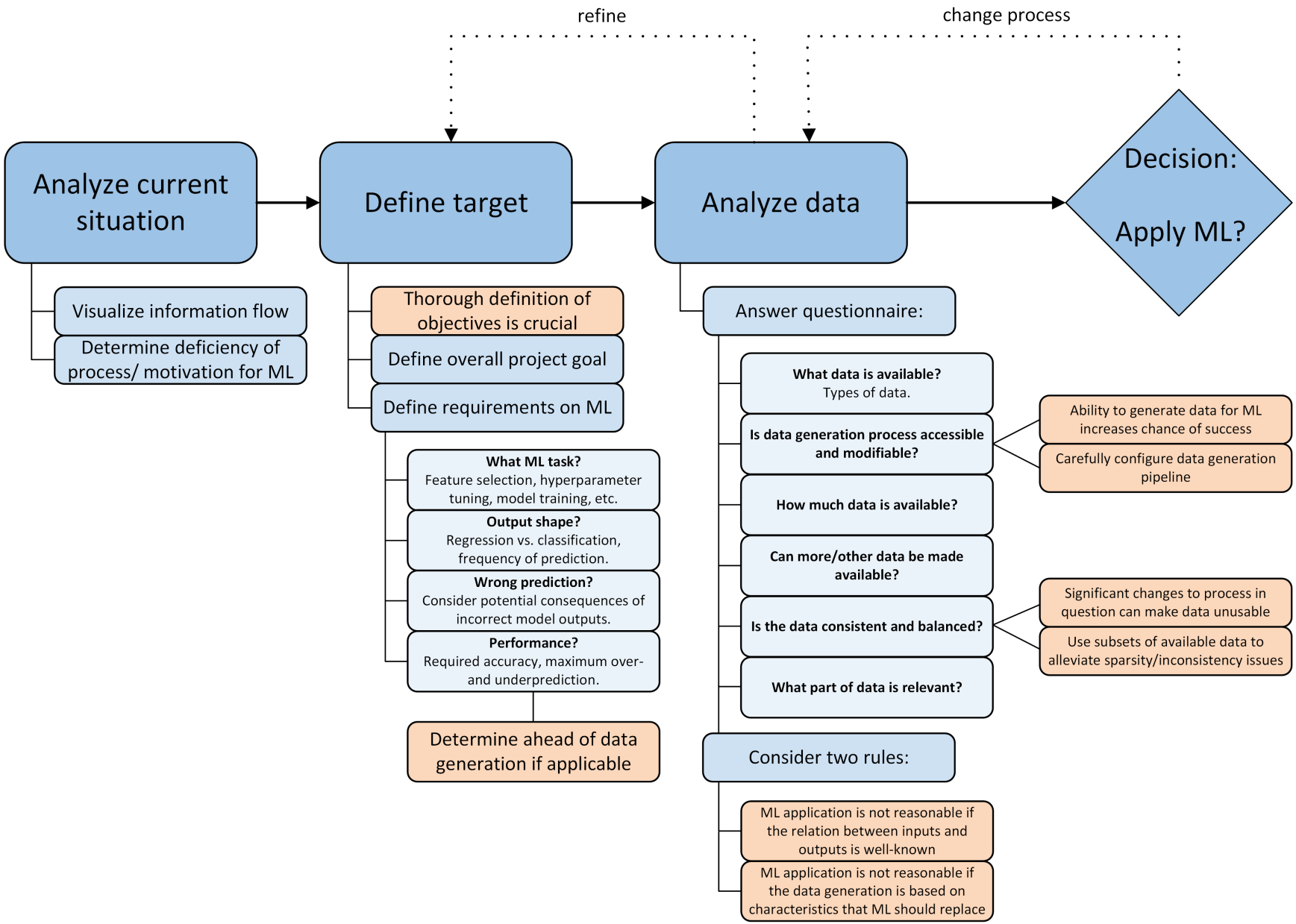


Figure 4.1: Final ML application guide with action steps in blue and lessons learned in orange

4.2 Further Observations

In addition to the guide in Figure 4.1, three key observations that were made during this study are summarized below:

- **Observation 1:** ML projects in the manufacturing industry should not start with data analysis or coding work. Although analyzing the current situation and defining the targets might seem overly simple, they are of great importance in order to allow an early decision on a project's feasibility and should not be omitted. Ideally, all steps of the presented methodology are carried out in close collaboration between both main parties, industrial client and ML expert. The first two steps predominantly rely on the knowledge of the industrial client. If the project is simply handed over to the ML expert, e.g. after a kickoff meeting, the ML expert might not be able to acquire the needed domain knowledge. A comprehensive analysis of the current situation and a thorough definition of all objectives close this knowledge gap, help to avoid misunderstandings, and structure the subsequent work on the project.

- **Observation 2:** ML projects require more time than expected. All case studies conducted for this research exceeded the initial time frame of 4-6 months. Aspects that caused delays were:

  - A lack of computational resources for data generation, data analysis, or training and testing different ML algorithms.
  - Waiting time until all relevant data is accessible, either due to manual efforts in creating a dataset or due to permission restrictions.
  - Deficiency of the initial target definition, resulting in redundant work and a refinement of the objectives.

- **Observation 3:** Regarding ML applicability, design processes have an advantage over manufacturing processes. This observation is deduced from different results obtained from the case studies. A possible explanation for this observation are two advantages of design processes:

- Design processes are highly or even fully digitalized and based on the usage of different design and simulation software. Consequently, all information is available digitally, which facilitates the data related work when establishing a ML solution. The data of manufacturing processes on the other hand, is often scattered across different production components and stored in different data formats. Additionally, manufacturing data might not be accessible from a central IT system or biased if content is manually created or transferred, e.g. on the shop floor.
- Designing a product requires fewer involved parties and process steps than producing it. This improves the comprehensibility of design data. Besides, adjusting a design process to generate suitable data for ML is more realizable than adjusting a manufacturing process. This holds true for the implementation of a ML solution as well.

Bibliography

C. M. Bishop and N. M. Nasrabadi. Pattern recognition and machine learning. *J. Electronic Imaging*, 16:049901, 2007.

E. Breck, N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich. Data validation for machine learning. In *Conference on Systems and Machine Learning (SysML). https://www. sysml. cc/doc/2019/167. pdf*, 2019.

J. P. U. Cadavid, S. Lamouri, B. Grabot, and A. Fortin. Machine learning in production planning and control: A review of empirical literature. *IFAC-PapersOnLine*, 52(13):385–390, 2019.

A. Diez-Olivan, J. Del Ser, D. Galar, and B. Sierra. Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0. *Information Fusion*, 50:92–111, 2019.

Z. Ge, Z. Song, S. X. Ding, and B. Huang. Data mining and analytics in the process industry: The role of machine learning. *IEEE Access*, 5:20590–20616, 2017.

P. Grzegorzewski and A. Kochanski. Data preprocessing in industrial manufacturing. In *Soft Modeling in Industrial Manufacturing*, pages 27–41. Springer, 2019.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

F. Iafrate. *Artificial intelligence and big data: The birth of a new intelligence*. John Wiley & Sons, 2018.

I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, NY, USA, 2002. ISBN 978-0-387-95442-4. doi: 10.1007/b98835.

M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

S. Landset, T. M. Khoshgoftaar, A. N. Richter, and T. Hasanin. A survey of open source tools for machine learning with big data in the hadoop ecosystem. *Journal of Big Data*, 2 (1):24, 2015.

C. Research. Data preparation & labeling for ai 2020. `https://www.cognilytica.com/2020/01/31/data-preparation-labeling-for-ai-2020/`, 2020. Accessed: 2020-06-12.

M. Sage. *Assessing the applicability of machine learning in manufacturing and design operations (under review)*. Master Thesis. Department of Mechanical Engineering, McGill University: https://escholarship.mcgill.ca, 2021.

S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

F. Tao, Q. Qi, A. Liu, and A. Kusiak. Data-driven smart manufacturing. *Journal of Manufacturing Systems*, 48:157–169, 2018.

T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben. Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1): 23–45, 2016.

B. Wujek, P. Hall, and F. Güneș. Best practices for machine learning applications. *SAS Institute Inc*, 2016.