A Reference Process for Judging Reliability of Classification Results in Predictive Analytics

Simon Staudinger[a], Christoph G. Schuetz[b] and Michael Schrefl[c]
Institute of Business Informatics, Data and Knowledge Engineering, Johannes Kepler University Linz, Austria

Keywords: Business Intelligence, Business Analytics, Decision Support Systems, Data Mining, CRISP-DM.

Abstract: Organizations employ data mining to discover patterns in historic data. The models that are learned from the data allow analysts to make predictions about future events of interest. Different global measures, e.g., accuracy, sensitivity, and specificity, are employed to evaluate a predictive model. In order to properly assess the reliability of an individual prediction for a specific input case, global measures may not suffice. In this paper, we propose a reference process for the development of predictive analytics applications that allow analysts to better judge the reliability of individual classification results. The proposed reference process is aligned with the CRISP-DM stages and complements each stage with a number of tasks required for reliability checking. We further explain two generic approaches that assist analysts with the assessment of reliability of individual predictions, namely perturbation and local quality measures.

1 INTRODUCTION

Organizations employ data mining to discover patterns in historic data in order to learn predictive models. The **CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata **M**ining (CRISP-DM) (Wirth and Hipp, 2000) serves as guideline for the proper application of data mining, aligning data analysis with the organization's business goals. The CRISP-DM comprises six stages: (i) business understanding, (ii) data understanding, (iii) data preparation, (iv) modeling, (v) evaluation, and (vi) deployment. The CRISP-DM is widely employed in data-analysis projects in various domains (Caetano et al., 2014; Moro et al., 2011; da Rocha and de Sousa Junior, 2010).

The predictive models that are learned from data allow decision-makers to make predictions about future events of interest and act accordingly. The predictions of learned models may be more or less accurate, raising the question about the reliability of individual predictions. Consider, for example, a classification model that allows a bank employee to decide whether a specific client will default on a requested loan in the future. If similar cases from the past cannot be clearly associated with a specific outcome, or similar cases have not been present in the input data, the prediction will not be very reliable. Nevertheless, the model will come up with a prediction, but it is up to the analyst to judge on the reliability of that prediction. If actions are based on unreliable predictions, this could lead to potentially costly failures and missed business opportunities.

When it comes down to judging the reliability of an individual prediction, the accuracy and similar quality measures of the entire model are often the only guidance available. The accuracy is a summary of the overall performance of the model, which is not enough to make a robust statement about the reliability of an individual prediction for a specific input case. For example, a model's training data points may not be evenly distributed in the feature space. A new data point located in a densely populated part of the feature space may obtain a more reliable prediction than a new data point in a sparsely populated part. The overall accuracy of the model would be the same for both predictions. Other works (Moroney, 2000; Capra et al., 2006; Reimer et al., 2020; Vriesmann et al., 2015) have shown that it can be fruitful to investigate models and collect model metrics not only on a global level but also in the local area of interest of the data. Calculating local quality measures, e.g., the local accuracy of a model for a specific prediction case, conveys a better impression of the reliability of an individual prediction than using the global measure

[a] https://orcid.org/0000-0002-8045-2239
[b] https://orcid.org/0000-0002-0955-8647
[c] https://orcid.org/0000-0003-1741-0252

Staudinger, S., Schuetz, C. and Schrefl, M.
A Reference Process for Judging Reliability of Classification Results in Predictive Analytics.
DOI: 10.5220/0010620501240134
In *Proceedings of the 10th International Conference on Data Science, Technology and Applications (DATA 2021)*, pages 124-134
ISBN: 978-989-758-521-0

for the model as a whole. Furthermore, taking inspiration from the field of numerical analysis, where the condition number describes the magnitude of change of a function's output in the face of small changes of the function's input (Cheney and Kincaid, 2012), we propose to employ *perturbation of input data* for individual prediction cases in order to better judge on the reliability of a prediction: If small changes in the input data of a specific prediction case lead to a completely different prediction, the reliability of that prediction is questionable. For example, in the context of the prediction of a loan default by a prospective customer, if a small change in the monthly income leads to an opposite prediction for loan default, the prediction may not be very reliable since a small change in monthly income is very likely to happen.

In this paper, we propose a reference process based on CRISP-DM for building predictive analytics applications that allow analysts to judge the reliability of individual classification results. We illustrate that generic reference process over a real-world data set[1] from a telemarketing campaign of a Portuguese bank (Moro et al., 2014). We propose that in order to better judge the reliability of individual predictions, an analyst must consider the actual input data, the data used for training the predictive model, and the specific procedures regarding collection and preparation of the data. The proposed reference process defines tasks along the entire CRISP-DM life cycle. During the business understanding stage, the reference process requires developers of predictive analytics applications to choose between different approaches for reliability checking. The data understanding and data preparation stages then require the gathering of further information about the available data and potential reliability problems arising from data collection. The information about the available data subsequently serves to select and configure the approaches for reliability checking, which are then evaluated regarding suitability for the given analytics problem. When the model is deployed, the additionally defined modules for reliability checking can be used by the analyst to better judge the reliability of an individual prediction and to better decide how to properly act on the prediction. Given an already trained black-box or white-box model, the tasks assisting in judging the reliability results can still be added, opening up the possibility to use the reference process for prediction models which are already in use.

We follow a design science approach (Hevner et al., 2004). The goal of design science research is the development of design artifacts that solve a practical problem. A design artifact can be a model but also a method or software tool. The contributions of this paper are two design artifacts: a generic reference process and the use of perturbation options within the reference process.

The remainder of this paper is organized as follows. In Section 2, we review related work. In Section 3, we give an overview of the proposed reference process. In Section 4, we describe the tasks related to data collection and data preprocessing as well as guidelines to adapt a specific classification problem. In Section 5, we focus on tasks related to modeling and evaluation. In Section 6, we discuss deployment of the reference process, illustrated on the use case. In Section 7, we conclude the paper with a summary and an outlook on future work.

2 RELATED WORK

In various domains, dynamic classifier selection (DCS) is used in multiple classifier systems to find the best classifier for a given classification problem and a given classification case (Cruz et al., 2018). In comparison to majority voting, where the final outcome is the class predicted by the most classifiers, DCS aims to find the best-fitting classifier for the individual case and use the prediction of this classifier as the final outcome. Different approaches to find the best classifiers, including the use of local regions within the feature space were already examined (Cruz et al., 2018; Didaci et al., 2005; Vriesmann et al., 2015).

The problem of judging reliability of predictions is related to metamorphic software testing. Metamorphic testing is concerned with the oracle problem in software testing. This problem occurs in systems where the correct behavior cannot be distinguished from the incorrect behavior due to missing formal specifications or assertions. For more information on metamorphic testing we refer to other works (Barr et al., 2015; Chen et al., 2018; Segura et al., 2016). The central points of interest in this area are the metamorphic relations, which describe the differences between input and output of a software system. The output is called a follow-up test case and can again be used as input for a metamorphic relation, creating a possibly infinite amount of potential test cases. If the real outcome of the software system is different than the expected one there is likely an error within this system. An example of a metamorphic relation is the case of two search queries, where the second query restricts the first query. If during metamorphic testing

[1]The employed data set is available on the UCI Machine Learning Repository via the link https://archive.ics.uci.edu/ml/datasets/bank+marketing (accessed: 1 March 2021).

the result of the second query is not a subset of the first query, the query implementation is faulty. The difficult and challenging part in metamorphic testing is to find metamorphic relations suitable for the existing software models to be tested. Metamorphic testing has also been used in the domain of predictive analytics. There are multiple works which examine the use of metamorphic relations to ensure the soundness of machine-learning classifiers (Xie et al., 2011; Saha and Kanewala, 2019; Moreira et al., 2020).

The main contribution of the paper is the reference process for judging the reliability of predictive analytics results. To the best of our knowledge, no other such process has been proposed yet. The proposed reference process contains tasks that include aspects of related work. The tasks adapt the following ideas from related work. First, while DCS focuses on the selection of the best classifier through the use of local measures, DCS does not consider using local measures to evaluate the reliability of individual predictions made by the same classifier. We use the local-region approach within our reference process to find differences between global and local measures for individual cases within the same classifier. Second, metamorphic testing uses slightly different input cases in combination with a metamorphic relation to ensure the correctness of a system. We employ the approach of input-case perturbation to test if small input variations influence the prediction.

3 REFERENCE PROCESS: OVERVIEW

We propose a reference process for judging reliability of classification results, which we organize along the CRISP-DM stages (Figure 1). In order to arrive at predictive analytics applications that facilitate judgment of reliability of individual predictions, at each stage of the CRISP-DM, additional tasks have to be considered by developers and analysts. During business understanding, appropriate approaches for reliability checking must be selected under consideration of the business and data mining goals. An example of such an approach is to use *perturbation of test cases* to find features in the data that are especially sensitive regarding the prediction. Data understanding in CRISP-DM is concerned with collection and review of the available data. Judging reliability at later stages requires the use of metadata and, therefore, the metadata need to be collected and documented during the data-understanding stage. Gathered metadata may include, for example, the scale of the feature, the precision of the measured feature value, and existing data restrictions, e.g., allowed feature ranges.

Raw data are typically not suitable as training data. Hence, the data preparation stage typically applies a multitude of techniques to increase the data quality and provide the data in a suitable format for training. Applying data preparation techniques can result in various reliability issues and, therefore, the employed data preparation techniques should be documented. For example, issues may arise when handling missing data or discretizing values.

In the modeling stage, two approaches for reliability checking are considered: the perturbation of input cases and the evaluation of local quality measures. Using the information from the previous steps allows the modeler to choose and configure these approaches with regard to the actual analysis case. Input perturbation aims to find sensitive features and possible borderline predictions. Local measures allow to compare the global performance of the model with the performance of the individual case. The configuration of these approaches is done for every data mining problem. Once defined, these approaches can be used to judge the reliability of different input cases once the model has been deployed.

The parameters of the approaches for reliability checking need to be fitted to the data mining problems and the individual use cases. A first assessment of the chosen parameters will be conducted during the evaluation stage. Example parameters include the range of the perturbation or the distance for the calculation of the local measures.

The last stage in the CRISP-DM is the deployment of the trained model. New input data that are passed to the model will receive a prediction. An analyst can use the defined and assessed approaches to judge the reliability of the received prediction. If none of the predefined approaches is suitable, the approaches can be adapted or new approaches can be defined in order to improve the quality of the judgment of the reliability of a prediction.

We note that the presented reference process, although illustrated and discussed on the example of classification, is not limited to the problem of classification but potentially also applicable for other prediction problems. In the following, we present in more detail the different tasks of the reference process along the different stages of the CRISP-DM.

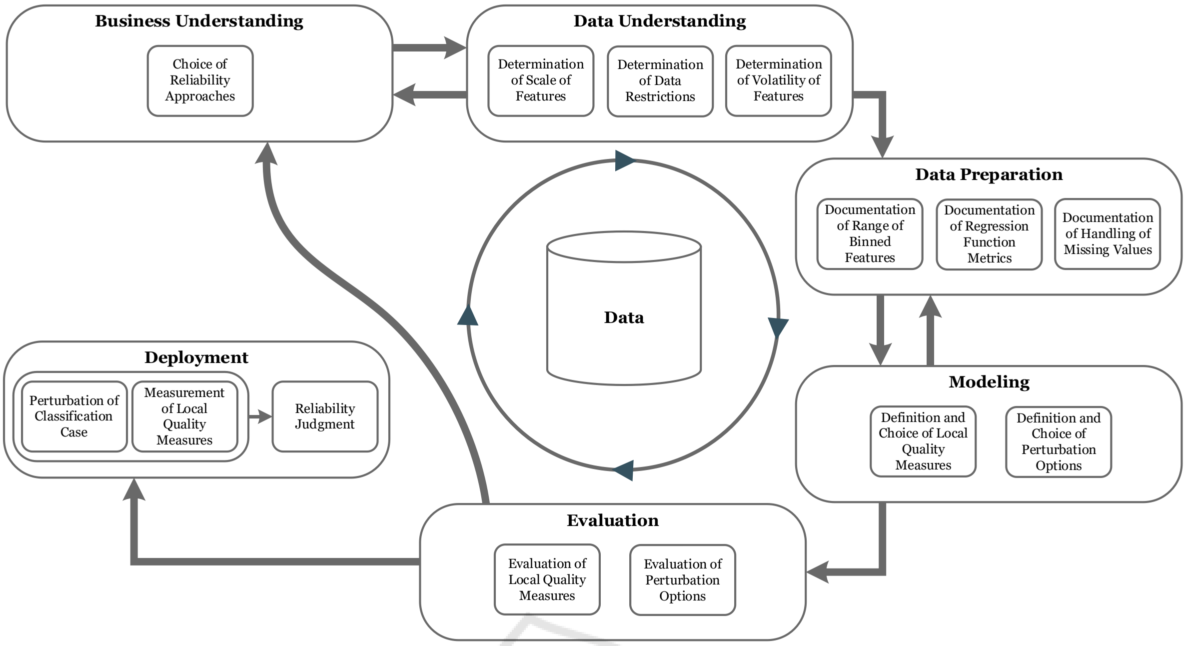

Figure 1: Overview of the reference process for judging reliability of classification results along the CRISP-DM stages.

4 BUSINESS UNDERSTANDING, DATA UNDERSTANDING AND DATA PREPARATION

Business understanding is concerned with the definition of both business and data mining goals that should be reached within the data mining process. In addition to these goals, it must be decided in which way the reliability of the predictions should be assessed. Given different reliability-checking approaches, the fit of each approach for the use case must be considered, and whether it is possible to employ those approaches. For example, if local measures for each new case are desired there must be access to the training data. If the organization is interested in finding sensitive input features, the perturbation approach may be a good fit in order to judge the reliability of individual predictions. In this paper, we describe two approaches, but future work will investigate other approaches, e.g., the use of genetic algorithms to find possible border cases.

There are different types of data that are of interest for the analysis, e.g., structured data, streaming data, images, and videos. During data collection and extraction, inaccuracies may slip into the data, or it might even be impossible to capture the exact value. If we know the range of these inaccuracies, or at least that a feature might be different compared to the real value, we can use this information later when judging the reliability of a prediction. A prime example of inaccuracy of the collected data is sensor imprecision (Guo et al., 2006). Information about imprecision of the collected data should be stated for every collection method that is employed.

Data understanding and data preparation are necessary stages before a data mining algorithm can be used to train a predictive model. Every saved data point has to be extracted from the real world. In this paper, we focus on the case of structured data and we provide guidelines that can be followed during the data understanding and data preparation stages in order to subsequently be able to judge the reliability of a prediction based on the collected information about the following characteristics of the data.

Level of Measurement. Each feature has a level of measurement, which should be determined during the data understanding. The level can either be nominal, ordinal, or cardinal.

Volatility of Feature Values. Features may have different degrees of volatility. For example, a feature that states if a customer has already been contacted can assume the values “yes” or “no”. If the month of a phone call is documented, there is no inherent uncertainty about the real-world month. In contrast, for example, wind speed may quickly change. The measured wind speed may be different if measured a few minutes earlier.

Feature Value Restrictions. Restrictions regarding feature values can either be formal or domain-specific. An example of a formal restriction is the age of a person, which cannot be negative. A possible example of a domain-specific restriction could be that a person under the age of 18 is not allowed to open a long-term deposit. The valid age range starts with the value 18.

Data Accuracy. Depending on how a feature value is collected, different kinds of precision can be stated. First, the accuracy may depend on whether the values were rounded when gathering the data. For example, a salary given in thousands is most likely not the exact value but in reality is slightly less or more. Therefore, it is important to document the range of the rounded value. For example, the interval $[950, 1050]$ may describe a given value of 1000 further. If any kind of technical device is used to collect the data, it should be stated how accurate the collected values are compared to the real values. This may be given either as an absolute value, e.g., the measured value is correct within the range $\pm$ 0.5, as a relative value, e.g., $\pm$ 1%, or as accurate to a certain number of decimal places. If features are ordinal or nominal, and other feature values would fit as well for an individual data point, this should be documented as well.

After the data have been collected and the quality documented, the data preparation stage includes different techniques to improve the data quality and to transform the data in a format that can then be used to train the model. Similar to data understanding, the employed techniques may be the reason why inaccuracies were included in the data. In order to tackle data quality issues concerning accuracy, consistency, incompleteness, noise, and interpretability, values can be altered or estimated. A possible example may be the rounding of values, e.g., a given income can be rounded to hundreds. Furthermore, dealing with missing values in the data is another potential concern when assessing the reliability of a prediction. Since these values are just estimates, the information carried by the values is less precise than the actual values. In the following, we enumerate several cases where reliability-related information should be collected.

Binned Features. To reduce the number of different data points, binning can be applied to raw data. As with rounded values, the binned value is an estimate, and its possible bin range should be documented.

Regression Functions. If a regression function is used during data preparation in order to obtain the feature values, evaluation metrics of the regression function should be documented, e.g., the R^2 value.

Missing Values. Different approaches exist for dealing with missing values. For example, the mean value of all available feature values may substitute for a missing value. Alternatively, the standard value, a placeholder, or a more sophisticated completion mode, e.g., Markov chains, can be used. The method for completing missing values should be documented.

The list of data preparation techniques mentioned here is not exhaustive. In any case, the performed data preparation technique, when altering the data, should be documented and that information can then be used in the further phases of the reference process to judge the reliability of the predictions.

5 MODELING AND EVALUATION

In this section, we discuss the definition and evaluation of reliability-checking approaches. Depending on the classification problem, different approaches and parameters may be useful, which have to be evaluated. The selected and evaluated approaches can then be used by the business user to get an insight into the reliability of a specific prediction. This paper describes two approaches: the perturbation of test cases and the use of local quality measures.

5.1 Perturbation

A perturbation option is a formal description of a function that generates neighboring values of a feature from the case considered for prediction, slightly different from the case's feature value. The perturbation approach was inspired by the field of numerical mathematics, where the condition number measures the effect of a changed input to the output of a given function (Cheney and Kincaid, 2012). Each perturbation option is assigned to a feature and takes the original value of the feature as input. Depending on the perturbation option, there can be additional input parameters, e.g., the precision of a sensor, which are required to apply the perturbation option during deployment. The perturbation options often rely on a specific scale of the feature. Since a defined perturbation option may not only be useful for one feature, the scale of the feature is added to the definition. Furthermore, each perturbation option is assigned one of three levels. These levels assist the analyst with

Perturbation Option	
Name:	percent10
Scale of Feature:	cardinal
Perturbated Feature:	featureName
Additional required values:	-
Level:	Orange
Generation Algorithm:	

```
for(i = 1; i<=10 ; i++){
  nextPertVal(orgValue * (1 + i/100))
  nextPertVal(orgValue * (1 - i/100))
}
```

Figure 2: Option 10percent.

Perturbation Option	
Name:	sensorPrecision
Scale of Feature:	cardinal
Perturbated Feature:	featureName
Additional required values:	sensorPrecision%
Level:	Red
Generation Algorithm:	

```
for(i = 1; i<=10 ; i++){
  nextPertVal(orgValue * (1 + (sensorAccuracy%/1000 * i)))
  nextPertVal(orgValue * (1 - (sensorAccuracy%/1000 * i)))
}
```

Figure 3: Option sensorPrecision.

judging the reliability of a prediction by indicating which information from the data understanding and data preparation stages led to the definition of the perturbed test case. The possible levels are red, orange, and green. A red-level perturbation option states that the prediction should not change while using values generated with this perturbation option. An orange-level perturbation option leading to a changed result needs further examination since, the considered case for prediction could be a border case to another classification label. A green-level perturbation option states that the prediction is expected to change when perturbing the values, which means that the change does not indicate a problem with the reliability of the actual prediction.

Figure 2 and Figure 3 show example perturbation options. The `orgValue` placeholder represents the value from the original test case and `nextPertVal` returns the perturbed value which is further used to create the perturbed test case. The first option (*percent10*) describes a relative perturbation of a cardinal value. The original value is increased/decreased relatively to the considered case's value, from $\pm$ 1% up to $\pm$ 10%. Since an increase/decrease of a value can lead to a changed prediction when reaching a border case in the feature space, the level of the perturbation option is orange, signifying that a changed prediction can happen when perturbing the original case, which requires the attention of the analyst. The second perturbation option (*sensorPrecision*) is a red-level option. The *sensorPrecision* option is intended to be used for a value generated by a sensor. Every sensor value is captured with a certain accuracy depending on the device used for sensing. The real value lies anywhere within the precision range. Since we do not know the real value, the values in the precision range should not change the prediction of a test case. The red-level states that a changed prediction should not happen for these values, because we do not know the real-world value within this perturbation range. If the prediction changes for a variation of the input value that is within the sensor precision, the prediction is not reliable.

A perturbed test case corresponds to the original case for which we want to predict a value but with one or more features swapped according to the value supplied by the perturbation option. Each perturbed test case is then passed to the model to obtain a prediction. Predictions are then used to judge the reliability of the original test case. For a combination of different perturbation options, the highest level of the options is assigned to the test case. Red is the highest level, followed by orange and green.

In the previous section, we described the collection and preprocessing of the data as well as the additional metadata that should be collected for judging reliability. The additional information can be used to define perturbation options based for a specific use case. The perturbation options may be reused or adapted for different features and classification problems. A guideline for the definition and specification of different perturbation options depending on different criteria follows.

Scale of Feature. Any feature for which the scale is known can use one of the following kinds of perturbation options. Nominal features do not provide any kind of order. Therefore, one option would be to swap all available values. For ordinal features it is also possible to swap all available features, but since we have an order given it would be possible to use a fixed number of steps in one or both directions. Cardinal features can either be altered by relative or absolute values. An example of a percent perturbation option can be seen in Figure 2.

Volatility. Knowing the volatility of a feature allows for the definition of different perturbation options. If a feature is known to be volatile in the real world it may be good to create perturbation options for checking the values around the original feature value. If a feature is known to be stable in the real world, it is less likely that there are inaccuracies in the values.

Accuracy. The accuracy of the data can also be a good hint for the creation of perturbation options for a feature. The known accuracy range or a known deviation can be used by a perturbation option to create values within an interval. The prediction is expected to stay the same for all values within the interval. The perturbation options dealing with this information should be assigned a red level. If values are estimated, e.g., using a default value for a missing value, binned feature values, or regression function values, it can also be useful to create perturbation options returning neighboring feature values. The creation of accuracy-related perturbation options can be done for any kind of known imprecision within the feature values.

Restrictions. If there are any value restrictions identified during the analysis of the data, these restrictions can be used to reduce the number of perturbed values. If a perturbation option would return a value violating the restrictions, the perturbed value can be discarded and is not used for further analysis.

The larger the number of perturbed cases that are created and for which a prediction is obtained, the more meaningful the judgment of the reliability of the original case will become. In the best case, all possible perturbed test cases are created and predicted. Since it is not practical and sometimes not even possible to create all potential perturbed cases due to restrictions regarding execution time or number of cases, it is necessary to use a specified operation mode. Each of the following operation modes creates perturbed cases until either all possible combinations are provided, or the specified execution time is reached. In the following, we describe three different operation modes for perturbation.

Operation Mode: Full. The first operation mode computes all possible combinations of perturbed cases. The operation mode starts by iterating through all the available perturbation options. There is no preferred order in which the perturbation options should

age	job	marital	default
20	technician	single	no
21	technician	single	no
19	technician	single	no
22	technician	single	no
20	technician	married	no
20	technician	divorced	no
20	technician	single	yes
21	technician	married	no
21	technician	divorced	no
...	...	...	...

Perturbation Option 1	Perturbation Option 2	Perturbation Option 3
20	single	no
21	married	yes
19	divorced	
22		

Figure 4: Operation mode: full.

be used. Each value of the selected perturbation option is inserted in the original case for the specified feature and the perturbed case is used as input for the model to obtain a prediction. In case that multiple perturbation options on different features are used, combinations of perturbation options are created. Combination of options continues until all possible combinations have been created and prediction values have been obtained. An example can be seen in Figure 4. The first line shows the original test case. Given the values from the perturbation options shown at the bottom of Figure 4, the displayed perturbed test cases are generated. Each value from the perturbation options is inserted into the original test case. All possible combinations of values are created.

Operation Mode: Prioritized. The second perturbation mode is similar to the first operation mode. If there is no time restriction in place, both will come up with the same perturbed cases. Nevertheless, the order in which the cases are created is changed. This mode allows the user to explicitly mark perturbation options as more important than others. For example, if an analyst knows that the features income and age are very important in the analysis, it can be stated that perturbation options applied on these features should be used with priority over other perturbation options. Each value of the perturbation option is inserted into the original case for a specified feature and is used to predict a result. Afterwards, all the combinations with the prioritized perturbation options are created.

age	job	marital	default
20	technician	single	no
21	technician	single	no
19	technician	single	no
22	technician	single	no
20	technician	single	yes
21	technician	single	yes
19	technician	single	yes
22	technician	single	yes

Figure 5: Operation mode: selected.

Following this, all the remaining perturbations will be created, predicted, and can be used for judging the reliability.

Operation Mode: Selected. The third mode only takes a defined number of perturbation options for the generation of the perturbed cases. Instead of generating all available perturbed cases, only combinations of predefined perturbed cases are generated and predicted, leading to a reduced subset of the cases with respect to the cases retrieved using Full and Prioritized mode. Columns with assured values or values that are uninteresting from a business perspective can be skipped for the benefit of faster execution time. There is no predefined order in which selected perturbation options are used. Figure 5 shows an example for the resulting perturbed test cases when only Perturbation Option 1 and Perturbation Option 2 from Figure 4 are used during the selected mode.

After creating the perturbation options based on the gathered information these perturbation options can be assessed if they are fit for the classification problem. Different perturbation options use parameters such as the percent value in the perturbation option shown in Figure 2. These parameters may be unsuitable for a specific feature or classification problem. During the evaluation stage of the CRISP-DM, it can be tested if the chosen perturbation options and parameters are suitable. For example, changing the percent values/absolute values when they are not suitable for the given test case. If broader or narrower perturbation options are needed to receive reasonable results while applying the perturbation options on test cases, they can be created or adapted before the model is deployed.

5.2 Local Measures

A different approach for gathering information about the reliability of a prediction is the use of local measures. If there is access to the training data, various local measures can give an insight into the reliability of the prediction. An example of a local measure is the local accuracy (Woods et al., 1997). The overall accuracy provides the number of correct predictions compared to the number of wrong predictions for the whole model whereas the local accuracy differentiates between different areas within the model. It can be useful to know if the accuracy changes in different areas of the feature space. Consider an overall prediction accuracy of 85% for the entire model. There may be different subspaces that have a different local accuracy compared to the overall accuracy. Therefore, we can evaluate the local accuracy using either a predefined number of training set neighbors or all training set neighbors in a predefined distance around the new classification case. The surrounding neighbors of a test case are selected using a distance function, which calculates every distance between the new test case and the given training set cases using, for example, Euclidean distance. This can give an insight into how well the algorithm performs in the input data space around the new classification case, compared to the overall accuracy of the whole model. A prediction for a case where the local accuracy of the model is much lower than the overall accuracy of the model as a whole should be mistrusted. For example, a new case in a feature space with only 60% local accuracy compared to 85% for the whole model has a significantly worse performance than stated by the global measure. To receive a meaningful judgment, there needs to be a reasonable number of training set neighbors affected by the calculation. If this is not the case the number of training set cases affected by the local accuracy should be increased, or the local accuracy should not be considered when judging the reliability of the prediction until the number is adapted.

Similar to the local accuracy, there is the local class ratio. If we have access to the training data of the model, we can use this information to provide an insight into the ratio of the same training set neighbor labels as our prediction compared to the different neighboring labels in the training set. We are interested in how many of these neighbors have the same label as our prediction. Accuracy measures just the overall correctness of all test cases. Since we are interested in a specific predicted label, the local class ratio states the amount of training set neighbors with the same real label within a predefined number of neighbors or all neighbors in a predefined distance around the new classification case. To receive a meaningful judgment, there needs to be a reasonable number of neighbors affected by the calculation. If this is not the case the number of training set cases affected by the local class ratio should be increased, or the local class ratio should not be considered while judging the reliability of the prediction until the number of neighbors in the training set is adapted to a suitable number.

The local measures described in this paper are just examples. There may be more local measures available depending on the problem or algorithm. For example, the number of classification labels in each node of a decision tree which state the distribution of them in the current node. The next section describes the use of the previously defined modules and shows how they are used to judge the reliability of a specific prediction.

6 DEPLOYMENT

During the deployment stage, the trained model and its reliability modules are deployed into production within the organization. Subsequently, new cases are served to the model as input for classification in order to make a prediction. An analyst can now use the previously defined perturbation modules to receive perturbed cases assisting with the judgment of the reliability of the prediction. Depending on the perturbation mode and the chosen perturbation options, multiple perturbed test cases are generated and predictions for those cases are obtained. The analyst receives an overview of how the prediction would change if the input values were changed according to the chosen perturbation options.

In case of multiple perturbation options being used for a single perturbed case, the perturbed case and its result are assigned the highest level of the used perturbation options. If the level is red, then the test case represents a case where the prediction should not change. An orange-level perturbed case could be a border case and, therefore, requires further consideration through an analyst.

For demonstration purposes, we trained a logistic regression function over the data from a real-world telemarketing campaign (Moro et al., 2014). The model aims to predict if a customer will subscribe to a long-term deposit when contacted via phone, by using different features such as age, marital status, education of the customer, contact information about previous campaigns or the existence of any kind of loan. The output of the prediction can be either “yes” or “no”. The previous call duration feature was excluded from the model because we do not know the duration of the phone call before performing it, thus it cannot be used for making a prediction upon which a decision is made whether to contact a potential client. We implemented several perturbation options and used those perturbation options on different new cases. As operation mode for the perturbation, we chose Full, but due to space considerations we only provide a small extract out of the generated perturbed test cases for illustration purposes. In addition, since we have access to the training data, we calculated the local accuracy for new cases.

Figure 6 shows example perturbed cases for a given test case. We perturbed the categorical features marital and default with available values and used the perturbation option from Figure 2 on the balance feature resulting in a total of 125 perturbed test cases, all created by orange-level perturbation options. The first row in the table represents the original test case which the model predicted with “no”, i.e., the customer will probably not subscribe to a long-term deposit according to the logistic regression model given the feature values. 84 perturbed cases returned the same prediction as the original case, 41 returned a changed prediction and require further examination.

The first six perturbed test cases shown in Figure 6 were generated by the perturbation option shown in Figure 2. Adding and subtracting a small number to the balance does not change the original prediction and is, therefore, no problem for reliability. The next two perturbed cases were generated by changing the marital feature with the other two available values, “married” and “divorced”. This perturbation does also not change the prediction and is, therefore, no cause for concern regarding the reliability. The next shown perturbed case changes the feature if a customer has currently a credit in default, with the allowed values “yes” and “no”. Applying this perturbation option changes the original prediction from “no”, the customer will not subscribe to a long-term deposit, to “yes”, the customer will subscribe to a long-term deposit. This means that if our potential new customer had any credit in default, this would change the prediction. This small change of input values, which could be the actual case considering that the potential customer could have had a loan in default at another bank without the organization knowing, means that the analyst should probably not blindly trust the prediction of the model. The analyst should consider contacting the customer despite the model having returned a “no” prediction. The perturbation mode further calculates all combinations of perturbation options and receives predictions for all the thus created perturbed test cases for further analysis.

The last two perturbed test cases in Figure 6 show the beginning of the combination of the default feature perturbation option with the 10 percent perturbation on the balance feature, which also leads to a prediction change.

age	job	marital	education	default	balance	housing	...	prediction
20	technician	single	secondary	no	2143.00	yes	...	no
20	technician	single	secondary	no	2164.43	yes	...	no
20	technician	single	secondary	no	2121.57	yes	...	no
20	technician	single	secondary	no	2185.86	yes	...	no
20	technician	single	secondary	no	2100.14	yes	...	no
20	technician	single	secondary	no	2207.29	yes	...	no
20	technician	single	secondary	no	2078.71	yes	...	no
...	...	...	...	...	...	...	...	...
20	technician	married	secondary	no	2143,00	yes	...	no
20	technician	divorced	secondary	no	2143,00	yes	...	no
...	...	...	...	...	...	...	...	...
20	technician	single	secondary	yes	2143,00	yes	...	yes
20	technician	single	secondary	yes	2164.43	yes	...	yes
20	technician	single	secondary	yes	2121.57	yes	...	yes
...	...	...	...	...	...	...	...	...

Figure 6: Perturbed test cases for the predictive model over the banking dataset.

The examples shown in Figure 6 were created by orange-level perturbation options. These perturbation options may change the prediction of a created perturbed test case due to reaching a label border in the feature space. A red-level perturbation option should not change the prediction in any case, since the real-world value, perturbed with this option, is anywhere within the perturbed range. An example is a value measured by a sensor with a given interval for the sensor accuracy; the real value lies within the margins of the sensor accuracy. If a red-level perturbation option changes a prediction, the prediction is not reliable.

The second approach, i.e., using local quality measures, was also applied to the test case shown in the example in Figure 6. We calculated the local accuracy for the test case in order to judge the prediction. The overall accuracy of the model is about 78%. The local accuracy is calculated based on the 1500 nearest training set neighbors, as measured using Euclidean distance, and has a value of 93%. That value means that the subspace of the model that is considered has a better performance than the overall performance of the used classifier. Since the overall performance of the model is sufficient for the analysis, there is little doubt about reliability from the point of view of this approach. Having local areas which have a better accuracy than the global accuracy means, in turn, that there are also areas and, consequently, test cases where the local accuracy is worse, constituting a potential problem for reliability.

7 SUMMARY AND FUTURE WORK

In this paper, we introduced a reference process for judging the reliability of classification results over structured data. We used the CRISP-DM and explained which tasks need to additionally be performed in each of the six stages of CRISP-DM in order to arrive at predictive analytics applications that allow for assessing the reliability of individual predictions. Different data sources and their preparation have different reliability-related information associated, which can be used in subsequent stages to configure reliability-checking approaches. After the deployment of the model, these approaches can be used to judge the reliability of the predictions. The described tasks in the data understanding, data preparation, and modeling stages must be performed once for every use case. Gathered information and defined reliability-checking approaches are then examined if they are appropriate to judge the reliability for the use case during the evaluation phase. Once receiving predictions for new test cases, analysts can use the previously specified approaches to judge the reliability of individual predictions. The judgment is performed for each individual case. Obtaining additional information about the reliability of individual predictions requires additional effort but improved reliability will benefit decision-makers in critical business decisions. In this paper, we described the reference process using a classification example over structured data, but we will investigate applications of the reference process with other machine-learning methods, e.g., regression or clustering, and different sorts of input data, e.g., images, videos, and natural-language text, in future work. A knowledge graph may serve for the documentation of the knowledge regarding the proper selection of the approach for reliability checking.

REFERENCES

Barr, E. T., Harman, M., McMinn, P., Shahbaz, M., and Yoo, S. (2015). The Oracle Problem in Software Testing: A Survey. *IEEE Trans. Software Eng.*, 41(5):507–525.

Caetano, N., Cortez, P., and Laureano, R. M. S. (2014). Using Data Mining for Prediction of Hospital Length of Stay: An Application of the CRISP-DM Methodology. In Cordeiro, J., Hammoudi, S., Maciaszek, L. A., Camp, O., and Filipe, J., editors, *Enterprise Information Systems - 16th International Conference, ICEIS 2014, Lisbon, Portugal, April 27-30, 2014, Revised Selected Papers*, volume 227 of *Lecture Notes in Business Information Processing*, pages 149–166. Springer.

Capra, A., Castorina, A., Corchs, S., Gasparini, F., and Schettini, R. (2006). Dynamic Range Optimization by Local Contrast Correction and Histogram Image Analysis. In *2006 Digest of Technical Papers International Conference on Consumer Electronics*, pages 309–310, Las Vegas, NV, USA. IEEE.

Chen, T. Y., Kuo, F.-C., Liu, H., Poon, P.-L., Towey, D., Tse, T. H., and Zhou, Z. Q. (2018). Metamorphic Testing: A Review of Challenges and Opportunities. *ACM Comput. Surv.*, 51(1):4:1–4:27.

Cheney, E. W. and Kincaid, D. R. (2012). *Numerical mathematics and computing*. Cengage Learning.

Cruz, R. M. O., Sabourin, R., and Cavalcanti, G. D. C. (2018). Dynamic classifier selection: Recent advances and perspectives. *Inf. Fusion*, 41:195–216.

da Rocha, B. C. and de Sousa Junior, R. T. (2010). Identifying bank frauds using CRISP-DM and decision trees. *International Journal of Computer Science and Information Technology*, 2(5):162–169.

Didaci, L., Giacinto, G., Roli, F., and Marcialis, G. L. (2005). A study on the performances of dynamic classifier selection based on local accuracy estimation. *Pattern Recognit.*, 38(11):2188–2191.

Guo, H., Shi, W., and Deng, Y. (2006). Evaluating sensor reliability in classification problems based on evidence theory. *IEEE Trans. Syst. Man Cybern. Part B*, 36(5):970–981.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS quarterly*, pages 75–105.

Moreira, D., Furtado, A. P., and Nogueira, S. C. (2020). Testing acoustic scene classifiers using Metamorphic Relations. In *IEEE International Conference On Artificial Intelligence Testing, AITest 2020, Oxford, UK, August 3-6, 2020*, pages 47–54. IEEE.

Moro, S., Cortez, P., and Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decis. Support Syst.*, 62:22–31.

Moro, S., Laureano, R., and Cortez, P. (2011). Using data mining for bank direct marketing: An application of the crisp-dm methodology. Publisher: EUROSIS-ETI.

Moroney, N. (2000). Local Color Correction Using Non-Linear Masking. In *The Eighth Color Imaging Conference: Color Science and Engineering Systems, Technologies, Applications, CIC 2000, Scottsdale, Arizona, USA, November 7-10, 2000*, pages 108–111. IS&T - The Society for Imaging Science and Technology.

Reimer, U., Tödtli, B., and Maier, E. (2020). How to Induce Trust in Medical AI Systems. In Grossmann, G. and Ram, S., editors, *Advances in Conceptual Modeling - ER 2020 Workshops CMAI, CMLS, CMOMM4FAIR, CoMoNoS, EmpER, Vienna, Austria, November 3-6, 2020, Proceedings*, volume 12584 of *Lecture Notes in Computer Science*, pages 5–14. Springer.

Saha, P. and Kanewala, U. (2019). Fault Detection Effectiveness of Metamorphic Relations Developed for Testing Supervised Classifiers. In *IEEE International Conference On Artificial Intelligence Testing, AITest 2019, Newark, CA, USA, April 4-9, 2019*, pages 157–164. IEEE.

Segura, S., Fraser, G., Sánchez, A. B., and Cortés, A. R. (2016). A Survey on Metamorphic Testing. *IEEE Trans. Software Eng.*, 42(9):805–824.

Vriesmann, L. M., Jr, A. S. B., Oliveira, L. S., Koerich, A. L., and Sabourin, R. (2015). Combining overall and local class accuracies in an oracle-based method for dynamic ensemble selection. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015*, pages 1–7. IEEE.

Wirth, R. and Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for Data Mining. page 11.

Woods, K. S., Kegelmeyer, W. P., and Bowyer, K. W. (1997). Combination of Multiple Classifiers Using Local Accuracy Estimates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4):405–410.

Xie, X., Ho, J. W. K., Murphy, C., Kaiser, G. E., Xu, B., and Chen, T. Y. (2011). Testing and validating machine learning classifiers by metamorphic testing. *J. Syst. Softw.*, 84(4):544–558.