Technical Report: Interpretable Machine Learning for Quality Engineering in Manufacturing- Importance measures that reveal insights on errors

Holger Ziekow, Ulf Schreier, Alexander Gerling, Alaa Saleh

{holger.ziekow,ulf.schreier,alexander.gerling,alaa.saleh}@hs-furtwangen.de

Abstract

This paper addresses the use of machine learning and techniques of interpretable machine learning to improve quality in manufacturing processes. It proposes analysis methods on top of SHAP values to elicit useful insights from machine learning models. These methods constitute novel importance measures that support quality engineers in the analysis of production errors. We illustrate and test the proposed methods on synthetic as well as on real-world data from a German manufacturer.

1 Introduction

Recent advancements in machine learning have created increased interest in leveraging the potential of this technology in a range of application domains. Within this paper we address the application of machine learning to quality engineering in manufacturing. Here, the goal is to leverage machine learning for identifying and reducing causes of production errors. Like in other applications, the black-box nature of many machine learning approaches poses challenges to the applicability. This creates the need for methods to explain machine learning models. Specifically, there are two prevalent reasons for explaining machine learning models in quality management for manufacturing. One is to gain trust in the model decisions. The other is to leverage insights of the model to drive human data analysis. That is, quality engineers want to be pointed to factors or combination of factors that help to understand and reduce production errors. In this paper we explore existing and new methods of explainable machine learning to address this need.

Recently, a range of methods have been proposed that are intended to make black-box machine learning models explainable (see e.g. Gilpin et al. [1] or Molnar [2] for an overview). Among the existing methods are several means to assess feature importance, i.e. means to quantify how important a feature is for the decision making of a given model. Feature importance can be used as guidance for humans about where they should focus their analysis of the data. However, the question is, what importance measures provide useful guidance. In this paper we argue that existing importance measures are not ideal for the application in manufacturing quality management. We also propose a range of new importance measures and test them on synthetic data as well as on real-world data from a German manufacturer.

The remainder of the paper is structured as follows. Section 2 presents the use case that motivates our work. Section 3 introduces our concept for identifying important features and defines corresponding measures. In Section 4 we evaluate the proposed importance measure on real-world data form a German manufacturer (SICK AG). In Section 5 we review related work. Section 6 summarize the content of this paper

Revised at 2019-11-29

2 Motivating use case

The methods proposed in this paper may be of general applicability, but are mainly motivated by the use case of quality engineering in manufacturing. Specifically, we draw the requirements from the PREFERML research project [3] and the participating manufacturing company SICK AG. The task of quality engineers is ensuring high product quality while keeping the production cost low. Therefore, they strive to identify and eliminate root causes of production errors. All products undergo rigid tests not only when they are finished. The final checks ensure that only high-quality products are shipped to the costumers. However, it is desirable for the manufacturer to sort out faulty products early in the production process and to avoid allocation of resources to later discarded products. Therefore, additional quality checks are conducted after each production step throughout the production line. Each checkpoint records a range of measurements and sorts out products that do not satisfy predefined criteria. The recorded data points are also the source for investigating error causes. For instance, a quality engineer might realize that products that show a value greater than X for property Y in production step A, have an increased chance of failing when checked in production step E. This insight might result in adjustments of check in step A to filter out such products early.

However, the data pool for investigation is very large. The manufacturer in the PEFERML research project takes several thousands of measurements for each product. For humans – who arguably struggle to comprehend more than 3 dimensions at a time – it is a very challenging task to find relevant relations in such a large pool of data. It is therefore desirable to have support from an artificial intelligence or machine learning system. The methods that we propose in this paper provide means to leverage machine learning for guiding the work of quality engineers.

3 Finding Important Features for Quality Engineers with SHAP

The goal is to point quality engineers to interesting measurements (features) in the quality test data. We consider a feature interesting in the target context, if it provides actionable insights for quality engineers to adapt thresholds in the current quality checks. Hence a feature is interesting if it (a) enables a simple rule for predicting errors and (b) the prediction with that rule is of sufficient quality to reduce production cost.

3.1 Shapley Additive Explanations

In our work we leverage local explanations and specifically SHAP values [4] to reason about feature importance. Local explanations address feature importance for individual data points. This is in contrast to global importance measures that capture the general (e.g. average) importance of a feature. Local explanations have the advantage that they can reveal if features are sometimes (possibly only a few times) of great importance. The contribution of such features may be hidden in global importance measures that aim to quantify an average importance. However, a feature that is very important a few times can be of major interest for quality management. This is because the interesting situations (predictable errors) are rare in a production process that has been optimized for years. Hence, a feature may be most of the time not important (e.g. because the value is in a good range) and very predictive in few cases (e.g. if the value is outside a good range). This feature is only helpful in very few but interesting cases. Also, this feature may yield a very helpful insight, e.g. that the values must be in a specific range (where they usually but not always are). With our methods we aim to identify such features.

The proposed feature importance metrics are not global or local in the classical sense. The identified features are important in the sense that they allow for good and simple predictions for

some regions. Another way to think about this is that we are looking for surrogate models in regions were the model works very well (and hence the phenomenon of interest is well predictable).

3.2 Illustrative Example for using SHAP values for Quality Management

SHAP values provide insights into predictions for individual data points. For a given data set this results in a set of explanations. Analyzing this set of explanations can yield additional insights into the workings of a model and the modelled phenomenon. Lundberg et al. [5] proposed and implemented some analysis on top of SHAP values and SHAP interaction values. These include clustering of explanations and visualization of interactions.

This chapter introduces additional methods for analyzing SHAP values and SHAP interaction values. In contrast to existing methods for analysis, the proposed methods do not aim at providing a holistic understanding of the analyzed model or phenomenon. Instead they aim at finding simple and good explanations for part of the analyzed phenomenon. This is motivated by the use case of quality engineering in manufacturing. It is not likely that all production errors can be well predicted based on quality logs. However, some errors may have a clear root cause that is reflected in the data. The proposed analysis methods should help quality engineers to find simple and high-quality rules to reduce certain errors. In other words, the aim is to elicit from the learned model if there are cases where simple and high-quality predictions can be made.

The envisioned workflow is that quality engineers use the analysis methods to identify features and relations that may be of interest for them. In a second step they use the results of the analysis as guidance for visually inspecting the quality data. Based on the visualization and their domain knowledge they can then decide on actions for improving production quality.

We argue that a feature or relation between features is of potential interest, if it enables (a) prediction rules with high prediction quality and (b) the respective rules are simple enough for human comprehension as well as for taking corresponding action (e.g. adjusting a threshold in a quality check). In the motivating use case, prediction quality is high enough, if the ratio between true positives TP and false positives FP is high enough. That is, TP/FP must be bigger than the cost of a false positive divided by the savings for a true positive. (We do not care about false and true negatives, because they do not result in costs or savings compared to the as-is process.)

We further argue that the support for the rule is less important. Errors are rare in highly optimized production lines and any prediction rule will likely have low support. It is more important to identify strong effects, because they enable clear actions. A weak but more common effect can yield a rule with good support, but is more difficult to exploit in the production process.

The core ideas behind the proposed analysis methods follow from the arguments above. That is, to sacrifice support in favor of confidence and simplicity. This is in contrast to established feature importance measures, which aim at finding features that are overall most important (e.g. important on average). We are looking for features and relations that are very important, but possibly only in a few cases. For the sake of illustration consider the subsequent simplified case:

Assume a fictional manufacturer that has produced 10000 product items. An intermediate quality test measures the features A and B. Both features turn out to be uniformly distributed between 0 and 1. There are relations that exist between the features and production errors: (1) Products with feature value A below 0.8 have roughly a 20 percent chance of failing the final quality check. (2) Products with feature value B above 0.99 have roughly a 98 percent chance of failing the test. Overall, about 16% of the sample data reflect faulty products.

To support the quality engineer, we aim to identify the feature that yields the most useful insights. We therefore train a machine learning model on the quality test data and analyze the feature importance. The feature importance score should point the quality engineer to the most useful feature. For illustration, we build a tree model (see Figure 1) and compute the typical feature importance scores (see Table 1). We used the Python XGBoost library to build the tree. For the sake of simplicity and illustration, we limited the number of trees to one. Hence, Figure 1 illustrates the first decision tree in the model[1].

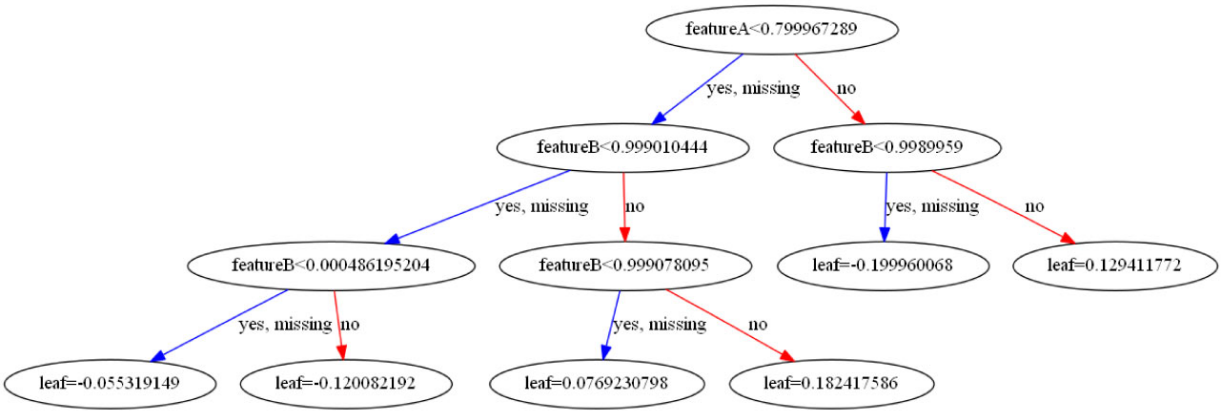

Figure 1. The first tree in the sample case model (using XGBoost)

Table 1. Established feature importance measures for the model in fig.1

Importance Measure	Score for feature A	Score for feature B
Gain	16.119608	7.564618
Cover	946.390676	676.451196
Weight	263	258
Total Gain	4239.456775	1951.671417
Total Cover	248900.747835	174524.408458
Gini-importance	0.680605	0.319395
Average SHAP values	0.866374	0.029806

Table 1 shows the values of typical[2] used feature importance measures for the presented sample model. Note that the typical feature importance measures all consider feature A as most important. However, feature A is of little use in the targeted application domain. Through investigating feature A, the quality engineer can build a rule of the form “IF feature A < 0.8 THEN ERROR”. However, this rule is not of practical use. Sorting out products with a predicted error would remove 80% of the products. Also, 80% of the discarded products have no error (i.e. are false positives). In contrast, an analysis of feature B would result in a prediction rule of the form “IF feature B > 0.99 THEN ERROR”. This rule affects a reasonable number of products and is almost always correct. It therefore enables a useful adjustment of the production process.

[1] Positive values in the leave nodes correspond to the prediction of an error an negative values to the prediction of no error

[2] We include average SHAP values despite their relative novelty, to contrast this aggregate measure with our proposed methods.

Note that the typical feature importance measures fail to rank the features as desired. Table 2 shows the results from the measures that we introduce in this paper. They all correctly rank feature B first. The specifics of each importance measure follow in the corresponding subsections below.

Table 2. Proposed importance measures for the model in fig.1

Importance Measure	Score for feature A	Score for feature B
Max SHAP	0.705525	7.591009
Range SHAP	6.420452	8.724322
Smoothed Range SHAP	61.053000	65.920289
Top-K SHAP (with k=10)	-19.984684	75.860939
Max Main Effect	0.667381	4.482486

3.3 Concept for Feature Importance measures on top of SHAP values

In this section we introduce a range of importance measure on top of SHAP values. All these importance measures leverage the local importance measures provided by SHAP values, to identify "locally interesting" features. By "locally interesting" we refer to features that yield good explanations for errors in at least some cases. This is in contrast to measures that identify features that are frequently or on average important.

Top Importance for Top-K Predictions

The idea behind the analysis of top-k predictions is to zoom in on the most relevant cases. We define top-k by ranking the predictions according to their confidence. In the target application we only care about predicting errors. Hence, we only look at error predictions with high confidence. There are several model specific ways for determining the confidence. Here we use a broad notion of confidence, which does not necessarily imply quantifiable confidence intervals. For instance, we use the SHAP values in our analysis to determine the top-k instances. We then average the local feature specific SHAP values for each of the top-k predictions. Leaning on the notation from Lundberg and Lee [4] and assuming that a high SHAP value refers to the error class, the importance measure is defined as follows[3]:

$$\text{Top-K}_i(f, D) = \frac{1}{k} \sum_{x \in \{d \mid k > |\{d' \in D \mid f(d) < f(d')\}|\}} \phi_i(f, x)$$

Here, f is the model function and D the data sets for evaluating the feature importance, i the feature to score, and $\phi_i(f, x)$ the SHAP value of feature i for model f and data point x. The intuition behind this measure is the following: In the target application we look for predictions with high confidence only. Hence, we only care about features that play a role in the predictions with high confidence. We then aggregate the local SHAP values for the most relevant predictions. An assumption behind this approach is that the top-k predictions are similar from the perspective of the model. That is, they yield predictions with high confidence for similar reasons. This may not always hold true, especially for larger k. Hence, a natural extension of this measure is to consider clusters within the top predictions.

[3] For the sake of simplicity we assume a total ordering of prediction scores.

Max SHAP

The idea behind *Max SHAP* is to look for features that can have a high contribution to the outcome. It is simply defined as the maximum SHAP value for a given feature in data set:

$$\text{Max SHAP}_i(f, D) = \max\{\phi_i(f, x) | x \in D\}$$

Here, we again assume that the outcome of interest is associated with high SHAP values. One may change the sign of the measure or substitute the maximum with a minimum function, if the opposite is the case.

The intuition behind this measure is that it captures the highest effect that a feature can have. The rationale is that a feature with a high maximal contribution is in some cases of high interest. A weakness of this measure is that it is sensitive to outliners. We argue that confidence is much more important in the targeted application domain than support. However, isolated outliers can lead to undesirable results. One may therefore alter this importance measure by taking a certain quantile instead of the maximum value.

Max Main Effect

The idea behind *Max Main Effect* is to look for features that have the highest effects on their own. That is, we explicitly ignore the effect of feature interactions, which are otherwise included in the SHAP values. Leaning on Lundberg et al. [4], we define the measure as:

$$\text{Max Main Effect}_i(f, D) = \max\left\{\phi_i(f, x) - \sum_{j \neq i} \phi_{i,j}(f, x) \,\middle|\, x \in D\right\}$$

Here, $\phi_{i,j}$ is the SHAP feature interaction as defined in [4], and with $\phi_i(f, x)$ we denote the dependency on the model f and the data set D.

The intuition behind this measure is an enhancement of the *Max SHAP* measure. Again, we assume that the outcome of interest is associated with high SHAP values. One may change the foresign of the measure or substitute the maximum with a minimum function, if the opposite is the case. The maximal SHAP value of a feature may be heavily dependent on the interaction of features. In this case, the high contribution of the feature cannot be assessed in isolation. With the *Max Main Effect* measure we aim at identifying features, which have a high contribution regardless of the other features. This is appealing in the targeted use case, because such features support simple decision rules and visual analysis with only one dimension.

Range SHAP

The idea behind *Range SHAP* is to look for features that have a strong impact on the outcome over their value range. Unlike *Max SHAP*, it considers also negative SHAP values. That it takes into account, if certain value ranges of the feature are an indication for no error. (Again, we assume that the outcome of interest is associated with high SHAP values.) We define the measure as follows:

$$\text{Range SHAP}_i(f, D) = \max\{\phi_i(f, x) | x \in D\} - \min\{\phi_i(f, x) | x \in D\}$$

The intuition behind this measure is to look for features that can have a strong impact on the model output in either direction. By looking at the range, we capture the strongest local effects. That is, the score is high if the feature contribution varies strongly between some cases.

Smoothed Range SHAP

The idea behind *Smoothed Range SHAP* is to suppress variance for data points with the same feature value. Due to feature interaction, the same feature value may correspond to different SHAP values. *Smoothed Range SHAP* averages over a sliding window to suppress variation for the similar feature values. Using the formula for moving average the measure is defined as follows:

$$\text{Smoothed Range SHAP}_i(f, D) = max\left(\left\{\frac{1}{W}\sum_{n=m-\frac{W-1}{2}}^{m+\frac{W-1}{2}}\phi_i(f, x_n)\,\middle|\,\frac{W-1}{2} \le m \le |D| - \frac{W-1}{2}\right\}\right) - min\left(\left\{\frac{1}{W}\sum_{n=m-\frac{W-1}{2}}^{m+\frac{W-1}{2}}\phi_i(f, x_n)\,\middle|\,\frac{W-1}{2} \le m \le |D| - \frac{W-1}{2}\right\}\right)$$

Here we assume that the data set D is sorted by x and x_i is the i-th position.

The intuition behind this measure is an enhancement of *Range SHAP*. If a feature interacts strongly with other features, the SHAP values may have a high variance for the same or similar feature values. However, it is more interesting for the target application to find changes in the model output, which correspond to different feature values. Such changes are more helpful for identifying simple prediction rules and are therefore emphasized by this importance measure.

4 Evaluation

To evaluate the proposed importance measures with real-world data, we analyzed quality logs from the German manufacturer SICK AG. Note that we omit some details in the data description that are of minor importance for the evaluation. This is to protect internal information of the manufacturer. The analyzed data cover a time span of roughly one year, and contain records about several ten thousand products of a specific type.

Specifically, we analyzed the test data from a production step A and the outcome of tests from the subsequent step B. We trained a model (i.e. XGBoost classifier) on 34% of the quality test data from step A, with the aim to predict errors in step B. In other words, we aim to predict if a product that passes the tests in step A will fail the tests in step B. If certain tests in A allow for a good prediction of errors in B, one may adjust the tests in step A to filter out corresponding products. One thereby saves the cost of performing step B on products that are going to fail the subsequent quality check.

For model training we considered roughly 100 test parameters from step A as features and the test results from the subsequent step B as label. The test results can have a range of outcomes. That is, a test can be passed, or failed for various reasons. For the sake of simplicity, we reduced the label to the binary outcome “passed” or “failed”.

Within this setup we analyzed the trained model with regards to feature importance. We tested the established and new introduced measures listed in Section 3. The introduced measures require data instances and the model as input. Here we used the training data. We used the XGBoost library for the implementation of the model[4] and the established importance measures. To implement the new proposed measures, we set up on top of the SHAP library for computing

[4] We used default parameters, with exception of scale_pos_weight. This parameter as adjusted to reflect the cost structure for false positives and true positives in the manufacturing line.

SHAP values and SHAP interaction values. The parameters K and W for our measures were set to $K=10$ (i.e. the top-10 predictions) and $W=50$ (i.e. a smoothing window of 50 values).

The analysis results in a ranked list of features for each tested importance measure. Table 3 shows the top 10 results for established measures and Table 4 depicts the top 10 results for the new introduced methods. Feature names are obfuscated to protect internal information of the manufacturer. In both tables we highlight feature Dfleft_col_1593 and Dfleft_col_379. We argue that - amongst the listed features - these two features are of most interest in the targeted application domain. Hence, we expect that an effective importance measure ranks these features high.

Figure 2 and Figure 4 provide the details for our argument about the importance of Dfleft_col_1593 and Dfleft_col_379. The figures show the distribution of feature values as histogram. The Y-axes displays the frequency of feature values and have logarithmic scale. The X-axis is scaled to cover the whole value range. We omitted axis labels to protect internal information of the manufacturer. The shading of the bars encode the percentage of errors (faulty product) in the respective bar (white 0% and black 100% errors). We show all features that are among the top three features of any tested importance measure.

Table 3: Features ranked by established importance measures

Average SHAP	Cover	Gain	Total Gain	Weight
Dfleft_id	Dfleft_col_1125	Dfleft_col_1125	Dfleft_id	Dfleft_id
Dfleft_col_832	Dfleft_col_832	Dfleft_col_832	**Dfleft_col_1593**	Dfleft_col_357
Dfleft_col_705	Dfleft_col_708	Dfleft_col_711	Dfleft_col_738	**Dfleft_col_1593**
Dfleft_col_1593	Dfleft_col_705	Dfleft_col_366	Dfleft_col_1125	**Dfleft_col_379**
Dfleft_col_1126	Dfleft_col_711	**Dfleft_col_1593**	Dfleft_col_832	Dfleft_col_1322
Dfleft_col_932	Dfleft_col_366	Dfleft_col_733	Dfleft_col_357	Dfleft_col_738
Dfleft_col_738	Dfleft_col_751	Dfleft_col_745	Dfleft_col_1214	Dfleft_col_1214
Dfleft_col_1566	Dfleft_col_725	Dfleft_id	Dfleft_col_711	Dfleft_col_1267
Dfleft_col_357	Dfleft_col_738	Dfleft_col_738	**Dfleft_col_379**	Dfleft_col_1266
Dfleft_col_711	Dfleft_col_1126	Dfleft_col_1126	Dfleft_col_1126	Dfleft_col_1566

Table 4: Features ranked by proposed importance measures

Max Main Effect	Max SHAP	Range SHAP	Smoothed Range SHAP	Top-K SHAP
Dfleft_col_1593	**Dfleft_col_379**	Dfleft_id	Dfleft_id	**Dfleft_col_1593**
Dfleft_col_379	**Dfleft_col_1593**	**Dfleft_col_1593**	**Dfleft_col_1593**	Dfleft_col_738
Dfleft_id	Dfleft_id	**Dfleft_col_379**	**Dfleft_col_379**	Dfleft_id
Dfleft_col_738	Dfleft_col_738	Dfleft_col_738	Dfleft_col_738	Dfleft_col_720
Dfleft_col_703	Dfleft_col_932	Dfleft_col_832	Dfleft_col_832	Dfleft_col_1125
Dfleft_col_708	Dfleft_col_1322	Dfleft_col_1322	Dfleft_col_1125	Dfleft_col_357
Dfleft_col_1125	Dfleft_col_357	Dfleft_col_357	Dfleft_col_720	Dfleft_col_1214
Dfleft_col_811	Dfleft_col_1214	Dfleft_col_703	Dfleft_col_1210	Dfleft_col_745
Dfleft_col_1322	Dfleft_col_720	Dfleft_col_1214	Dfleft_col_1266	Dfleft_col_1126
Dfleft_col_1214	Dfleft_col_703	Dfleft_col_932	Dfleft_col_1267	Dfleft_col_711

As Figure 2 shows, the features Dfleft_col_1593 and Dfleft_col_379 have desirable properties for the targeted use case (however the error numbers that support this observation for Dfleft_col_379 are rather small). For both features one can observe an interval with a low error rate. Outside this interval, the error rate is very high. This allows quality engineers to derive simple rules of the form "IF value > X or value < Y THEN ERROR". For Dfeft_col_1593 and Dfleft_col_379 this rule would apply in few cases (remember the logarithmic scale) and – according to the training data – has high prediction quality. Several of the other identified features share this property of feature Dfleft_col_1593 and Dfleft_col_379. However, the corresponding relations are weaker. That is, the error rates outside the respective intervals are lower (lighter color in the figure) and/or the rules would apply to fewer instances. We therefore argue that the feature Dfeft_col_1593 and Dfleft_col_379 are most important and should be ranked high by the importance measures. Note that this is the case for the new introduced measures and in particular for *Max Main Effect* and *Max SHAP*

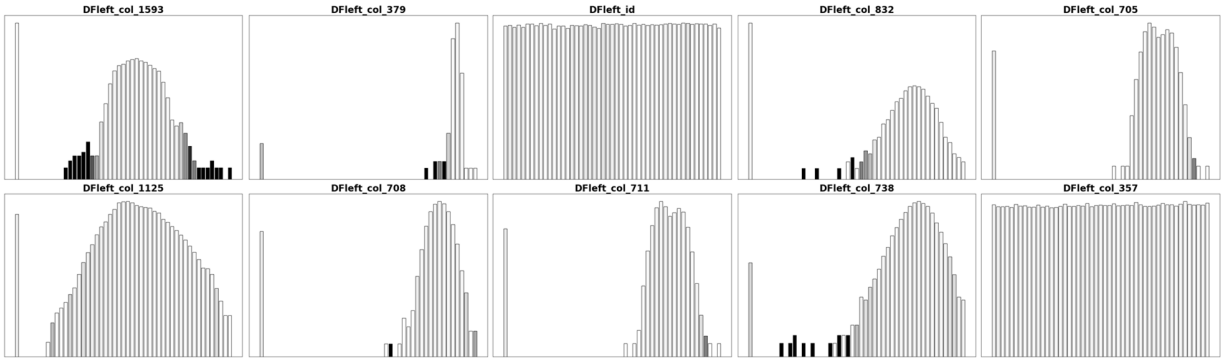

Figure 2: Distribution of feature values for training data. (Y-axes with logarithmic scale, color coded error percentage, white=0%, black=100%).

Furthermore, the features Dfleft_id and Dfleft_col_357 stick out. These features do not show a clear relation with errors in Figure 2. Figure 3 provides more details on these features. For Dfleft_id and Dfleft_col_357 it shows the SHAP values and feature values for each data point in the training data. The plots provide insights on why these features are considered important by some measures. For some data points the features have strong contributions to the model output. However, the variance of the contribution is high, in particular in regions with potential high impact. (This can be seen in the vertical arrangement of points below or above point with high contribution). The figure implies that these features have strong interactions with other features and do not enable good explanations on their own. Thus, the features are not useful for simple prediction rules that consider only one value. It may be interesting to further investigate the feature interaction in search for slightly more complex relations (e.g. comprising two or three features). However, this is beyond the scope of this paper and subject to future work.

As a side note, the feature Dfleft_id is an identifier value that roughly resembles a counter. This feature is obviously not useful or general predictions and a domain expert would discard it for prediction models. However, it can be useful to analyze data in retrospect. The fact that this feature has an impact in the model hints at production problems in certain time frames. This insight can be helpful for quality engineers.

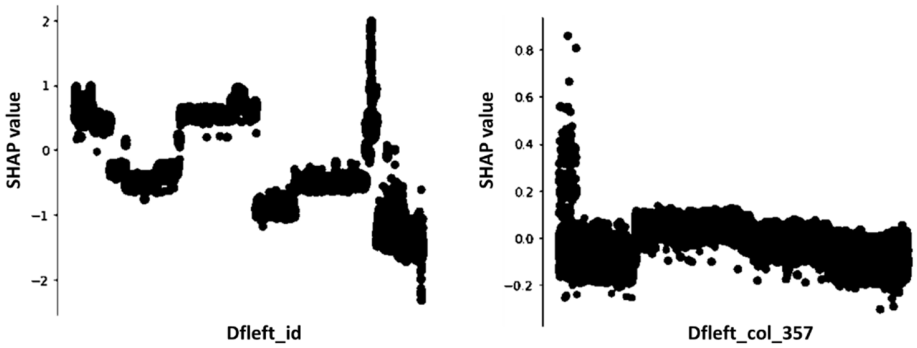

Figure 3: SHAP values for Dfleft_id and Dfleft_col_357

For our analysis we split the data into training and test data. However, the evaluation on the training data may be more meaningful in the targeted use case. This is because the main goal is not to build a general prediction model, but to point quality engineers to interesting phenomena in the data (possibly only in retrospect). In this case the human can judge the validity of the findings based on domain knowledge and validation through a separate test set may be of less importance. (See [2] for a more detailed discussion of using test sets or training sets for evaluating approaches of interpretable machine learning). In our test case, the insight from the model would have been available after 34% of the analyzed period and before the test data is available. However, it is still interesting to see how derived rules would have played out on the test data (see Figure 4). Overall, we find that insights from the training set continue to be valid in the test set. Yet, the total frequency of errors decreased. This is not surprising, since the observed process is subject to continuous improvements by the quality engineers.

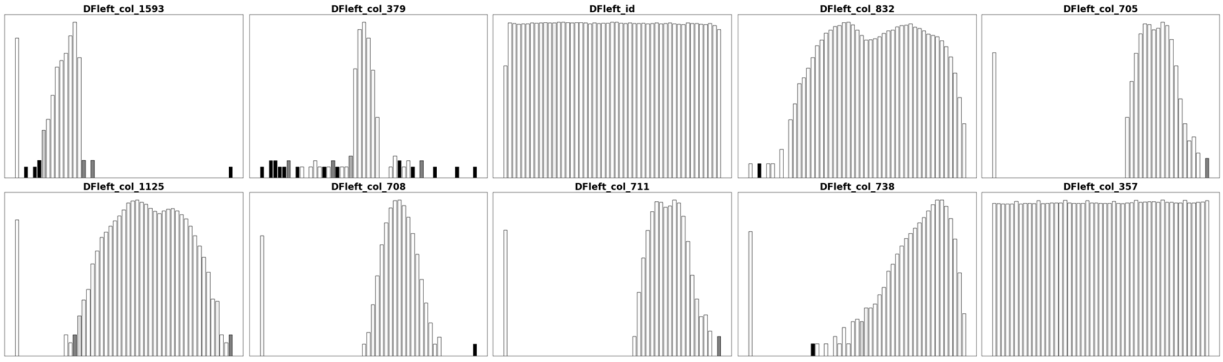

Figure 4: Distribution of feature values for test data. (Y-axes with logarithmic scale, color coded error percentage, white=0%, black=100%).

5 Related Work

Recently there has be a strong increase of interest in works on interpretable machine learning[5]. Such works aim to address the opaqueness of many machine learning models and to provide human understandable explanations. (See [1] or [2] for an overview.) Works from this category are generally related to our work. However, we are not aware of any approach that is tailored to the specific needs of quality engineering in manufacturing. The existing works typically aim at

[5] We use the term to refer to approaches of explainable ML as well, while acknowledging an ongoing debate on how to specify the difference between the terms.

providing a holistic understanding of a model. In contrast, our work aims at electing specific insights from the model that are helpful for quality engineers.

More specifically related to our work are works on feature importance. We can distinguish between global importance measures and – more recently introduced – local importance measures. Global importance measures like gain [6], or and similar measures for tree models (e.g. as implemented in [7]) are well established. They are some form of aggregate that aim at capturing the typical (e.g. average) importance of a feature. This type of aggregate conceals features that are only important in some rare cases. However, such rare cases can be of most interest in quality management for manufacturing.

Local importance measures such as LIME [8] and SHAP [4] explain feature importance for individual data points. This is useful for understanding specific predictions. However, they require additional analysis to gain insights beyond the scope of single data points. Closely related to our work is the work of Lundberg et al. [5], who address visualizations on top of SHAP values. Such visualizations are helpful for a detailed analysis. Yet, they show individual data points and leave the interpretation to the user. In contrast, our work points quality engineers to features of interest. Lundberg et al. [5] also use mean SHAP values as global importance measure. This is related to our work in the sense that it is an importance measure on top of SHAP values. However, the measure is not designed for the specific needs of our use case and – as our experiments show – it is less effective than our measures in this context.

Cohen et al. leverage Shapley values in a feature selection mechanism [9]. Thereby they indirectly define feature importance on top of Shapley values, similar to [5]. However, they focus on maximizing the overall performance of classifiers. That is, like other global importance measures, they aim at capturing the typical importance of a feature. Therefore – unlike the measures that we propose – their analysis is not tailored to identifying features that are useful for quality engineers.

6 Conclusion and Future Work

In this paper we introduced feature importance measures that are tailored to the needs of quality engineers in manufacturing. They leverage SHAP values to identify locally important features. Along synthetic and real-world data we demonstrated the benefits of these measures. Our tests indicate that “Max Main Effect” and “Max SHAP” are most promising among the five introduced measures. In future work we plan to further investigate the strength and weaknesses of the proposed measures to derive recommendations about their application.

One other direction of future work is expanding the concept to better leverage local information on feature interaction. In this paper we focus on strong effects that relate to a single feature. The rationale is that such effects can be easy illustrated and comprehended by humans. However, we intend to explore more complex effects that occur for certain combinations of feature values. Here, the analysis of SHAP interaction values is a natural extension to the concepts that we introduce in this paper.

Another direction of future work is to build regional models on top of the identified relevant features. The goal is to make some interesting phenomena understandable that the model picked up on. That is, beyond identifying interesting features, we aim to directly provide interesting explanations. Again, in this context we mean by interesting that the explanation is simple and yields good prediction quality. We plan on expanding on the idea to build surrogate trees by (a) building trees only for regions and not as surrogate for the entire model, (b) building trees on a subset of features, and (c) training on the original data and not the model predictions. The rational is to produce local

explanation of the real world and not the model (i.e. aiming to produce a simple model that works well in certain situations).

Overall, we believe that the foundation laid by SHAP values provides many opportunities for eliciting relevant insights from a machine learning model. There is a big spectrum between global importance measures and data point specific feature contributions that provide rooms for insights on various levels of granularity. With the use case of quality engineering in manufacturing we provide an example for the benefits of such analysis.

Acknowledgements

This project was funded by the German Federal Ministry of Education and Research, funding line "Forschung an Fachhochschulen mit Unternehmen (FHProfUnt)", contract number 13FH249PX6. We also like to thank the manufacturer SICK AG for the cooperation. The responsibility for the content of this publication lies with the authors.

References

[1] L. H. Gilpin *et al.,* "Explaining explanations: An overview of interpretability of machine learning," in *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, 2018, pp. 80–89.

[2] C. Molnar, "Interpretable machine learning," *A Guide for Making Black Box Models Explainable*, vol. 7, 2018.

[3] H. Ziekow *et al.,* "Proactive Error Prevention in Manufacturing Based on an Adaptable Machine Learning Environment," *From Research to Application*, p. 113, 2019.

[4] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.

[5] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," *arXiv preprint arXiv:1802.03888*, 2018.

[6] L. Breiman, *Classification and regression trees*: Routledge, 2017.

[7] xgboost developers, *XGBoost Documentation - Python API Reference.* [Online] Available: https://xgboost.readthedocs.io/en/latest/python/python_api.html. (last access 2019-07-15)

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[9] S. Cohen, G. Dror, and E. Ruppin, "Feature selection via coalitional game theory," *Neural Computation*, vol. 19, no. 7, pp. 1939–1961, 2007.