



# Impact of material data in assembly delay prediction—a machine learning-based case study in machinery industry

Fabian Steinberg<sup>1</sup> · Peter Burggaef<sup>1</sup> · Johannes Wagner<sup>1</sup> · Benjamin Heinbach<sup>1</sup>

Received: 26 July 2021 / Accepted: 16 January 2022 / Published online: 12 February 2022  
© The Author(s) 2022

## Abstract

Designing customized products for customer needs is a key characteristic of machine and plant manufacturers. Their manufacturing process typically consists of a design phase followed by planning and executing a production process of components required in the subsequent assembly. Production delays can lead to a delayed start of the assembly. Predicting potentially delayed components—we call those components assembly start delayers—in early phases of the manufacturing process can support an on-time assembly. In recent research, prediction models typically include information about the orders, workstations, and the status of the manufacturing system, but information about the design of the component is not used. Since the components of machine and plant manufacturers are designed specifically for the customer needs, we assumed that material data influence the quality of a model predicting assembly start delayers. To analyze our hypothesis, we followed the established CRISP-DM method to set up 12 prediction models at an exemplary chosen machine and plant manufacturer utilizing a binary classification approach. These 12 models differentiated in the utilization of material data—including or excluding material data—and in the utilized machine learning algorithm—six algorithms per data case. Evaluating the different models revealed a positive impact of the material data on the model quality. With the achieved results, our study validates the benefit of using material data in models predicting assembly start delayers. Thus, we identified that considering data sources, which are commonly not used in prediction models, such as material data, increases the model quality.

**Keywords** Production control · Assembly · Prediction methods · Lead time reduction · Machine learning · Supervised Learning · Classification algorithms · Data analysis

## 1 Introduction

Manufacturing companies are challenged to succeed in dynamic international markets requesting high-quality products, flexibility, on-time delivery, and a reasonable cost structure [1–3]. Here, short delivery times and adherence to delivery dates is a key factor to differentiate from competitors. A typical example of this is the machine and plant manufacturing industry producing complex products consisting of numerous components [4, 5]. Many of these components are customized enabling tailor-made solutions for the customers' requirements. In general, the manufacturing process of machine and plant manufacturers starts with

the design of the product and components, followed by the production planning, the purchasing of raw materials, and the production process to manufacture the individual components needed in the subsequent assembly process. In parallel to the production process, the components required in the assembly are also purchased from suppliers. The task of the assembly is to assemble a product of higher complexity with predefined functions with a certain quantity of components in a partly multi-stage process in a given time [6]. Furthermore, in the assembly many material flows converge, leading to a high potential of delays [7]. Thus, an essential factor for meeting the delivery date is the start of the assembly on time and a prior timely supply of the components needed for assembly. Subsequently, components produced in the processes upstream of the assembly have a direct influence on the performance of the assembly process. Assuming that all components are required to start the assembly process, even a single component supplied behind schedule will lead to a delayed start of assembly [8]. To meet delivery dates, it would be

✉ Fabian Steinberg  
fabian.steinberg@uni-siegen.de

<sup>1</sup> Department of Mechanical Engineering, Chair of International Production Engineering and Management (IPEM), University of Siegen, Siegen, Germany

helpful to predict these delayed individual components (we call them assembly start delayers) in the early stages of the manufacturing process. Based on an early prediction, measures such as close communication with the supplier, extra shifts to temporarily increase production capacity, or utilizing a different workstation can be derived to speed up the manufacturing process and thus, prevent assembly start delays.

With the increasing development of machine learning (ML) and the availability of big data, ML-based prediction models are becoming more and more established in the field of production planning and control. ML models have already been successfully applied to predict lead times of manufacturing processes [9] and to predict assembly start delayers [10]. Our previous research already showed that predicting assembly start delayers utilizing a binary classification is the recommended approach and outperforming approaches utilizing a lead time prediction to identify assembly start delayers [10]. Furthermore, when setting up and training a prediction model, the used data model has a central influence on the model quality of the prediction model [11, 12]. For example, Burggraef et al. [9] have already discovered that material data defining all characteristics of the product to be manufactured such as geometric specification, weights, or the material itself are rarely used in ML-models to predict lead times.

Looking at the business process of a machine and plant manufacturer in contrast to the usage of material data in prediction models, it is noticeable, that the products of machine and plant manufacturers are typically tailor-made for each customer need [13, 14]. As the product's characteristics strongly influence the needed processes for its manufacturing [15], the design phase of machine and plant manufacturers including the material data specified within the design phase also has a non-negligible influence on the manufacturing process. Consequently, we assume that the usage of material data in a model predicting assembly start delayers has an impact on its model quality. Nevertheless, material data are currently only rarely used in prediction models. But, so far, a validation that the material data influence the respective model quality has not yet been performed.

Thus, our manuscript aims to set up an ML-based model for the prediction of assembly start delayers and to analyze and systematize the influence of material master data on the model quality. As a research method, we apply a case study at a machine and plant manufacturer. With the achieved results, our paper provides two main contributions:

- We developed a model to predict assembly start delayers utilizing a machine learning classification approach.
- We identified that material data influence the model quality of a model predicting assembly start delayers. However, there was only a slight influence.

Our paper is structured as follows. Section 2 first introduces the product structure and manufacturing processes in an

engineer-to-order environment as well as available approaches to identify and predict assembly start delayers. Section 3 elaborates on our approach to quantify the impact of material data on the model quality predicting assembly start delayers utilizing ML. In Sect. 4, the results are presented and discussed. Section 5 critically reviews the limitations of our approach and the results obtained. Furthermore, the implications for further research are derived. Finally, a summary is given in the last section.

## 2 State of the art

The products of machine and plant manufacturers typically consist of several hundred to several thousand components. These are procured from suppliers or manufactured in the company's production facilities. Purchased components can be procured on an order-anonymous basis, such as for standard components, and an order-specific basis, such as for special and drawing components. The procurement of components from suppliers as well as the manufacturing of components in the in-house production belong to processes upstream of the assembly [16]. Since the assembly is a convergence point where several material flows converge, the risk of delays due to missing components is increased [17].

One established model to analyze converging material flows is the assembly flow element developed by Schmidt [18] with further developments and applications in the assembly flow diagram and supply diagram [16, 18]. In all models, the so-called completer is the last inflow to an assembly order and is therefore the component that was supplied last by the processes upstream of the assembly. A completer can be completed on time—before the planned start date of the assembly, or late—after the planned start of the assembly. A late finalization of a completer, therefore, leads to a delay in the start of assembly. In this manuscript, we define such components as “assembly start delayers” (see also Chapter 1). Assuming that all components are necessary to start the assembly, the schedule variance of the assembly start delayer determines the earliest possible start date of the assembly. Accordingly, a temporal acceleration of the manufacturing and/or procurement process of an assembly start delayer has the biggest potential to push a delayed assembly start back to the target date. However, the supply diagram is primarily designed to analyze data relating to the past and to identify general issues such as an overall bad assembly supply situation in individual assembly areas. To derive case-specific countermeasures to accelerate individual production orders, further analysis is needed.

In production, typically scheduling techniques are used to derive order sequences and to calculate lead times of work orders used to determine the start dates and end dates of the respective orders and subsequently to determine the assembly start delayers [19]. The order sequence is defined according to certain rules considering for example the available production

capacities, the technical requirements, the demand dates, and the system status [8, 20, 21]. Further, especially for remanufacturing systems, also environmental objectives are considered [22, 23]. To optimize the lead time of an order, the determination of its waiting time depending on the machine's utilization is essential [24]. Here, performance curves considering functional relationships between logistic parameters such as lead times, throughput, and stock play a key role [24, 25]. Nevertheless, deviations from the schedule may occur leading to an inaccurate determination of the assembly start delayers. Besides determining the assembly start delayers based on calculated lead times utilizing scheduling techniques, it is also possible to predict lead time directly. By predicting the lead times, completion dates can be determined early and deviations from the schedule can be detected [26]. In the past, many approaches for the prediction of lead times have been established. For example, Cheng and Gupta [27] investigated methods from the field of operations research (OR) such as Constant (CON), Random (RAN), or Total-Work (TWK). With the increasing development of ML, new methods for predicting lead times have emerged (see, for example, [28–31]).

A systematic literature review conducted by Burggraef et al. [9] has analyzed existing approaches focusing on the prediction of lead times in the research fields of ML and OR and classified them according to the three criteria data class, data origin, and used method/algorithm. Looking at the data class, the authors identified that the majority of publications examined use order data and information about the system status of the production system. In detail, 95 % of their 42 publications examined use order data, and 62 % use information about the system status. Jia, Zhang et al. [32], Berlec and Govekar [33] or Gramdi [34] for example use order data such as start and end dates of orders or order-specific processing times for the prediction of lead times, whereas the authors in [28] and [35], for instance, use a combination of order data and information about the system status such as the machine utilization, processing times or the queue length. In contrast to the order data and information about the system, with 24 % of the 42 publications examined, machine data are slightly less used. For example, the authors in [36] include the machine ID and the authors in [37] include the so-called 'equipment data' containing information about machines and tools in their prediction models. Further, Burggraef et al. [9] identified Gyulai, Pfeiffer et al. [38] and Karagolan and Karademir [39] with a portion of only 5 % of the 42 publications examined as the only authors who include material data such as dimensions or specifications of the product in their prediction models. These findings highlight that material data were rarely used compared to order data, information about the system status, and machine data.

That being said, the business process of a machine and plant manufacturer typically hinges on tailor-made products for each customer need [13, 14]. Thus, the design phase in the business process and the associated documents herein can be

said to have a non-negligible influence on the desired product. Furthermore, the product design is also the basis for the production planning determining the process to manufacture the respective components [15]. Accordingly, the material data specified in the design phase are also influencing the manufacturing process. Consequently, we assume that the usage of material data in a model predicting assembly start delayers has an impact on its model quality. Nevertheless, material data are currently only rarely used in prediction models.

Utilizing the findings of the systemic literature review in [9], the authors in [10] applied different ML algorithms on a total of 24 different prediction models on four different levels of detail to identify the modeling approach with the highest model quality in predicting assembly start delayers. Their models on the coarsest level of detail predicted assembly start delayers utilizing a binary classification. Their models on the three finer levels of detail predicted assembly start delayers via a prediction of different lead times (component lead times, order lead times, and operation lead times) utilizing a regression approach and subsequent postprocessing operations to identify the assembly start delayers. After training the 24 prediction models based on a real data set of a machine and plant manufacturer and evaluating their model quality, they identified the coarsest level of detail utilizing the binary classification as the best modeling approach. Thus, one of their findings was, that performing a binary classification to predict assembly start delayers outperformed the prediction of assembly start delayers based on a prior prediction of lead times utilizing a regression model. Accordingly, for our approach, applying a binary classification is recommended to predict assembly start delayers. Furthermore, the authors in [10] already used material data in all of their 24 prediction models leading to good results. Nevertheless, as they did not systematically analyze the impact of material on the model quality, there is still no analysis available proving that material data have an impact on the quality of models predicting lead times.

In summary, there are models available for the prediction of lead times, but they are not explicitly used for the prediction of assembly start delayers. Currently, there is only one approach available focusing on the prediction of assembly start delayers in the field of machine and plant manufacturers comparing a direct prediction of assembly start delayers with an indirect prediction based on a previous lead-time prediction. But still, there is no analysis performed on the impact of material data on the quality of models predicting lead times.

Consequently, in this work, we will focus on investigating the influence of material data on the quality of models predicting assembly start delayers. This systemic analysis is completely novel compared to recent research. For this purpose, the following research question is posed, considering the previous explanations: "What effect does the use of material data have on the model quality of a model

predicting assembly start delayers?” Following our argumentation that the products of machine and plant manufacturers are typically designed tailor-made to meet the specific customer needs and that the material data, therefore, characterize a product, we formulate the following working hypothesis: “The model quality for the prediction of assembly start delayers increases when utilizing material data.”

### 3 Modelling approach

Examining an exemplary use case is an established approach in the field of machine learning, especially in lead-time prediction (see, for example, [37, 39–41] and assembly start delayer prediction (see, for example, [10]). One motivation for examining an exemplary use case is to gain insights for real needs, such as the need of a manufacturing company, rather than to develop theories without practical relevance [42]. Accordingly, investigating an exemplary use case to answer our research question and to study our working hypothesis is an appropriate and established approach and thus, was our approach of choice. Furthermore, as this work extends our previous research in the prediction of assembly start delayers [10], we investigated the same case at the previously chosen representative machine and plant manufacturer.

The methodology used in this manuscript is following the established Cross Industry Standard Process for Data Mining (CRISP-DM) [43, 44] consisting of the six phases Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment.

#### 3.1 Business understanding

In the Business Understanding phase, we derived objectives and requirements from a business perspective and converted them into a data mining problem. The objective from a business perspective was to prevent delays due to missing components in the final assembly so that the predefined due date of a customer order can be met. Early detection of components that have a higher tendency of late finishing in their preprocessing stages would be helpful to prevent a subsequent delay in the final assembly, as the production planer can accelerate the order in the preprocessing stages. The company under observation develops machines for steel production, which are made up of several hundred components. These components are both procured from suppliers and are manufactured in-house. An analysis carried out in the company beforehand showed that approx. 95 % of the assembly start delayers are components produced in the company’s production. Thus, the scope of our prediction model was constrained to the components manufactured in-house. In the process upstream of the assembly, these in-house components are processed by various machines for mechanical and welding operations.

In the prediction model, the components were classified as “assembly start delayer” (ASD) or “no-assembly start delayer” (NASD) which was identified as a suitable modeling approach in our previous research [10]. For this classification, a slightly modified version of the definition of the assembly start delayers given in chapter 2 is applied: Instead of considering only one single assembly start delayer as a date determining factor for the assembly start according to the definition of Beck and Schmidt [16, 18] and thus, assigning the highest potential for improvement to this component, several assembly start delayers were considered for each assembly order. This extension is recommended, since considering only one assembly start delayer is not revealing whether this single one is an outlier or whether a large portion of the components is completed at a similar time. The modified assembly start delayer classification was defined as follows: If the schedule variance of a component is larger than or equal to 80 % of the maximum schedule variance of all components of an assembly order, which is the schedule variance of the actual assembly start delayer, then this component is considered as an assembly start delayer. In detail, we utilized the formula

$$Class_i = \begin{cases} ASD & SV_{i,j} \geq 0.8 * SV_{j,max} \wedge SV_{i,j} > 0 \\ NASD & SV_{i,j} \geq 0.8 * SV_{j,max} \vee SV_{j,max} < 0 \end{cases} \quad (1)$$

to assign one of the two classes ASD or NASD to every component  $i$ , where  $SV_{i,j}$  is the schedule variance of component  $i$  of assembly order  $j$ , calculated by

$$SV_{i,j} = CD_{i,j} - TSD_j \quad (2)$$

where  $CD_{i,j}$  is the completion date of component  $i$  of assembly order  $j$  and  $TSD_j$  is the target start date of assembly order  $j$ , and  $SV_{j,max}$  the maximum schedule variance of all components of assembly order  $j$ , calculated by

$$SV_{j,max} = CD_{j,max} - TSD_j \quad (3)$$

where  $CD_{j,max}$  is the latest completion date of all components of assembly order  $j$  (the completion date of the respective completer).

The time of application of the prediction models (prediction time) and thus, the time of gaining knowledge about potential assembly start delayers should be as early as possible within the production process, as the production planer can accelerate the order in the manufacturing processes upstream of assembly stages. For the prediction models within this study, we set the date of order creation and thus, the completion of order planning as prediction time. At this point, all necessary information, such as bill of materials, operations, and machine assignments, are available.

Summarized, we converted the business objective to a binary classification problem. Subsequently, to answer the research question, and with our hypothesis that the model quality for the prediction of assembly start delayers increases when utilizing

material data, we derived our data mining approach: We compared ML-based binary classification models using a data set including material data with ML-based binary classification models using the same data set but excluding the material data (cf. Fig. 1). For both cases, “including material data” and “excluding material data,” we applied several ML-algorithms such as tree-based classifiers, support vector machines, or neural networks utilizing the Scikit-learn library or Keras library in Python (further details about the ML-algorithms used are explained in chapter 3 D). In total, 12 models were created, six per case utilizing different ML algorithms. Thus, with our approach we compared the performance of the different ML algorithms in both cases to identify the impact of material data on the model quality and the best performing ML algorithm by evaluating the achieved model qualities. Such a systemic analysis of the impact of material data on the model quality is completely new in recent research (see Chapter 2).

To evaluate the different achieved model qualities, we applied a confusion matrix, since the output of all ML models is the binary classification “assembly start delayer / no assembly start delayer.” The evaluation of the model quality with a confusion matrix is an established method and has

already been demonstrated in other studies (see, for example, [45, 46]). Based on the confusion matrix, we calculated Matthew’s correlation coefficient (MCC) and the F-score as established evaluation metrics to compare the performance of the different ML algorithms on both data sets. As recommended by the authors in [10, 47], the MCC considers the balanced ratios of all four confusion matrix categories and thus, is the most informative metric to evaluate a confusion matrix. Considering the MCC also ensured that our model was not just predicting the majority class in our data set, which is “no assembly start delayer.” Furthermore, as recommended by the authors in [10] we considered the F-score as an evaluation metric since it is focusing on the prediction of positives (assembly start delayers) only, which is the most important category in our case of interest. For the F-score, we used the  $F_2$ -score in detail considering the recall two times as important as precision. This weighting is based on the assumption that it seems more important to identify as many of the actual assembly start delayers as possible, in case of doubt even more than exist, and to define acceleration measures for them, than not to identify individual assembly start delayers at all. By evaluating each ML model with these metrics, the impact

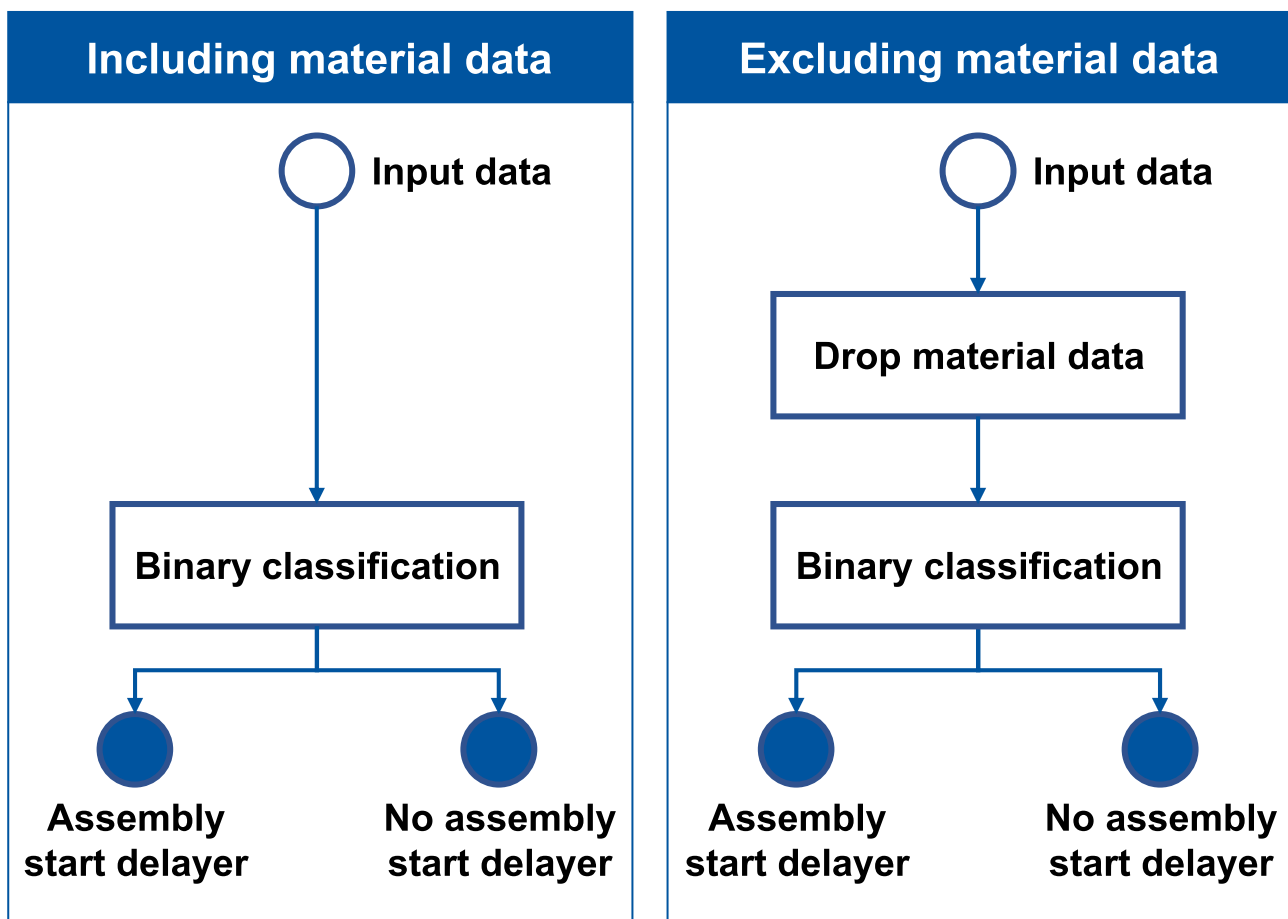


Fig. 1 Modeling architecture to quantify the impact of material data on the model quality

of material data on the quality of a model predicting assembly start delayers can be determined. Furthermore, with the MCC and  $F_2$ -score, we use the same metrics as in our previous research [10] and thus ensure comparability.

### 3.2 Data understanding

In the data understanding phase, according to the authors in [44], we collected and analyzed the data to identify data quality problems and to develop a solid understanding of the dataset. The data were collected from the Enterprise Resource Planning (ERP) and Advanced Planning and Scheduling System (APS) of the plant and machine manufacturer under observation with a period under review of one year. In detail, we collected data from the four data classes order data, machine data, material data, and system status, and thus follow the recommendation of the authors in [9]. The data export consisted of several separate CSV files containing assembly orders, the corresponding production orders and operation as well as information on the material and the systems status. To better join the different files, we set up an entity-relationship diagram (see Fig. 2) enabling us to identify the primary keys, which are the prerequisite for their connection.

The complete dataset consisted of 356 assembly orders comprising 1,506 components supplied by the in-house production and thus, is equal to our previous research [10]. These 1,506 in-house components are manufactured by a total of 3,187 production orders comprising 15,772 operations. With our modified definition of an assembly start delayer, we had a total of 24 % “assembly start delayers” and 76 % “non-assembly start delayers” of all in-house components.

Further, as recommended by the authors in [44] we focused on gaining a better understanding of the data and developing first ideas of relevant data fields for the prediction of assembly start delayers by performing an exploratory data analysis. In detail, we utilized several graphical techniques such as box-plots, scatter plots, or Pareto charts. For example, we analyzed the distribution of the total number of operations needed to manufacture ASDs and NASDs (cf. Fig. 3a) showing a slight deviation between both classes. Components manufactured in more operations have a slightly higher tendency of becoming an ASD. As another example, we plotted the distribution of the gross weight of ASDs and NASDs as an initial study of the impact of material data (cf. Fig. 3b). ASDs have a slightly higher mean and median gross weight than NASDs. Heavier components may need extra handling effort and transport time and therefore have a higher tendency of becoming an ASD.

### 3.3 Data preparation

With the gathered understanding of the data, we continued with preparing the final dataset for training the models by transforming and cleaning the initial raw data. In detail, we

continued to identify the relevant data field for the prediction models by performing a correlation analysis as recommended by the authors in [48]. Subsequently, after further data pre-processing operations such as discretization, decomposition, normalization, and aggregation (see, for details, [49, 50]), we defined the features for our data model resulting in 17 features, although not all features are applied in all models (see Table 1).

Since tree-based classifiers from the Scikit-learn library and neural networks from Keras library can only be trained on numerical variables in Python [51], the categorical variables such as “component name”, “dispatcher” and “priority” were converted to Boolean values by performing One-Hot-Encoding. The number of features increases to a total of 375 features. Due to the One-Hot-Encoding, our data set was transformed into a sparse matrix containing equal information but in a higher dimensional room. This sparse matrix could for example hinder the optimization of a neural network, due to a not neglectable number of zeros as input of the model. Furthermore, the encoded features could have a dependency on each other. To investigate the correlations between the features, we created a  $375 \times 375$  correlation matrix in form of a lower triangular leading to 71,631 individual correlation coefficients which were assigned to five bins of different correlation strengths (cf. Table 2) according to the established rules recommended by the authors in [52, 53]. Initially, 1.4 % of all feature-pairs showed at least a moderate correlation a correlation coefficient higher than 0.5 and 1.5% of features pairs have low correlation (Tables 1–5). This indicates an existing dependency between our features. Thus, a Principal Component Analysis (PCA) was performed to avoid a sparse matrix and to reduce the dependencies between the features to ensure a good model quality. The improvement of the model quality by using a PCA has already been demonstrated in other studies (see, for example, [54]). By performing PCA, the 379 features were transformed into 46 principal components, which explain most of the variance of the original features. After performing PCA, we again performed a correlation analysis and assigned all correlation coefficients to the equal five bins (cf. Table 2), showing that all principal component pairs have a negligible correlation.

For training and evaluating the models, the dataset was divided into training and test sets with a ratio of 80 % training data to 20 % test data. In selecting the ratio, we followed established ratios. These are approx. 75 % - 80 % training data to 25 % - 20 % test data [55].

### 3.4 Modeling

The subsequent modeling phase covered the development of ML models and the calibration of the hyperparameters to optimal values [44]. All ML models predict assembly start delayers using a binary classification, which was identified as the best modeling approach in our previous research [10]. Thus, components are classified as

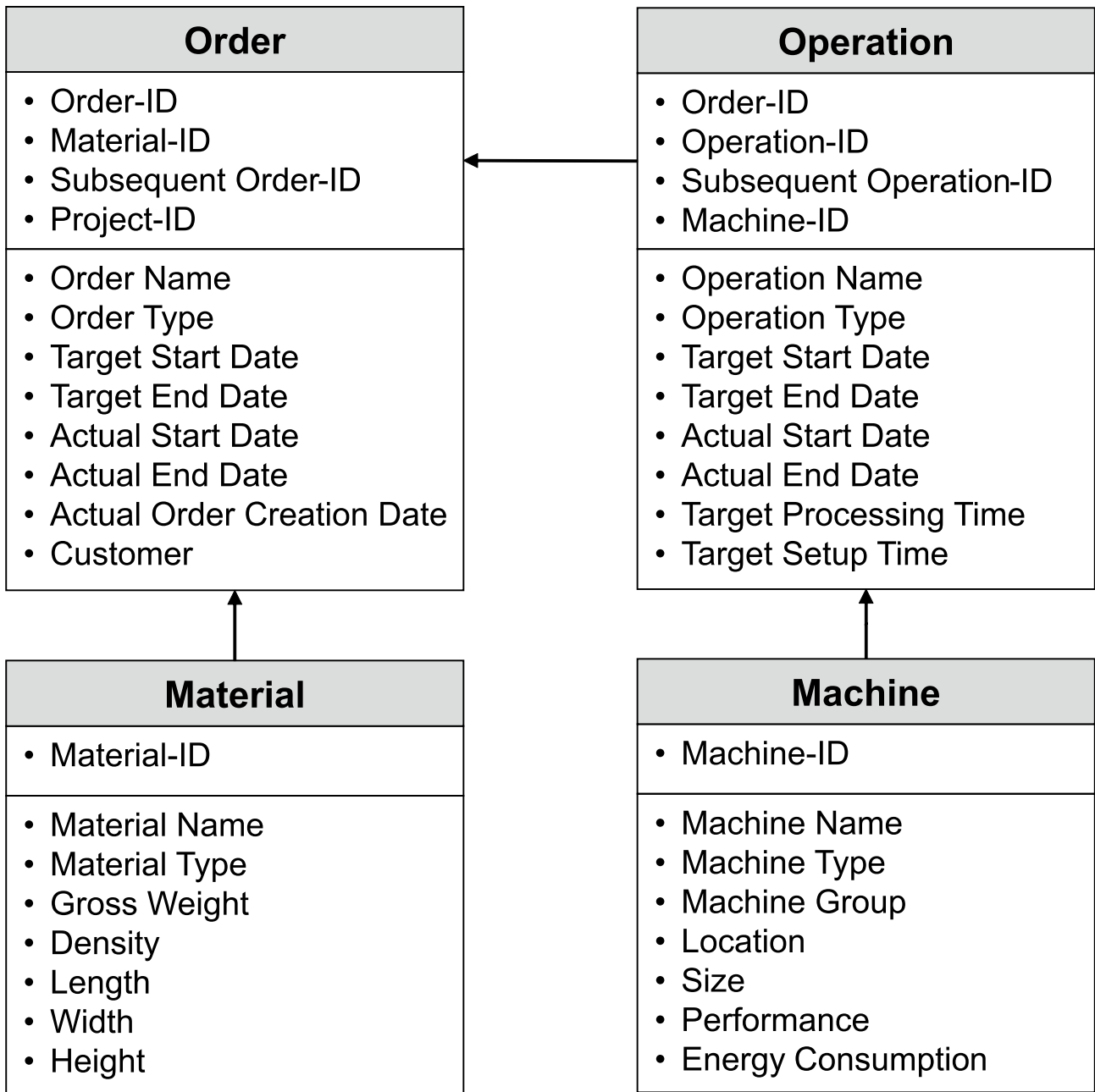
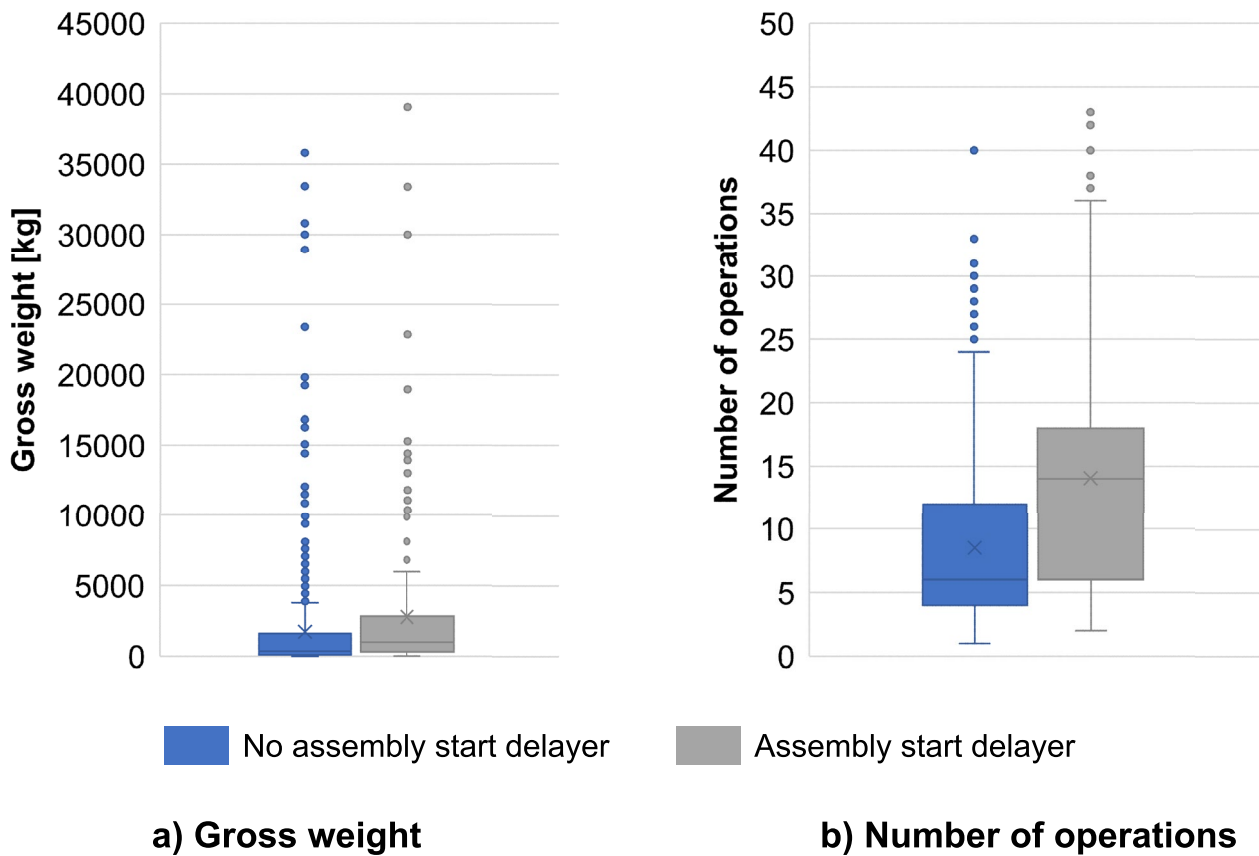


Fig. 2 Entity-relationship diagram with an excerpt of available features per data table

“assembly start delayer” or as “no assembly start delayer.” To ensure the comparability of all ML models, we chose the same set of ML algorithms on both data sets. In detail, we compared the performance of a Support Vector classifier (SVC), a Decision Tree (DT) classifier, a Random Forest (RF) classifier, an Adaptive Boosting (AdaBoost) classifier utilizing a DT-classifier as a base estimator, a Gradient Boosting (GB) classifier and a Multilayer Perceptron (MLP), since they are established approaches for binary classifications [56–58]. For the MLP, specifically, a

double hidden layer feedforward net with stochastic gradient descent (SGD) optimizer was applied. The number of nodes was 46 nodes on the input layer to cover all input features after performing One-Hot-Encoding and PCA, 50 nodes on each hidden layer, and one node on the output layer for the binary classification. The number of hidden layers, the number of nodes on the hidden layers, and the activation function on the hidden layers were defined by continuous optimization of the model quality. In detail, we compared different network architectures ranging from one



**Fig. 3** Excerpt of the exploratory data analysis: Impact of gross weight and number of operations on assembly start delayers

to ten hidden layers with 1 to 100 nodes per hidden layer. The best network structure was the above-mentioned double hidden layer net. As activation function for the output layer, a sigmoid function was chosen, which is particularly suitable for binary classifications [59]. For the hidden layers, we applied a ReLU function as activation function after comparing it with the sigmoid function, tanh function and He function regarding the reached model qualities. All classification models were implemented in Python 3.7 utilizing the Scikit-learn library and Keras library. An overview of the optimized hyperparameters used in each of the classification models is given in the appendix in Tables 4 and 5.

In summary, we created 12 different prediction models to classify components as ASD or NASD. These 12 models differentiated in the utilization of material data—including or excluding material data - and in the utilized ML algorithm—six algorithms per material data case. The target was to quantify the effect of utilizing material data on the quality of a model predicting assembly start delayers while comparing different ML algorithms, which is a novel approach compared to recent literature. As metrics to evaluate the model quality, we used the MCC and F-Score based on a confusion matrix.

## 4 Evaluation of model application

In the evaluation phase, the applied models were thoroughly evaluated to check whether they meet the targets of our data mining approach [44]: Quantifying the impact of material data on the quality of a model predicting assembly start delayers. Thus, we split the two data sets—including and excluding material data—into two separate train and test data sets. Subsequently, we trained and tuned all ML algorithms based on the train data sets and then evaluated the achieved model qualities based on the two test data sets. The results are documented in Table 3.

Upon evaluating the metrics, it is particularly noticeable that the models trained on the data set including material data achieved the best results. Furthermore, the best results per data set were both achieved by the GB classifier. With an MCC of 0.67 and an  $F_2$ -score of 77 %, the GB classifier utilizing material data outperformed the GB classifier not utilizing material data with an MCC of 0.62 and an  $F_2$ -score of 71 %. Thus, comparing the best ML model per data set already indicates a dependence of the model quality on the material data.

Additionally, we created boxplots showing the spread in the  $F_2$ -score and MCC of all ML models utilizing the two



**Table 1** Features used in the prediction model

Data class	Feature	Including material data	Excluding material data
Order data	Target lead time	X	X
	Total number of orders	X	X
	Total number of operations	X	X
	Target processing time	X	X
	Target setup time	X	X
	Order creation-delay	X	X
	Priority	X	X
	Operation type	X	X
	Dispatcher	X	X
	Number of production areas a component/order passes through	X	X
System status	Number of orders in system	X	X
Material data	Gross weight	X	
	Component name	X	
Machine data	Production area	X	X
	Workstation type	X	X
	Workstation number	X	X
	Workstation capacity	X	X

different data sets (cf. Fig. 4). With the boxplots, the overall dependency of the model quality on the material data independent of the considered ML algorithm was visualized. The distribution of the  $F_2$ -score and MCCs of the ML models trained on the dataset including material data differed from the respective distribution of the ML models trained on the dataset excluding material data. This indicated that, overall, the ML models trained on the dataset including material data performed better than those excluding material data. Thus, the comparison of the overall spread of the ML models emphasizes the indication that material data have an impact on the quality of models predicting assembly start delayers.

Finally, we performed a statistical test to validate our working hypothesis. In detail, we performed two paired-samples t-tests, also referred to as dependent sample t-tests, both for MCC and  $F_2$ -score. This paired-samples t-test is used to assess whether the population means of two related samples differ. Thus, with the two paired-samples t-tests, we compared the

means of the two samples 'ML models including material data' and 'ML-models excluding material data' individual for MCC and  $F_2$ -score. Additionally, we considered that the applied ML algorithms in each of the two samples were equal. Applying both tests revealed a p-value for MCC of approx. 0.003 and for the  $F_2$ -score of approx. 0.005. Consequently, since both p-values were less than 0.05, the difference between the two samples in both the MCC and  $F_2$ -score was statically significant. Accordingly, the impact of our considered material data on the model quality was statistically significant as well.

Consequently, the working hypothesis could be confirmed. The model quality significantly increased when material data were considered. However, in our case, there was only a slight increase in the MCC with an average of 0.04 and the F-score with an average of 3 %. Thus, we further analyzed possible explanations for this small impact only and hypothesized prospects to further increase the benefit of utilizing material data. The reason for the small impact of material data observed could be that the considered material data—gross weight and component name—contain too little information to describe the characteristics of the components. Other information of the component such as dimensions, volume or number, and specification of features in the component's CAD model like drill holes, shaft shoulders, radii, or surface roughness could further increase the impact of material data. For example, the transportation, stocking, and handling effort of a component do not solely depend on its weight, but also other characteristics like dimensions and volume. For instance, the dimensions of a component determine whether the component can be easily transported by a forklift or crane, and thus, indicates an impact on an increase in transport times. Furthermore, the number and specification of a component's features indicate its complexity and need for special processing operation influencing the processing time. Thus, considering additional material data could increase the model quality.

In summary, we could answer our research question with our main contribution that the model quality of an ML-based model predicting assembly start delayers is significantly increasing when using material data. Thus, our study proved that models predicting assembly start delayers benefit from utilizing material data. In our exemplary case, we included the material data gross weight and component name in our prediction model significantly increasing the model quality. With these results, our approach is the first to systematically analyze the influence

**Table 2** Correlation between features before and after PCA following the bin sizes of [52, 53]

Bin	Correlation Coefficient	Before PCA	After PCA
Very high correlation	1.0 to 0.9 (−1.0 to −0.9)	0.3%	0.0%
High correlation	0.9 to 0.7 (−0.9 to −0.7)	0.6%	0.0 %
Moderate correlation	0.7 to 0.5 (−0.7 to −0.5)	1.5%	0.0%
Low correlation	0.5 to 0.3 (−0.5 to −0.3)	6.5%	0.0%
Negligible correlation	0.3 to 0.0 (−0.3 to −0.0)	91.1%	100.0%

**Table 3** Reached model qualities of all prediction models

Model	Including material data		Excluding material data		Deviation	
	MCC	F <sub>2</sub> -score	MCC	F <sub>2</sub> -score	MCC	F <sub>2</sub> -score
SVM	0.55	72%	0.47	67%	0.08	5%
DT	0.56	68%	0.52	66%	0.04	2%
RF	0.52	70%	0.50	68%	0.02	2%
GB	0.67	77%	0.62	71%	0.05	6%
AB	0.60	72%	0.56	68%	0.04	3%
MLP	0.59	73%	0.56	71%	0.02	2%

of material data on model quality in predicting assembly start delayers.

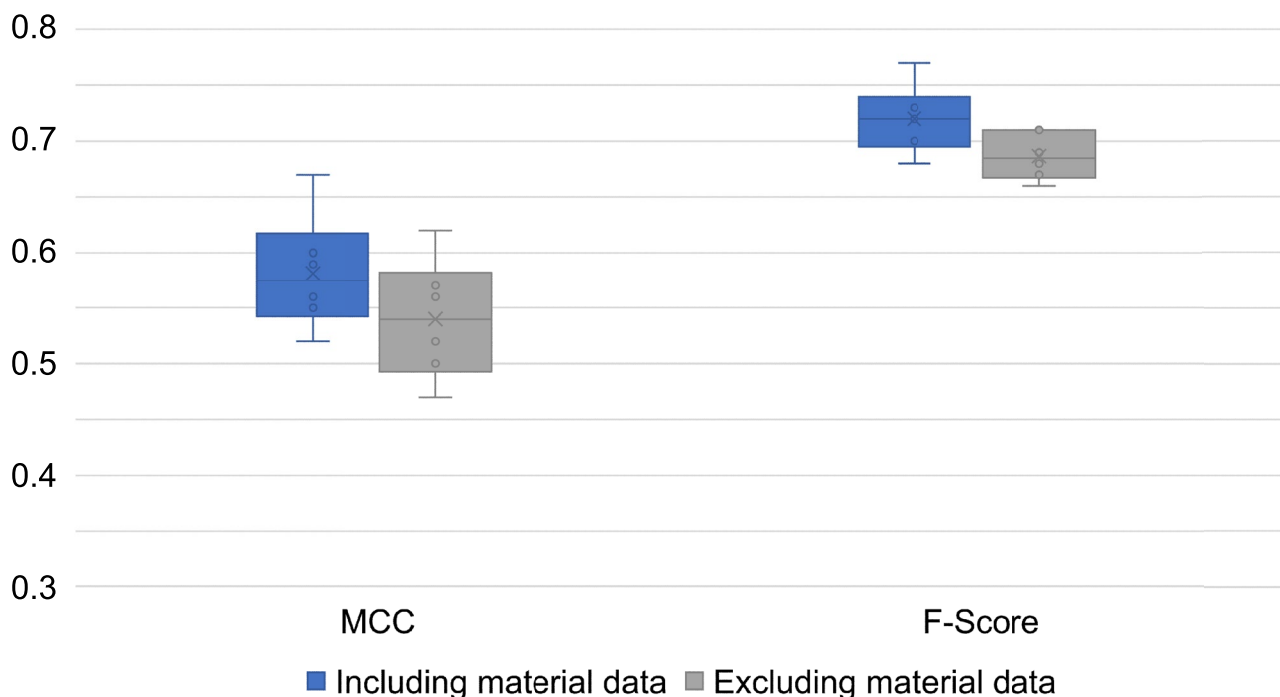
## 5 Limitations and implications for further research

In this work, we only considered one single machine and plant manufacturer as an exemplary case. Although a case-based approach is common in the field of machine learning (see, for example, [10, 37, 39–41]), the findings might remain case-specific and might not be generalizable. Accordingly, future research should validate the achieved findings considering additional machine and plant manufacturers in further case studies.

Nevertheless, in our work, we were able to show that material data have a positive influence on model quality for

predicting assembly start delayers. However, the verifiable influence of the material data on the model quality was only small. We suspected the small range of data fields from the material data as a possible reason for this. Further material data could improve the model quality and thus strengthen the influence of the material data. Accordingly, future research should set up a model to predict assembly start delayers with additional material data.

The addition of further material data could also improve the generally low model quality. With a maximum MCC of 0.67 and a maximum F<sub>2</sub>-score of 77%, the model quality is still too low for a successful practical application of the model, as there are still many false positive and false negative predictions. In general, the model quality depends on the input data, the utilized ML algorithm, and the complexity of the modeling approach [11, 60–62]. Together with our previous study [10], we already analyzed several different ML algorithms



**Fig. 4** Boxplot of MCC and F-Score for all prediction models on each of the four levels of details

and modeling approaches. Thus, we infer that neither further optimization of the ML algorithm nor the modeling approach used is likely to lead to a significant improvement of the model quality. Instead, we infer that an enhancement of the input data could further improve the overall model quality, as the database also has an essential influence on the model quality [11, 12]. In our study, we already proved that material data influence the model quality. Consequently, we encourage further studies to consider additional data fields from the area of material data when setting up a model predicting assembly start delayers to further optimize the model.

Together with our previous work in the same research field [10], our findings observed are a good starting point in the prediction of assembly start delayers and the influence of material data on the model quality. As we could easily access the considered material data and integrate it into our data set, we added value to our model without much additional effort for data acquisition. Consequently, we have shown that it is worth also considering data, which might not have any influence on the model quality at first glance, and consequently is not commonly used. For future research in the field of applied machine learning, the elaboration of the database should be extended to other easily accessible data sources, even if they are not typically considered for the respective use case.

## 6 Conclusion

At machine and plant manufacturers, the manufacturing process typically begins with the design of the product and its components before planning and executing the production process to manufacture the individual components needed in the subsequent assembly process. An essential factor for meeting a delivery date is the start of the assembly on time and a prior timely supply of the components needed for assembly. Subsequently, components produced in the processes upstream of the assembly have a direct influence on the performance of the assembly process. To meet delivery dates, we set up a supervised learning model to predict potentially delayed individual components (we call them assembly start delayers) in the early stages of the manufacturing process. Currently, machine learning models in the related area of lead time prediction typically include information about the system status, the machines, and the orders in their prediction model and do not consider material data [9]. As the design of a product is a central process for machine and plant manufacturers and the components are typically tailor-made to meet the customer's needs, we assumed that material data influence the model quality. Thus, we formulated the following working hypothesis: "The model quality for the prediction of assembly start delayers increases when utilizing material data." To verify

the working hypothesis, we applied the established CRISP-DM procedure at an exemplary chosen machine and plant manufacturer. Here, we created 12 different prediction models to classify components as "assembly start delayer" or "no assembly start delayer." These 12 models differentiated in the utilization of material data—including or excluding material data—and in the utilized ML algorithm—six algorithms per material data case. The target was to quantify the effect of utilizing material data on the quality of a model predicting assembly start delayers while comparing different ML algorithms. As metrics to evaluate the model quality, we used the MCC and F-Score based on a confusion matrix.

Evaluating the different quality metrics of the 12 prediction models revealed a positive impact of the material data on the model quality. Thus, the working hypothesis could be confirmed. However, in our case, there was only a slight increase in the MCC and F-score. As a possible explanation for the small impact on the model quality, we suspect the limited information about the material considered in our model—gross weight and component's name only. Adding further information about the material such as dimensions, volume, or number and specification of features in the component's CAD model like drill holes, shaft shoulders, radii or surface roughness could further increase the impact of material data. Nevertheless, even with our limited consideration of material data. We verified, that utilizing data, which is commonly not used in prediction models increases the model quality.

In total, we successfully analyzed the impact of material data on the quality of models predicting assembly start delayers and gave insights into the performance of different modeling approaches. With our results, we achieved our two main contributions: First, we developed a model to predict assembly start delayers utilizing a machine learning classification approach. Second, we identified that material data influence the model quality of a model predicting assembly start delayers. However, there was only a slight influence. With our findings, for future machine learning approaches in the area of production planning and control, we recommend considering data sources apart from typically used data sources as well. We were able to show that even atypical data sources can contribute to an improvement of the model.

## Appendix

The hyperparameters used in the prediction models were optimized utilizing a grid search and cross-validation algorithms (GridSearchCV) from scikit learn. Tables 4 and 5 summarize the utilized hyperparameters in the different models on the four levels of detail.

**Table 4** Hyperparameters of the prediction models including material data

Model	Hyperparameters					
	c	penalty	loss	dual		
SVM	1	l2	hinge	1000	true	
	min samples split	min samples leaf	max features	max depth	learning rate	n estimators
DT	2	1	None	None	-	-
RF	2	1	70	8	-	800
GB	15	6	80	7	0.2	485
AB	30	15	80	1	0.2	1600
MLP	momentum	dropout rate	batch size	epochs	learning rate	
	0.9	0.2	64	450	0.001	

**Table 5** Hyperparameters of the prediction models excluding material data

Model	Hyperparameters					
	c	penalty	loss	dual		
SVM	100	L2	hinge	1000	true	
	min samples split	min samples leaf	max features	max depth	learning rate	n estimators
DT	2	3	None	None	-	-
RF	2	1	80	9	-	800
GB	16	6	80	7	0.2	694
AB	30	19	55	5	0.2	1600
MLP	momentum	dropout rate	batch size	epochs	learning rate	
	0.9	0.2	64	2000	0.001	

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Data availability** Not applicable.

**Code availability** Not applicable.

## Declarations

**Conflicts of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Burggräf P, Wagner J, Lück K, Adlon T (2017) Cost-benefit analysis for disruption prevention in low-volume assembly. *Production Engineering Research and Development* 11(3):331–342. <https://doi.org/10.1007/s11740-017-0735-6>
- Peng DX, Lu G (2017) Exploring the impact of delivery performance on customer transaction volume and unit price: Evidence from an assembly manufacturing supply chain. *Prod Oper Manag* 26(5):880–902. <https://doi.org/10.1111/poms.12682>
- Joseph AT (1999) Formulation of manufacturing strategy. *Int J Adv Manuf Technol* 15(7):522–535. <https://doi.org/10.1007/s001700050098>
- Coronado AE, Lyons AC, Kehoe DF, Coleman J (2004) Enabling mass customization: extending build-to-order concepts to supply chains. *Prod Plan Control* 15(4):398–411. <https://doi.org/10.1080/0953728042000238809>
- Fredriksson P, Gadde LE (2005) Flexibility and rigidity in customization and build-to-order production. *Ind Mark Manag* 34(7):695–705. <https://doi.org/10.1016/j.indmarman.2005.05.010>
- Nof SY, Wilhelm WE, Warnecke HJ (1997) *Industrial assembly*. Springer US, Boston, MA and s.l. <https://doi.org/10.1007/978-1-4615-6393-8>
- Whitney DE (2004) *Mechanical assemblies: their design, manufacture, and role in product development*. Oxford series on advanced manufacturing. Oxford Univ. Press, New York
- Lödging H (2013) *Handbook of manufacturing control: Fundamentals, description, configuration*. Springer Heidelberg. <https://doi.org/10.1007/978-3-642-24458-2>
- Burggräf P, Wagner J, Koke B, Steinberg F (2020) Approaches for the prediction of lead times in an engineer to order environment - a systematic review. *IEEE Access* 8:142434–142445. <https://doi.org/10.1109/ACCESS.2020.3010050>
- Burggräf P, Wagner J, Heinbach B, Steinberg F (2021) Machine learning-based prediction of missing components for assembly – a case study at an engineer-to-order manufacturer. *IEEE Access*

11. Zheng A, Casari A (2018) Feature engineering for machine learning: Principles and techniques for data scientists, 1st edn. O'Reilly, Beijing, Boston
12. Kilkenny MF, Robinson KM (2018) Data quality: Garbage in - garbage out. *Health Information Management: Journal of the Health Information Management Association of Australia* 47(3):103–105. <https://doi.org/10.1177/1833358318774357>
13. Schuh G, Stich V (eds) (2012) *Produktionsplanung und -steuerung*, 4th edn. VDI-Buch Springer Vieweg, Berlin and Heidelberg
14. Gosling J, Naim MM (2009) Engineer-to-order supply chain management: a literature review and research agenda. *Int J Prod Econ* 122(2):741–754. <https://doi.org/10.1016/j.ijpe.2009.07.002>
15. Slack N, Brandon-Jones A, Johnston R (2016) *Operations management*, 8th edn. Pearson, Harlow, England and London and New York
16. Beck S (2013) *Modellgestütztes Logistikcontrolling konvergierender Materialflüsse: Zugl.: Hannover, Univ., Diss. 2013, Berichte aus dem IFA, vol 2013,3. PZH-Verl., Garbsen*
17. Reinhart G, Cuiper R, Loferer M (1998) In: Reinhart G (ed) *Montage-Management: Lösungen zum Montieren am Standort Deutschland, TCW-Report. TCW, München, pp 7–11*
18. Schmidt M (2011) *Modellierung logistischer Prozesse der Montage: Zugl.: Hannover, Univ., Diss., 2010, Berichte aus dem IFA, vol 2011,1. PZH Produktionstechn. Zentrum, Garbsen*
19. Sun Y, Zhang C, Gao L, Wang X (2011) Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects. *Int J Adv Manuf Technol* 55(5–8):723–739. <https://doi.org/10.1007/s00170-010-3094-4>
20. Baker KR, Trietsch D (2009) *Principles of sequencing and scheduling*. John Wiley, Hoboken, N.J. <https://doi.org/10.1002/9781119262602>
21. Vieira Guilherme E, Herrmann Jeffrey W, Lin Edward (2003) Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *J Sched* 6(1):39–62. <https://doi.org/10.1023/A:1022235519958>
22. Wang W, Tian G, Yuan G, Pham DT (2021) Energy-time tradeoffs for remanufacturing system scheduling using an invasive weed optimization algorithm. *J Int Manag* 137(3):1602. <https://doi.org/10.1007/s10845-021-01837-5>
23. Tian G, Zhang H, Feng Y, Jia H, Zhang C, Jiang Z, Li Z, Li P (2017) Operation patterns analysis of automotive components remanufacturing industry development in China. *J Clean Prod* 164(3):1363–1375. <https://doi.org/10.1016/j.jclepro.2017.07.028>
24. Schömig A, Eichhorn D, Obermaier G (2007) In: Waldmann KH, Stocker UM (eds) *Operations Research Proceedings 2006: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Jointly Organized with the Austrian Society of Operations Research (ÖGOR) and the Swiss Society of Operation, Operations Research Proceedings. Springer-Verlag, s.l, pp 467–472. https://doi.org/10.1007/978-3-540-69995-8*
25. Engelhardt C (2000) *Betriebskennlinien: Produktivität steigern in der Fertigung. Hanser, München*
26. Raaymakers W, Weijters A (2003) Makespan estimation in batch process industries: a comparison between regression analysis and neural networks. *Eur J Oper Res* 145(1):14–30. [https://doi.org/10.1016/S0377-2217\(02\)00173-X](https://doi.org/10.1016/S0377-2217(02)00173-X)
27. Cheng T, Gupta MC (1989) Survey of scheduling research involving due date determination decisions. *Eur J Oper Res* 38(2):156–166. [https://doi.org/10.1016/0377-2217\(89\)90100-8](https://doi.org/10.1016/0377-2217(89)90100-8)
28. Öztürk A, Kayaligil S, Özdemirel NE (2006) Manufacturing lead time estimation using data mining. *Eur J Oper Res* 173(2):683–700. <https://doi.org/10.1016/j.ejor.2005.03.015>
29. Alenezi A, Moses SA, Trafalis TB (2008) Real-time prediction of order flowtimes using support vector regression. *Comput Oper Res* 35(11):3489–3503. <https://doi.org/10.1016/j.cor.2007.01.026>
30. Wang C, Jiang P (2019) Deep neural networks based order completion time prediction by using real-time job shop RFID data. *J Intell Manuf* 30(3):1303–1318. <https://doi.org/10.1007/s10845-017-1325-3>
31. Sha DY, Liu CH, Gao L, Wang X (2005) Using data mining for due date assignment in a dynamic job shop environment. *Int J Adv Manuf Technol* 25(11):1164–1174. <https://doi.org/10.1007/s00170-003-1937-y>
32. Jain V, Raj T (2018) An adaptive neuro-fuzzy inference system for makespan estimation of flexible manufacturing system assembly shop: a case study. *Int J Syst Assur Eng Manag* 9(6):1302–1314. <https://doi.org/10.1007/s13198-018-0729-6>
33. Berlec T, Govekar E, Grum J, Potočnik P, Starbek M (2008) Predicting order lead times. *J Mech Eng Sci* 54(5):308–321
34. Gramdi J (2009) In 2009 International Conference on Computers & Industrial Engineering. IEEE, pp 674–677. <https://doi.org/10.1109/ICCIE.2009.5223807>
35. Ioannou G, Dimitriou S (2012) Lead time estimation in mrp/erp for make-to-order manufacturing systems. *Int J Prod Econ* 139(2):551–563. <https://doi.org/10.1016/j.ijpe.2012.05.029>
36. Weng W, Fujimura S (2016) In 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI). IEEE, pp 975–979. <https://doi.org/10.1109/IIAI-AAI.2016.111>
37. Lingitz L, Gallina V, Ansari F, Gyulai D, Pfeiffer A, Sihn W, Monostori L (2018) Lead time prediction using machine learning algorithms: a case study by a semiconductor manufacturer. *Procedia CIRP* 72:1051–1056. <https://doi.org/10.1016/j.procir.2018.03.148>
38. Gyulai D, Pfeiffer A, Nick G, Gallina V, Sihn W, Monostori L (2018) Lead time prediction in a flow-shop environment with analytical and machine learning approaches. *IFAC-PapersOnLine* 51(11):1029–1034. <https://doi.org/10.1016/j.ifacol.2018.08.472>
39. Karaoglan AD, Karademir O (2017) Flow time and product cost estimation by using an artificial neural network (ANN): A case study for transformer orders. *Eng Econ* 62(3):272–292. <https://doi.org/10.1080/0013791X.2016.1185808>
40. Tirkel I (2013) Forecasting flow time in semiconductor manufacturing using knowledge discovery in databases. *Int J Prod Res* 51(18):5536–5548. <https://doi.org/10.1080/00207543.2013.787168>
41. Singh S, Soni U (2019) In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, pp 422–425. <https://doi.org/10.1109/CONFLUENCE.2019.8776892>
42. McCutcheon DM, Meredith JR (1993) Conducting case study research in operations management. *J Oper Manag* 11(3):239–256. [https://doi.org/10.1016/0272-6963\(93\)90002-7](https://doi.org/10.1016/0272-6963(93)90002-7)
43. Shearer C (2000) The CRISP-DM model: the new blueprint for data mining. *Journal of Data Warehousing* 5(4):13–22
44. Wirth R, Hipp J (2000) Crisp-DM: Towards a standard process model for data mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*. (1)
45. Davis J, Goadrich M (2006) In: Cohen W, Moore A (eds) *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*. ACM Press, New York, New York, USA, pp 233–240. <https://doi.org/10.1145/1143844.1143874>
46. Powers DMW (2020) Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *International Journal of Machine Learning Technology* 2:1 (2011). pp 37–6. [arXiv-ID 2010.16061](https://arxiv.org/abs/2010.16061)
47. Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 21(1):6. <https://doi.org/10.1186/s12864-019-6413-7>
48. Kuhn M, Johnson K (2016) *Applied predictive modeling*, corrected, 5th printing. Springer, New York
49. Han J, Kamber M, Pei J (2012) *Data mining: Concepts and techniques*, 3rd edn. The Morgan Kaufmann series in data management systems (Elsevier/Morgan Kaufmann, Amsterdam)

50. Dong G, Liu H (2018) Feature engineering for machine learning and data analytics. Chapman and Hall/CRC Data Mining and Knowledge Discovery Ser. Chapman and Hall/CRC, Milton
51. Verdhan V (2020) Supervised learning with Python: concepts and practical implementation using Python, 1st edn. Apress and Imprint: Apress, Berkeley CA. <https://doi.org/10.1007/978-1-4842-6156-9>
52. Hinkle D, Jurs SG, Wiersma W (2003) Applied statistics for behavioural sciences, 5th edn. Houghton Mifflin Company, Boston
53. Mukaka MM (2012) A guide to appropriate use of correlation coefficient in medical research. Malawi Medical Journal: The Journal of Medical Association of Malawi 24(3):69–71
54. Mohamad-Saleh J, Hoyle BS (2008) Improved neural network performance using principal component analysis on matlab. International Journal of the Computer, the Internet and Management 16(2):1–8
55. Basheer I, Hajmeer M (2000) Artificial neural networks: fundamentals, computing, design, and application. J Microbiol Methods 43(1):3–31. [https://doi.org/10.1016/S0167-7012\(00\)00201-3](https://doi.org/10.1016/S0167-7012(00)00201-3)
56. Jeatrakul P, Wong KW In: 2009 Eighth International Symposium on Natural Language Processing. IEEE, 20.10.2009–22.10.2009, pp 111–115. <https://doi.org/10.1109/SNLP.2009.5340935>
57. Dreiseitl S, Ohno-Machado L (2002) Logistic regression and artificial neural network classification models: a methodology review. Journal of Biomedical Informatics 35(5-6):352–359. [https://doi.org/10.1016/s1532-0464\(03\)00034-0](https://doi.org/10.1016/s1532-0464(03)00034-0). [http://dx.doi.org/10.1016/S1532-0464\(03\)00034-0](http://dx.doi.org/10.1016/S1532-0464(03)00034-0)
58. Kumari R, Srivastava SK (2017) Machine learning: a review on binary classification. Int J Comput Appl 160(7)
59. Ghatak A (2019) Deep learning with R. Springer, Singapore
60. Sessions V, Valtorta M (2006) The effects of data quality on machine learning algorithms. ICIQ 6:485–498
61. Jordan MI, Mitchell TM (2015) Machine learning: Trends, perspectives, and prospects. Science 349(6245):255–260. <https://doi.org/10.1126/science.aaa8415>
62. Robinson S (2008) Conceptual modelling for simulation part I: definition and requirements. J Oper Res Soc 59(3):278–290. <https://doi.org/10.1057/palgrave.jors.2602368>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.