DNV·GL

RECOMMENDED PRACTICE

DNVGL-RP-0510 Edition April 2020

Framework for assurance of data-driven algorithms and models

DNV GL AS

FOREWORD

DNV GL recommended practices contain sound engineering practice and guidance.

Any comments may be sent by e-mail to *rules@dnvgl.com*

CHANGES – CURRENT

This is a new document.

CONTENTS

SECTION 1 GENERAL

1.1 Introduction

Uses of artificial intelligence (AI), machine learning (ML) and other data-driven techniques have become increasingly widespread in recent years. Many organizations now seek to capitalize on the potential such techniques offer to do things better, do things faster, and/or do things that were previously impossible.

In industrial contexts, data-driven techniques are being used in a variety of applications including:

— early detection of machinery failures (before they happen) and condition-based maintenance (CBM)
— semi- and fully-automated technical verification of documents
— prediction of unwanted events (accidents or other safety incidents)
— automatic classification of text-based maintenance logs and inspection findings
— detection of cracks or other defects.

For the purposes of this RP, a data-driven model is a computational unit / program / function which makes predictions, and whose configuration is determined by a training operation on data. As shown in Figure 1-1 a data-driven application contains one or more data-driven models, and uses the predictions of the model for some business purpose.

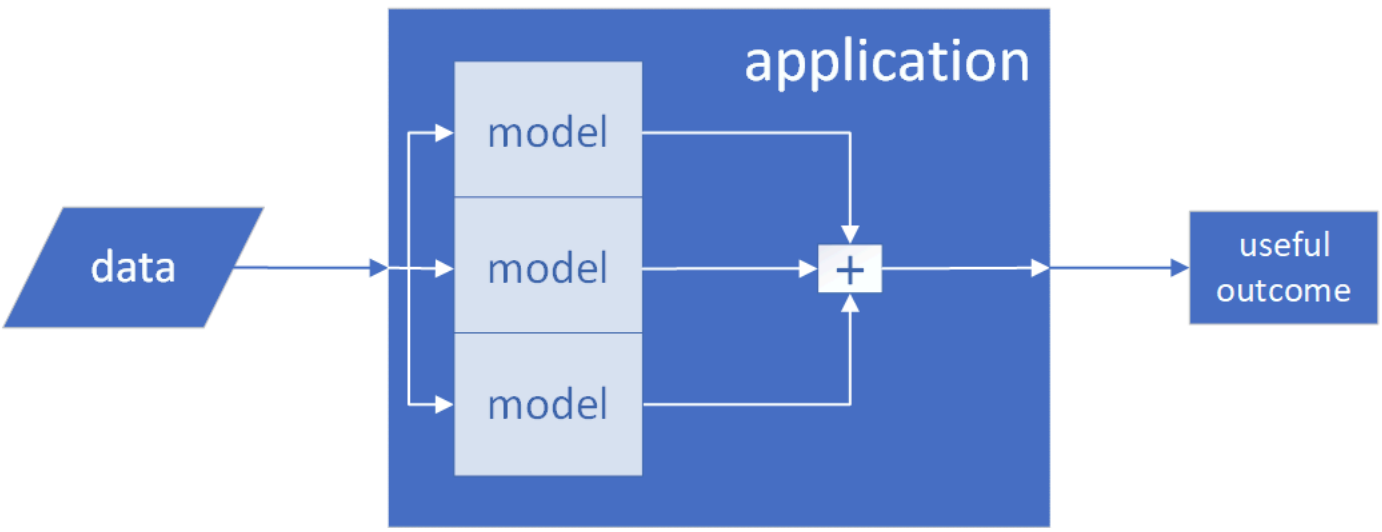

Figure 1-1 Relation between model and application.

The amount of responsibility humans are willing to hand over to any data-driven application depends on the criticality of the task it will perform, and the level of trust they have in the application. An application which suggests purchases for a consumer based on their previous shopping history has a lower criticality than an application which steers an autonomous vehicle. Most data-driven applications in use today have low criticality: their use is restricted either to low consequence scenarios, e.g. suggesting purchases, or to scenarios in which the application provides decision support for a human end user. But there is growing interest see e.g. /9/ from both vendors, consumers, industry and regulatory bodies in widening the scope in which data-driven applications can be used, to perform tasks with higher criticality and/or to move from decision support to decision taking.

However, difficulties remain in establishing trust that a data-driven application will operate as required, safely and reliably. The complexity of the data and the training algorithms, coupled with the lack of any standard approach to establishing trust in such applications, lead many to take a conservative approach and simply refuse to adopt such technologies until the field has matured. DNV GL considers this a missed opportunity, and maintains that it is possible to enable trust in data-driven applications through a systematic and data science-oriented consideration of risk.

All the stakeholders in a data-driven application need trust:

— The owner/investor wants confirmation that the application is trustworthy so as to limit exposure to liability, and guarantee a return of investment (ROI).
— Regulatory bodies require that the application is trustworthy before it is approved for use and allowed on the market.
— The end user demands that the application is trustworthy and safe before they will use it.

To date no widely-recognized standard exists for assessment / assurance of data-driven applications. This RP aims to fill that gap.

1.2 Objective

This RP defines a framework of claims. This RP defines a framework which can be used to systematically establish that an application including one or more data-driven models is suitable for its intended use, and that the risk of failure and/or unsuitable output is within tolerable limits.

1.3 Scope

This RP is organized as a set of claims organized according to the typical activities in application development and operation. For a specific data-driven application, the RP may be used to compile an assurance case. For each claim it is expected that arguments and/or evidence are provided showing how that claim is fulfilled. Claims and evidence together comprise the completed assurance case. For an introduction to assurance cases see /23/.

1.4 Application of this document

1.4.1 General

It is envisaged that the framework described in this RP can be used in three distinct yet complementary ways:

1) To codify best-practice by providing a standardized approach to risk assessment for data-driven applications, making it easy to communicate the risk picture of an application to all stakeholders.
2) To structure self-assessment of the reader's own data-driven application to help ensure that the pitfalls common to data-driven projects are avoided.
3) To construct a formal assurance case /23/ for a particular application, which can be used as input to:
    1) independent assurance of the application (assurance is the process of verifying that prescribed evidence is supplied), and/or
    2) independent risk assessment of usage of the application, assessing risk associated with the intended use of the application.

The appendices describe several concrete scenarios in which the RP is applicable:

— as a framework for performing:
    — risk assessment of a data-driven application , see [A.1]
    — risk assessment of a hybrid application containing both data-driven components and physics-based or rule-based components , see [A.2]
    — direct performance/risk assessment of a data-driven application , see App.B
— for assurance of data-driven applications used for predictive maintenance , see App.C.

Although the claims in this RP are organized according to the steps defined in the de facto industry standard development model CRISP-DM, the RP is also applicable to applications comprising data-driven models developed using other processes.

1.4.2 Limitations

The main limitations of applicability of this RP are:

— This RP is applicable to assurance of a wide range of data-driven applications, but it does not offer comprehensive cover for applications for use in highly critical processes (i.e. involving high safety risk, high environmental risk or high economical risk). However, the RP does refer to criticality in numerous places: this is primarily to identify which areas to focus on if the RP is used in applications with somewhat high criticality.
— This RP is applicable to supervised learning applications (i.e. the data from which the application's model is trained contains both features (or independent variables) and a target (or dependent variable or label), and the training task is to learn the mapping from features to target such that at prediction time the model can be supplied with features and can predict the target). Applications comprising unsupervised and/or reinforcement-learning models are not currently included.
— In-depth analysis of ethical and societal implications, concerns for fairness, privacy or confidentiality are not comprehensively supported. The *EU Ethics guidelines for trustworthy AI* (European Commission, 2019) /8/ may be a suitable resource if the application of interest has a significant ethical / fairness aspect.
— This RP does not contain guidance in the related fields of data quality and data management. See DNVGL-RP-0497 for detailed guidance.

1.4.3 Audience

The anticipated audience for this RP includes:

— enterprises who want to use data-driven applications in a risk-aware way
— producers and vendors (e.g. data scientists) who want to convince their customers that a data-driven application is safe/reliable
— quality managers who want to document compliance and risk level of an application
— other stakeholders in projects that develop / utilize data-driven applications, who want to understand which aspects of the system contain risk
— third-party organizations (such as DNV GL) seeking to provide independent assurance of data-driven applications
— regulatory bodies who want to understand the risk of a data-driven application.

1.5 References

Table 1-1 DNV GL documents

Document code	*Title*
DNVGL-CP-0484	Approval of service supplier scheme
DNVGL-RP-0497	Data quality assessment framework
DNVGL-RU-OU-0300	Fleet in service

Table 1-2 External documents

Reference no.	*Source*
/1/	CRISP-DM 1.0 Step-by-step data mining guide. SPSS. Version 1.0. 1999
/2/	CAP 670 ATS Safety Requirements. CAA. Version 3. 2014
/3/	Moore, Geoffrey A. Crossing the Chasm. Harpers Collins. pp. Ch. 6. 1991
/4/	Project Management https://en.wikipedia.org/wiki/Project_management
/5/	ASUM: Analytics Solutions Unified Method. IBM Analytics Services. See ftp://ftp.software.ibm.com/software/data/sw-library/services/ASUM.pdf
/6/	Agile software development, https://en.wikipedia.org/wiki/Agile_software_development
/7/	REASON NOR-STA support system for achieving and assessing conformance to NORms and STAndards www.nor-sta.eu
/8/	EU Ethics Guidelines for trustworthy AI https://ec.europa.eu/futurium/en/ai-alliance-consultation, April 2019
/9/	Bloomfield, R.; Khlaaf, H.; Ryan Conmy, P.; Fletcher, G. Disruptive Innovations and Disruptive Assurance: Assuring Machine Learning and Autonomy https://ieeexplore.ieee.org/document/8812789 Computer, Vol 52, Issue 9, pp82-89, 2019
/10/	ISO 5725-1:1994 Accuracy (trueness and precision) of measurement methods and results - Part 1: General principles and definitions, 1994
/11/	Muller, A. and Guido, S. Introduction to Machine Learning with Python: A Guide for Data Scientist. O'Reilly, 2016
/12/	LIME https://lime.readthedocs.io/en/latest/
/13/	SHAP https://shap.readthedocs.io/en/latest/
/14/	Molnar, C. Interpretable Machine Learning https://christophm.github.io/interpretable-ml-book/, 2017
/15/	EU Regulation 2016/679 (GDPR) https://eur-lex.europa.eu/eli/reg/2016/679/oj, 2016
/16/	McKinney, W. Python for data analysis, O'Reilly, 2013
/17/	ISO 13379-1:2018 Condition monitoring and diagnostics of machines - Data interpretation and diagnostics techniques - Part 1: General guidelines, 2018
/18/	ISO 8000-8:2015 Data quality - Part 8: Information and data quality: Concepts and measuring, 2015
/19/	Gebru, T.; Morgenstern, J.; Vecchione; B., Vaughan, J.; Wallach, H., Daume, H.; Crawford, K. Datasheets for datasets, https://arxiv.org/abs/1803.09010, 2020
/20/	Chollet, F. Deep learning with Python, Manning 2017
/21/	ISO 8000-120:2016 Master data: Exchange of characteristic data: Provenance, 2016
/22/	W3C Dataset Exchange Use Cases and Requirements, https://www.w3.org/TR/dcat-ucr/, 2019
/23/	NASA/CR-2015-218802 Understanding and evaluating assurance cases, https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20160000772.pdf, 2015
/24/	IEC 60300-3-11 Dependability management - Part 3-11: Application guide - Reliability centred maintenance, 2009

Reference no.	*Source*
/25/	BS EN 13306:2017 Maintenance - maintenance terminology, 2017
/26/	ISO 8000-2:2017 Data quality - Part 2: Vocabulary, 2017
/27/	ISO 13372:2012 Condition monitoring and diagnostics of machines - Vocabulary, 2012
/28/	IEC 60812:2018 Failure modes and effects analysis (FMEA and FMECA), 2018

1.6 Definitions and abbreviations

Table 1-3 Definition of verbal forms

Term	*Definition*
shall	verbal form used to indicate requirements strictly to be followed in order to conform to the document
should	verbal form used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others
may	verbal form used to indicate a course of action permissible within the limits of the document

Table 1-4 Definition of terms

Term	*Definition*
accuracy	for measurements, the combination of trueness and precision, see /10/. See also classifier accuracy
algorithm	generic term for a set of instructions, typically implemented as computer code, which perform a specific task. The code which takes training data and produces a model is itself an algorithm, but in this RP algorithm is used a synonym for model
application	piece of software intended to be used for a specific, defined, purpose. See also data-driven application
assurance case	structured collection of claims with supporting evidence, used as the basis of assurance. For an introduction to assurance cases, see /23/
analytics solutions unified method	IBM Analytics Services' adaptation of CRISP-DM, see /5/
baseline	A simple, understandable reference model / value used to check that the developed model is providing a performance improvement
claim	statement pertaining to an application, which shall be evaluated to be true for the application to be assured. Claims are supported by evidence
classifier accuracy	for classifiers, the ratio of the total number of true positives and true negatives to the total number of evaluation samples. In machine learning terminology this is usually referred to simply as accuracy. In this RP classifier accuracy is used to avoid confusion with /10/. See also accuracy
classifier precision	for classifiers, the ratio of the number of true positives to the sum of true positives and false positives. In machine learning terminology this is usually referred to simply as precision. In this RP classifier precision is used to avoid confusion with /10/. See also precision
confidence	figure of merit used in class-based verification, see App.C

Term	*Definition*
criticality	relative measure of the quality, state, or degree of being of the highest importance, e.g. an application is critical if its inadequate output will lead to unacceptable consequences, and data inputs are critical if inadequate data will lead to such inadequate outputs. Note: this definition does not include probability (in contrast to the definition of criticality used in FMECA). See risk to understand how probability is incorporated
cross-industry standard process for data mining	open standard process developed through a European Union funded project, see also /1/
data-driven application	application which includes at least one data-driven model. The parameters and configuration of a data-driven model are automatically determined (or learned) from data using a suitable algorithm.
data-driven model	model which is made by applying a suitable training algorithm to a set of data
data mining	term used in the original CRISP-DM procedure. In this RP the more current term data-driven modelling is used
evidence	documentation or other proof supporting a claim. Risk assessment comprises the evaluation of evidence
failure mode	ways or modes in which a component or system may fail
false negatives	test data points with a true value of positive for which a classifier predicts negative
false positives	test data points with a true value of negative for which a classifier predicts positive
infrastructure	the set of project-independent systems which together support application development. Includes project management, change management, version control, software development tools, issue tracking and operational status monitoring
machine learning	sub-field of AI concerned with performing a specific task without using explicit instructions
model	computer-based representation of some process and/or entity, which is typically used to make predictions and/or other useful transformations of input data
model hyperparameter	parameter which specifies an aspect of model training. Hyperparameters are typically set manually.
model parameter	parameter which forms part of a trained model. Model parameters are typically determined automatically, during training.
precision	for measurements, see /10/, a measure of the consistency of repeated measurements. See also classifier precision
reason	tool supporting assurance case reasoning, see also /6/
recall	for classifiers, the ratio of the number of true positives to the sum of true positives and false negatives
risk	any situation involving exposure to unacceptable consequences from the intended use of an application. For the purposes of risk assessment, the risk of an event is defined as the event probability multiplied by the severity of the event, where both factors can be expressed in either qualitative or a quantitative terms
true negatives	test data points with a true value of negative for which a classifier predicts negative
true positives	test data points with a true value of positive for which a classifier predicts positive

Term	Definition
unacceptable consequence	adverse outcome resulting from the intended use of an application, related to e.g. safety, security, confidentiality, privacy, ethics, fairness, societal concerns, protecting the environment, and customer needs for business performance, effectivity and efficiency
value proposition	a statement describing the key properties of a product or application including what it is, who it is for, what it does, and how it is different to alternatives, see also /3/
version control	function provided by a suitable software tool facilitating management of change to documents and/or computer programs

Abbreviations used in the RP are defined in Table 1-5.

Table 1-5 Abbreviations

Abbreviation	*Description*
AI	artificial intelligence
ASUM	analytics solutions unified method /5/
CBM	condition-based maintenance
CM	condition monitoring
CMMS	computerized maintenance management system
CRISP-DM	cross-industry standard process for data mining /1/
DM	data mining
ETTF	estimated time to failure
FMEA	failure mode and effects analysis
FMECA	failure mode, effects, and criticality analysis
FMSA	failure mode and symptoms analysis
ML	machine learning
NPV	net present value
RCM	reliability-centered maintenance /24/
RP	recommended practice
RUL	remaining useful life

1.7 Procedural requirements

1.7.1 Documentation requirements

To perform assurance of a specific application each claim defined in this RP should be supported by documentary evidence. The RP defines the framework, it does not make requirements for which documentation is required or what level of detail is required: such requirements shall be provided by the party, such as regulatory body, customer, end user, using the framework.

The process of gathering, maintaining and assessing documentary evidence may be facilitated by using a computerized assurance case tool.

1.7.2 Certification requirements

No certification schemes for data-driven applications currently exist.

1.7.3 Recommended documentation

It is recommended that the following documentation items are collected to support the assurance case.

- project plan
- risk analysis
- system architecture
- development and runtime environment
- data design, format specifications, metadata, semantic descriptions
- service level agreements for related systems and services
- data quality service level agreement, if any, for input data
- data management relevant for the data in scope
- IT security requirements and limitations
- application design
- test plan
- test report (may be compiled automatically)
- operation environment and plan, including:
  - the users and their context of use
  - acceptance criteria and success criteria
  - processes for all relevant phases and iterations during the lifetime of the application.

SECTION 2 FRAMEWORK

2.1 Assurance case

The RP defines a standardized assurance case /23/ for assembling and assessing claims and evidence, as shown in Figure 2-1. For any specific application of interest an assurance case may be compiled by populating the claims (specified in this RP) with application-specific evidence (documentation). As shown in Figure 2-2 the complete assurance case then forms the basis of subsequent assessment.

Figure 2-1 An assurance case is comprised of claims and supporting evidence.

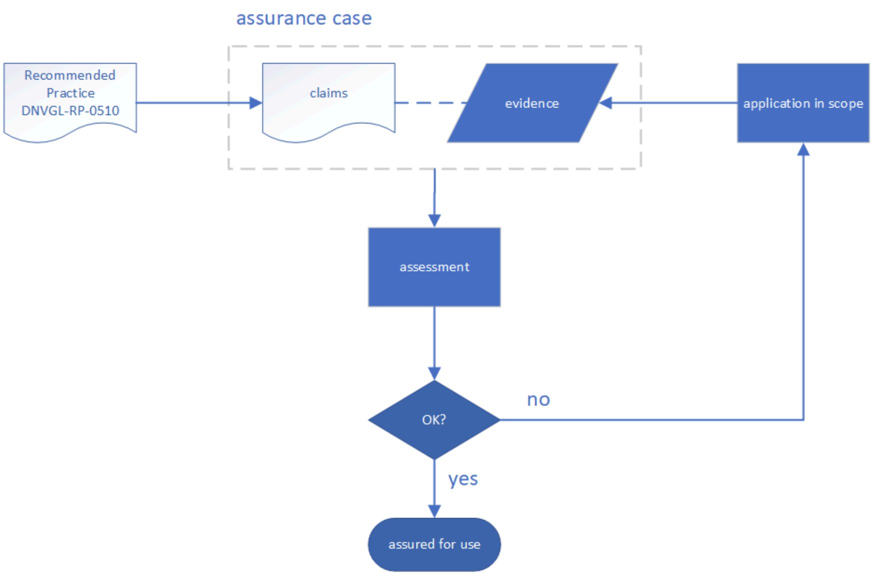

Figure 2-2 This RP defines a set of claims, which together with evidence for the application of interest, constitute an assurance case.

The assurance case defined in this RP is organized according to the steps in development, evaluation and deployment commonly used in industry. These coincide with the steps defined in the cross-industry standard process for data mining (CRISP-DM) model /1/, more recently extended by IBM to ensure the consistency and efficiency of analytical processes /5/. The RP is, however, equally applicable to applications in which the data-driven component is developed using other processes.

There are several ways in which an assurance case can be structured, but all typically utilize a complete basis, a description of the application which sufficiently covers the relevant risks. For example, some risk assessment approaches use component lists as the basis for analysis, assuming that application risk can be correctly identified starting from a complete list of sufficiently simple components. As mentioned above, in this RP the assurance case is structured according to the steps in the CRISP-DM process. DNV GL considers that this deconstruction of the development process, the life-cycle and the operational context of the application represents a complete basis, sufficient to capture the risk picture of the application and its intended use. This approach assumes of course that the process deconstruction is appropriate and complete i.e. has no missing steps or unsuitable sequences of steps.

A separate effort to ensure that the RP is complete and fulfils its main objective has been to cross-check the assurance case defined here against the principles of the assurance case defined by standard CAP670, SW 01: *Regulatory Objectives for Software Safety Assurance* in ATS Equipment /2/. The CAP670 assurance case is less focussed on the details of the applied development process and is instead focused on assessing the validity of requirements and verifying their implementation, as well as proper configuration control. This different perspective makes CAP670 particularly suitable for cross-checking the assurance case defined in the current RP. It is considered that the goal of constructing an assurance case is the correct identification and assessment of risk in the end product used as intended and the mitigation of that risk to acceptable levels. If this risk could be measured directly on the end product, such evidence could be directly used as assurance (which is why CAP670 calls that direct evidence). Unfortunately, direct evidence is only conclusive on very simple systems, and since data-driven applications are complex, direct evidence may not be conclusive.

DNV GL recognizes that some applications are more critical, business- and safety-wise, than others. DNV GL recommends that the level of evidence provided reflects the criticality of the application: applications with high criticality require more evidence than applications with low criticality. This increasing rigour is referred to in some of the notes related to the claims.

2.2 Structure of framework

As described in [2.1], an assurance case comprises claims and evidence. The claims are listed in Sec.3 and Sec.4. DNV GL believes this suite of claims constitutes a complete basis.

This RP defines a framework for organizing claims and evidence to assess the risk associated with using a data-driven application for a particular use. The framework distinguishes between two distinct types of claims, as shown in Figure 2-3:

1) claims about the capabilities of the organization and development environment in which the data-driven application is developed, maintained and operated, see Sec.3
2) claims about the specific steps used in the development, deployment and operation of the application, see Sec.4

Requirements on the performance and failure behaviour of the application are established as part of the process, and the application's conformance to these requirements is assured as part of Sec.4.

[2.3] includes a list of recommended documentation to support the assurance case.

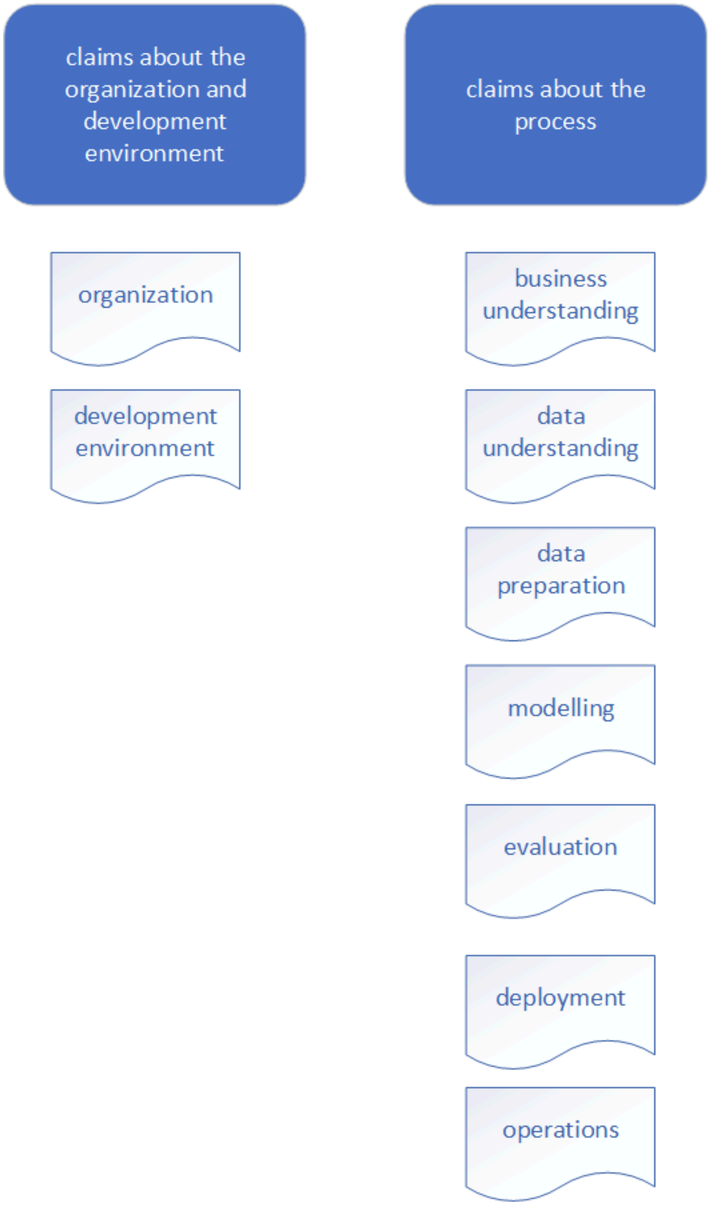

Figure 2-3 The framework includes claims about the organization and development environment, and claims about the process.

2.3 Structure of claim

Each claim is presented as a simple statement, followed by a short explanation. Guidance notes are given to suggest what type of evidence is appropriate. It is not possible to give an exhaustive description of evidence required. This is because i) the scope of assurance of data-driven applications is wide, and this framework is designed to be used for assurance of both simple and complex applications; and ii) the field is evolving rapidly.

SECTION 3 CLAIMS RELATED TO THE ORGANIZATION AND DEVELOPMENT ENVIRONMENT

3.1 General

The claims in this section relate to the infrastructure and development / operation environment.

3.2 The organization

3.2.1 Competencies and organizational maturity

3.2.1.1 Claim: the organization supporting the project has sufficient maturity in carrying out data science.

To get good results from data-driven applications, it is important that the organization has sufficient maturity in executing data science projects. Doing good data science requires a suitable mix of computing expertise, data engineering, statistical knowhow and subject-matter/domain knowledge. Running effective data science projects requires an organization which has the necessary infrastructure, tools and work processes to support data science.

3.2.1.2 Claim: the team assigned to the project has sufficient resources, time, and competence in the use of the relevant methods tools and platforms to successfully execute the project.

Guidance note:

It may be necessary to enlist the competence of other stakeholders and participants outside the developing organization to ensure the quality of the end product. Understanding competence needs is especially important early in the project when manning is being planned. Verifying that the identified competence needs are met is an important step at all stages of the project. It may be necessary to re-evaluate competence requirements if new challenges arise.

Evidence to support this claim may include competence matrices and formal competence requirements for the various activities.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

3.3 The development environment

3.3.1 Configuration management

3.3.1.1 In a data-driven modelling project it is essential that up-to-date requirements and acceptance criteria are used in all phases of the development cycle. To ensure that these items are up-to-date, sufficient change management / version control of key documents is required. It is important that the requirements and acceptance criteria are explicitly linked from the documentation to the requirements implementation and the evaluation of acceptance criteria in the data and modelling design. This linking makes it easy to demonstrate that all requirements and acceptance criteria have been properly satisfied and in which version they have been satisfied.

3.3.1.2 Claim: all documentation used as evidence for the assurance case is managed under sufficient version control and is clearly linked to the items (with version numbers) to which it relates.

3.3.1.3 Claim: the requirements, including objectives, success and acceptance criteria, are managed under version control and clearly linked to the data design, data sets, models and modelling design.

3.3.1.4 Claim: data design and data sets and integrations are managed under sufficient version control and clearly linked to the requirements and the models.

Guidance note:

Guidance on change management in a data management context can be found in DNVGL-RP-0497. Suggestions for templates for data set specification are given in [4.3.2.1].

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

3.3.1.5 Claim: features are managed under sufficient version control and clearly linked to requirements, data design and data sets.

3.3.1.6 Claim: model design and the model itself are managed under sufficient version control and clearly linked to requirements, data design, data sets and features.

3.3.1.7 Claim: data sets used for test and monitoring are managed under sufficient version control and clearly linked to the model and data design.

Guidance note:

See guidance note in [3.3.1.4].

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

3.3.2 Multiple iterations

3.3.2.1 As described in [1.1] an application is composed of one or more models. In practice both application and model(s) are often developed according to iterative processes: iterative / agile development has shown to be effective in many branches of IT (see /6/ and references therein) and in data science CRISP-DM /1/ and ASUM /5/ are widely used. For data-driven applications, there are two scales at which iterative development occurs, the application level and the model level:

1) At the application level there is process cycle iteration. This iterative cycle includes application development, application deployment and operations and may include error handling, redundancy and other fall-back mechanisms. For a project to show maturity, evidence shall be provided that the application requirements are complete and satisfactorily met, sufficiently free of risk and unacceptable consequences. The claims in the current subsection are related to process cycle iteration.
2) At the model level, development includes iteration over business understanding, data understanding, data preparation, modelling, evaluation and deployment. During each iteration a model may be evaluated against user requirements/acceptance criteria. If these requirements are not met, some or all of the development steps are revisited. Claims relating to model iteration are covered later in Sec.4.

3.3.2.2 Claim: the application development process has been iterated until sufficient confidence is obtained that the requirements and acceptance criteria are fulfilled.

Guidance note:

Evidence to support this claim may include performance requirements and corresponding test results, showing that the performance of the application meets the requirements.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

3.3.2.3 Claim: the application development process has been iterated until sufficient evidence is obtained that the application, operating in its intended environment, satisfies the needs and expectations with a tolerable risk level.

Where [3.3.2.2] is concerned with performance, [3.3.2.3] is concerned with risk: together these two claims address the primary objective of the assurance process, as defined in [1.2].

Guidance note:

Evidence to support this claim may include documentation of application failure modes and risk assessment for the intended use of the application, demonstrating that the risk associated with use of the application is tolerable.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

3.3.3 Infrastructure and coding

3.3.3.1 It shall be shown that the tools and systems used for data access, manipulation, modelling, evaluation and deployment are of sufficient quality to meet the business objectives without introducing risk. While the popularity of third-party open-source libraries has accelerated adoption of data-driven models, it should not be assumed that such libraries are bug-free and/or secure.

3.3.3.2 Claim: the infrastructure used to support application development and operation, including its change management procedure, is sufficiently documented.

Guidance note:

Evidence for this claim may include a data flow diagram and/or workflow diagram covering all phases from data preparation through model development to application deployment and operation.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

3.3.3.3 Claim: the coding of the application, its model(s) and relevant tools has been done following established software development practices.

Guidance note:

Evidence to support this claim may include documentation of the tools / languages used, information about how code quality is confirmed (such as the use of code style conformance tools), unit test reports and test coverage reports and integration test results.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

3.3.3.4 Claim: all data used in development and operation of the application is managed by a data management regime meeting the short- and long-term needs.

Guidance note:

Evidence to support this claim may include documentation of a sufficient data management maturity level, contracts giving access to needed data, cost control on external data, IPR rights to utilize data for the given purpose and change management in place for distributed data sources consumed. DNVGL-RP-0497 describes a framework for managing data quality, which includes data management maturity assessment.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

SECTION 4 CLAIMS ABOUT THE PROCESS

4.1 Introduction

The claims in this section relate to the development, deployment and operations process and are organized according to the steps of the CRISP-DM workflow process (see /1/), but the claims are applicable to other workflows too.

4.2 Business understanding

4.2.1 General

A critical factor in the development of data-driven applications is the understanding of the business case: the primary reason why the application is being built. A well-executed process, which includes input from business/domain experts at an early stage, directs the process towards applications that properly address the business objectives and takes the business environment into account, ensuring the desired results and avoiding unacceptable consequences post deployment. This includes understanding the business needs, helping provide the application with user-driven goals, and the monetary, safety, regulatory and societal constraints that the application will face in the real world when deployed.

Business understanding should result in well-defined requirements and evaluation criteria which can later be used to evaluate how well the application is aligned with the intended purpose.

4.2.2 Determine business objectives

4.2.2.1 Sufficient understanding of the business case, and well-defined objectives and success criteria are important for a successful project. It may be helpful to formulate a use case, user stories and/or a value proposition (see, for example, /3/). This directs attention towards the project goal and facilitates communication between the stakeholders.

Early in the development process it may not be possible to explicitly define what will be predicted, but completing a value proposition statement as early as possible in the project process will guide the data preparation stage.

A value proposition should address the following issues:

— Who is the intended user of the application?
— How will they use it?
— What will they achieve through using it?
— How is that beneficial?

An alternative or supplement to a value proposition is a collection of use cases which give concrete examples of the use of the application in practice. These bring focus to discussions concerning the business understanding and are useful aids in communication between the different participants and stakeholders in the development process.

4.2.2.2 Claim: business context is sufficiently documented.

Guidance note:

Evidence to support this claim may include a value proposition, description of use cases, or other documentation which make it clear where the application is shall used, by whom, and for which purpose.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.2.3 Claim: business objectives are clearly defined and understood.

Guidance note:

Evidence to support this claim may include a statement of the purpose of the application. This may be the same evidence as for claim [4.2.2.2].

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.2.4 Claim: business success criteria are sufficiently and objectively defined.

The business success criteria play a central role in determining whether to use / deploy / operate the application and whether further development is necessary / warranted. If the criteria are properly defined then the application's success can be assessed objectively with little room for doubt or interpretation.

Guidance note:

Evidence to support this claim may include measurable requirements on application performance (measured in business terms, such as cost savings, time savings, number of unwanted events prevented, mean time to failure or other KPIs).

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.3 Assess situation

4.2.3.1 Assessing the situation is important for several reasons. Firstly, it lays the groundwork for a successful project according to plan and budget. Secondly, it involves analysis and understanding of both the intended and the unintended operation of the application. This assessment is crucial to managing risk in the deployed application.

This phase consists of identifying resources, assumptions and constraints, which in turn guide the data analysis goals and the project plan.

Goal-oriented development focuses on functionality used and performance obtained when the application operates according to expectations. Risk management, on the other hand, considers what happens in unexpected situations, or if some part of the application fails. This way of thinking is effective in identifying and managing risk, avoiding unacceptable consequences and, in more critical applications, ensuring safe operation.

It is recommended to complete [4.2.3] (assess situation) and [4.2.4] (determine data-driven modelling goals) in two iterations:

1) high level: consider claims [4.2.3.2]-[4.2.3.8] and how they relate to the modelling goals in [4.2.4]
2) low level: consider claims [4.2.3.9]-[4.2.3.13] with focus on detailed technical success and risk, and how they relate to the modelling goals in [4.2.4].

4.2.3.2 Claim: key stakeholders and resources for a successful development, deployment and operation are identified and assigned.

Guidance note:

There is a requirement of participation of key stakeholders and resources from the customer, from the developing organization, data providers, hosting and technology providers, and possibly also from additional sources providing domain knowledge or independence in assurance. It is important to secure the right competencies and project participants early, especially for the more critical steps of the project. Evidence to support this claim may include a competence map and CVs.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.3.3 Claim: available data resources are identified and a sufficient plan for accessing and handling the data is prepared.

Access to sufficient quality data and data management capabilities is crucial and shall be secured as early as possible. This consideration includes data both for training and development, and data available during operation / deployment.

Guidance note:

Evidence to support this claim may include a list of data sources and outline plan of how data will be collected both during development and post deployment. A suitable template for a data set datasheet is provided by /19/.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.3.4 Claim: project requirements are sufficiently defined.

The requirements should always be valid, complete and up-to-date. The requirements shall be maintained by suitably competent resources.

Guidance note:

It is important to identify project requirements, assumptions and constraints early to establish a good project plan. These may include deliverables, milestones, timelines, budget and costs, work processes, quality management, and reporting. Be aware that requirements, assumptions and constraints related to work processes can touch upon technical success, e.g. inadequate requirements for using specific processes, methods or tools can have a negative effect on the technical success and set unwanted constraints.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.3.5 Claim: project assumptions and constraints are sufficiently identified and understood.

The application should be sufficiently identified/defined to sufficiently describe who or what provides input, who or what will use the output, and how the output will be used. When an application is composed of sub-models, it is important to include also these details. Post-processing steps performed on the raw model output should also be included.

Guidance note:

Evidence to support this claim may include a list of key assumptions, and for each assumption documenting the basis for the assumption, the arguments supporting making the assumption, and the potential consequences if that assumption is not valid.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.3.6 Claim: project risks are sufficiently identified and mitigated as needed.

Guidance note:

It is best practice to identify the risks relevant to operational success and to compile a risk register and contingency or mitigation plan. This risk register should be updated throughout the project lifetime. The assumptions listed in [4.2.3.5] should be included in the risk register.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.3.7 Claim: a suitable terminology for the project is sufficiently defined and made available for use in the project.

A factor for success is to establish a common language and terminology in the project. It is recommended that this includes terms relevant for business understanding as well as for data understanding. Communication in data-driven modelling projects often involves terminology from the business domain, from data-driven modelling, data management and from computer science and other IT fields.

Guidance note:

Evidence to support this claim may include the glossary / terminology list itself.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.3.8 Claim: the application design is sufficiently defined.

Guidance note:

Evidence to support this claim may include the application design documents.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.3.9 Claim: the cost versus benefit has been sufficiently estimated as motivation for the project.

A solid motivational grounding through a cost versus benefit analysis should be established to substantiate if the benefits and outcomes of the project justifies the costs and efforts. It is recommended to be realistic in this analysis, not to over-sell or overestimate the benefits, nor to underestimate the cost and effort. A project can still be established even if the risk of failure is high, as long as all stakeholders are aware and accept this.

Guidance note:

Evidence to support this claim may include a cost versus benefit analysis document.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.3.10 Claim: failure modes are identified.

Data-driven applications have many failure mechanisms: failures can be caused by data quality issues, software, hardware, physical components, human error, or by malicious acts by adversarial agents taking advantage of security vulnerabilities.

An assessment of the criticality level of the application should be performed. Criticality relates to the severity of unacceptable consequences matched with the probability or frequency of the inadequate or failed application output leading to it. The criticality level should be set accordingly and used as a basis for determining the type / amount of evidence required. Critical failure modes and causes should be assessed, the risk associated with them quantified and action to mitigate risk taken. Some mitigating actions can lead to changes in system design, triggering new iterations of previous steps, but most risk requirements can be used in later phases as acceptance and evaluation criteria.

It is recommended to identify unacceptable consequences relevant to the application in this step. In more critical applications, these could be safety-related: accidents, material damages, hazards, and/or low performance. This RP is not targeted for safety-critical applications, but by doing this kind of exercise certain safety risks can at least be ruled out.

Failure modes may relate to ethics, privacy, confidentiality, security, and/or fairness. [4.2.3.11] and [4.2.3.12] are directed specifically at ethical and legal considerations. This phase provides an opportunity to document such aspects as unacceptable consequences and to develop the application with these issues in mind. But note that this framework does not comprehensively cover assurance of applications in which such failure modes occur (see the limitations in [1.2]).

Guidance note:

Evidence to support this claim may include documentation of the failure modes, risk and criticality analysis, the risk register, and list of mitigating actions.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.3.11 Claim: ethical consequences of use / misuse of the application have been considered.

This RP does not provide a comprehensive treatment of ethical considerations of data-driven applications, see /8/ for further guidance.

Guidance note:

Evidence to support this claim may include an analysis of ethical aspects linked to the failure modes identified in [4.2.3.10].

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.3.12 Claim: legal consequences of use / misuse of the application have been considered.

This RP does not provide a comprehensive treatment of legal considerations of data-driven applications.

Guidance note:

Evidence to support this claim may include an analysis of legal implications linked to the failure modes identified in [4.2.3.10].

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.3.13 Claim: the requirements on the application and data to meet the data-driven modelling and business goals and success criteria, and to mitigate unintended or missing inputs/outputs/interactions, are sufficiently defined.

This claim plays a central role in establishing whether the application meets the objectives defined in [1.2]. The performance of the application, and the risk of unacceptable output, form part of the evidence provided here.

Guidance note:

Evidence to support this claim may include:

- the requirements on data. This should include requirements on both the size of the data and the quality of the data. DNVGL-RP-0497 provides guidance on data quality metrics.
- the requirements on application performance. This should be expressed in business success criteria terms (e.g. KPIs)
- requirements on handling of inadequate or missing input / output.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.4 Determine data-driven modelling goals

4.2.4.1 Business goals and success criteria are typically non-technical end goals of the project. Modelling goals are more technical. Mapping the business objectives to data-driven modelling goals requires good communication between those with domain expertise and those with data science expertise.

4.2.4.2 Claim: the prediction target of the data-driven application is clearly defined and understood, based on the business objectives and success criteria.

The prediction target is the quantity the data-driven model will predict. For regression models this is a quantity (e.g. pressure, temperature, time-to-failure). For classifiers this is a class label.

Guidance note:

Evidence to support this claim may include a definition of the prediction target, and an explanation of how being able to predict the target will allow the application to meet the business success criteria.

For example, if the business objective is "reduce machine breakdown repair costs", then the target arrived at might be "predict machine failures N days in advance". The evidence provided to support this claim should then explain how being able to predict the machine failures N days in advance (the target) will realize a reduction in repair costs (the business objective). It is important to be specific in the definition: if N is not specified, then there is a risk that a model might be developed which performs very well for, say, N=1, but which does not help reduce costs at all because it takes 3 days to mobilize a maintenance intervention.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.4.3 Claim: the success criteria of the data-driven model (how well it predicts the target) are sufficiently and objectively defined.

Modelling success criteria are the most objective acceptance criteria that can be defined in reaching the modelling goals. If the activity [4.2.3] (assess situation) has been performed for technical success (risk and requirements analysis), the criteria can be heavily based on or be part of this.

If the criteria are of subjective character, the characteristics/competence of those who will judge if the criteria are met should be identified.

Guidance note:

Evidence to support this claim may include the specifications of the tests to be performed to determine if the success criteria are met. Requirements for reproducibility of these results should be specified.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.2.5 Produce project plan

4.2.5.1 Claim: an up-to-date project plan, covering development, test and deployment, is sufficiently defined.

The plan should include stages, tasks, resources, durations, timelines, milestones, deliverables, processes and methods defined and appropriate for the project purpose. The project plan should also address project risk, its relations to the project schedule and project quality.

Together with the project plan and the detailed plans for each phase, the evaluation strategy for each phase should also be determined.

The project plan should be reviewed and updated throughout the project at each phase.

This RP does not elaborate further on general project management practices. See /4/ for more information.

4.2.5.2 Claim: tools and techniques for data-driven modelling have been sufficiently assessed and selected.

Based on the information available on objectives and available data, techniques to be used for the application should be assessed at this stage and suitable tool(s) selected. Even if this might be changed at a later stage when more information is available, it should be recognized that technique- and tool changes are potentially time consuming and expensive. A good initial choice and a review at a later time is important for project success. It is crucial to establish a good overview of techniques and tools, together with their strengths and weaknesses. It is recommended that this is reviewed by an external entity.

The data-driven modelling tools and techniques selected shall be properly described and planned in the project plan.

Guidance note:

Evidence to support this claim may include arguments supporting the choice of tools / techniques selected compared to available alternatives. Factors affecting the choice may include speed, reliability, cost, ease-of-use, transparency, traceability, level of integration with other tools, security and privacy.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.3 Data understanding

4.3.1 General

Understanding the data is an important phase in which the available data is collected, described and explored to assess the data quality. Data engineering competence is required to collect and prepare the data. Data science competence is required to develop an understanding of the properties and possible shortcomings of the data. Key questions to be asked during this phase include: is the data complete and of sufficient size? Is the data representative for the defined problem? Are there data quality issues? The answers to these questions may then be used to decide whether the data is sufficient for the modelling phase, whether action shall be taken to obtain more / better quality data, or whether the project goals should be adjusted.

4.3.2 Collect initial data

4.3.2.1 Claim: required data sources are identified and data are collected from all required sources.

This is a data engineering step, involving accessing databases, data fetch and transformation. The data collection step shall be sufficiently documented.

Guidance note:

Evidence to support this claim may include:

— evidence that the data engineering step can be replicated by another person (with the necessary skills)
— evidence that a full data set according to requirements has been obtained
— metadata: this may be documented by a data sheet /19/, /21/ and /22/. Describe the provenance, lineage, semantics, formats etc.
— documentation showing the evaluation of contractual issues, intellectual property rights and the cost of accessing and using the data sources.

Already at this point, it is possible to estimate if the data set is sufficient for the needs and requirements, and, if it is not, the project resource planning phase should be revisited.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.3.3 Describe data

4.3.3.1 Claim: data properties have been analyzed and found in accordance with requirements / expectations.

The data is described at a high level, this focuses on the quantity and gross properties of the data and forms the basis for making an initial assertion on the suitability of the data for the intended purpose.

Guidance note:

Evidence to support this claim may include a data sheet (/19/, /21/ and /22/) and/or high level statements about the quantity and gross properties of the data, the number of data items, the number of features per data item, and relations between the features. Finer detail information may be provided in [4.3.4] below. Data quality is considered in section [4.3.5].

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.3.4 Explore data

4.3.4.1 Claim: data exploration has been successfully executed.

In this step a deeper investigation into the data's properties is performed. Exploratory data analysis provides a context in which to understand model interpretability / explainability measures determined as part of [4.5].

Guidance note:

Evidence to support this claim may include the results of exploratory data analysis. Many analysis techniques are available, /16/ gives a good overview. For example:

— the distribution of key attributes (for example, the target attribute of a prediction task)
— value counts and histograms
— scatterplots showing pairwise relationships between small numbers of attributes
— simple aggregations and summary statistics
— properties of significant sub-populations
— clustering analysis, PCA analysis or other unsupervised learning analysis
— other simple statistical analyses. These analyses may directly address the data-driven modelling goals, they may also contribute to or refine the data description and quality reports, and feed into the transformation and other data preparation steps needed for further analysis.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.3.5 Verify data quality

4.3.5.1 Claim: a data quality report exists and demonstrates the suitability of the data with respect to requirements.

The data quality necessary for proceeding to the next phase should be assessed and provide information about data completeness and representation, and the occurrence of missing or erroneous data. This information should be checked against the (data) requirements specified in [4.2.3]. Detailed knowledge of the available data may be used to assess the sensitivity to missing or erroneous data on the application performance and relate to risks identified earlier. Solutions to quality problems should be presented at this stage.

The choice of data repair strategy is dependent on the data-driven modelling goals. In some cases, replacing missing data with an average value might be a suitable solution, in other cases it might be better to delete the record or replace the data with a conservative value.

For example, AIS (ship position and ID) data often has holes in it. It does not make sense to replace missing position values by average values. Missing data could be replaced by interpolating between existing points, but in this case, it shall be ensured that the ship speed is feasible for the 'repaired' data and that the ship avoids sailing over land: this constrains the interpolation to generate only viable ship tracks, avoiding land. If the data gap cannot be interpolated in a realistic way it might be better to delete the record entirely or leave only part of the track. This example illustrates that data quality verification can be a complex task, and in some cases warrants a separate project in its own right.

DNVGL-RP-0497 provides a framework for data management with focus on data quality.

4.4 Data preparation

4.4.1 General

The data available from the data understanding phase can now be prepared for modelling. The raw data sets to use are selected, quality issues are fixed (data cleaning, compensation for known errors) and data sets constructed. Finally, data sets are integrated, formatted and harmonized to make them ready for input to modelling. Feature extraction, emedding and encoding occur as part of this step.

4.4.2 Select data

4.4.2.1 Claim: data is selected according to business objectives. Fields and records which are needed are clearly identified.

Data is selected based on the relevance to the data-driven goals, data requirements and initial data collection in [4.2.2].

Selection of data will often be performed iteratively in a loop between data preparation and modelling, eventually identifying the data set which is most suitable for the task. Unnecessary fields may be removed or it may become apparent that additional fields are needed.

4.4.3 Clean data

4.4.3.1 Data cleaning should address the data quality issues identified in the data quality report [4.2.5]) as well as any other data quality issues anticipated in the future.

There are several standard approaches to dealing with many of the common data quality issues. The appropriate approach to a specific data quality problem depends on the application of interest, how the quality-affected data is intended to be used and the severity of the problem. See DNVGL-RP-0497 for more information.

4.4.3.2 Claim: missing data records are adequately handled.

Guidance note:

Several strategies exist for dealing with missing data, ranging from simply ignoring the missing data, to imputing missing records using dedicated models. Evidence to support this claim may include details of the strategy used for dealing with missing data records, with an evaluation of risk associated with this strategy.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.4.3.3 Claim: missing field values are adequately handled.

Guidance note:

Evidence to support this claim may include details of the strategy used for filling in or imputing missing field values, and an evaluation of risk associated with this strategy.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.4.3.4 Claim: data quality issues are adequately handled.

This claim covers any handling of data quality issues not already covered by claims [4.4.3.2] and [4.4.3.3] above.

Guidance note:

Evidence to support this claim may include any strategy for data quality issue handling not already included in the evidence to support claims [4.4.3.2] and [4.4.3.3] above.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.4.3.5 Claim: the resulting data set is sufficiently consistent.

Guidance note:

Evidence to support this claim may include cross-checks on data consistency and spot-checks for known events / trends / statistical properties of the data.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.4.4 Assemble data

4.4.4.1 Claim: sufficient data for modelling can be constructed / assembled / integrated from the multiple data sources available.

Constructing data is a key activity in preparing a suitable data set for modelling and may involve deriving or transforming data attributes and/or creating new records. Data from different sources shall be properly integrated or joined. This step can also involve aggregating or summarizing information from multiple sources.

4.4.5 Format data

4.4.5.1 Claim: the format of all relevant data fields has been systematically analysed and adapted where necessary. All data formats are adequate for the subsequent analysis.

A final transformation or re-formatting operation is often performed in preparation for modelling, without changing the information content of the data. This step may also involve reordering or rearranging data.

4.5 Modelling

4.5.1 General

In the modelling phase, key technical choices are made concerning modelling techniques and test designs outlined in the project plan [4.2.5]. The phase is by nature iterative where the data-driven model is (re)-trained, assessed, and adjusted, until satisfactory results are obtained. In each iteration adjustments can be made to the data partitioning (train / test), feature engineering (feature coding, embedding) and training strategy (cross-validation, hyperparameter tuning).

Reuse of the same train / test partition in each iteration can lead to overfitting, resulting in a model with poor generalization capabilities. This is commonly avoided by partitioning the data into three sets: a held-out test set (which is used only for final evaluation to estimate expected model performance), a training set (used to train the model) and a validation set (used for evaluation of the model). The training / validation set partition may be shuffled within, or between, iterations.

Many resources exist to guide the practitioner towards making good choices in this phase, for example /11/ and /20/.

4.5.2 Select modelling technique

4.5.2.1 Claim: the modelling technique is clearly defined and appropriate for the task.

The choice of modelling technique should be based on what was selected and planned for in the project plan [4.2.5]. Several techniques may be chosen to run as parallel modelling tasks. Requirements for the chosen technique(s) shall include technical considerations, as well as aspects related to transparency, interpretability, technology understanding, processes, organization, competence or regulatory requirements.

Guidance note:

For guidance on suitable modelling techniques based on scope of the analysis, see for example /11/ and /20/.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.2.2 Claim: modelling assumptions are listed, and it is checked that the assumptions apply to the application.

Assumptions shall be clearly identified and hold true for the application. Claims concerning the assumptions for the deployable application are detailed in /1/.

Guidance note:

Modelling assumptions vary widely based on the modelling task and chosen modelling technique. The modelling assumptions should include statements regarding the fundamental assumptions made by the chosen technique, and an assessment of whether such assumptions are justified for the problem. For example linear regression models assume: linearity and additivity of the problem; data to be identically and independently distributed; homoskedasticity and; normality of errors. Applications involving time series typically make assumptions regarding stationarity. Other important assumptions may relate to the behaviour of the model to specific data quality issues, for example in the case of missing input data.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.2.3 Claim: the chosen model is sufficiently transparent (explainable/interpretable) or can be de-constructed such that verification/validation is possible.

There are indications, e.g. /8/, that regulatory frameworks are giving increasing focus to transparency. For some applications the requirements on transparency may be zero, i.e. a black box is suitable, and it may be sufficient that the performance of the model satisfies acceptance criteria [4.5.5.5]. For others, a high level of transparency is required. In some jurisdictions there may be additional requirements on transparency for applications involving personal information (e.g. customer or employee details in the EU under GDPR /15/). It is recognized that specifying requirements on transparency is challenging since this field is relatively young and immature.

Guidance note:

Depending on the modelling technique chosen, the model and/or its predictions may be to a certain extent transparent (explainable and/or interpretable). Model explainability / interpretability /14/ is often expressed by extracting or deriving importance measures of input features and gives a global indication of which input parameters are the most important. Tools such as LIME /12/ and SHAP /13/ may be used to give explainability insights. Prediction explainability brings interpretation to predictions, explaining why a model made a certain prediction. This is an active area in AI research and new techniques for explainability and interpretability are emerging.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.2.4 Claim: a model baseline is defined.

The cost (in time and resources) of training, deploying and maintaining a model is closely linked to the complexity of the model. The cost is only justified if the performance is significantly better than a simple model baseline. A model baseline may be used (later in [4.5.5.6]) to check that the model complexity is warranted.

Guidance note:

It is good practice to choose a baseline modelling technique which gives simple, interpretable models, even though the performance of such models may be far inferior to the main model. A baseline model can be as simple as i) for regression problems: a linear regression model or ii) for classification problems: a classifier always estimating the most frequent class observed in the training data. A baseline model facilitates understanding of the problem and provides a reference against which the cost/benefit tradeoffs of more complex and computationally demanding algorithms can be quantified. For example, if the problem being solved is rather simple, then there is no benefit in using a highly complex and computationally demanding technique.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.3 Generate test design

4.5.3.1 Claim: a suitable test design description is available.

It is important to establish a suitable test design to assess the quality of the model. This should include requirements related to the amount of data, the splitting of data between training / validation / testing, and other properties of the test data. The test design should also include consideration of the reproducibility of test results. As far as possible the test design should make use of the acceptance criteria identified in [4.2.3]. Specific properties of the test data are the topics of claim [4.5.4.4] and the claims in [4.5.5] below.

4.5.4 Build model

4.5.4.1 The core modelling activity is an iteration of modelling and model assessment activities, using the chosen modelling technique and established test design.

4.5.4.2 Claim: parameters are listed, and settings are specified, together with any relevant rationale.

Guidance note:

Evidence to support this claim may include the input parameters and hyperparameters used during modelling, such as topography, activation functions and optimization metrics.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.4.3 Claim: model parameters are included in configuration management.

During iterative modelling it is important to have full control of the model techniques and parameters to understand their effect and to be able to roll back to earlier versions.

Guidance note:

Evidence to support this claim may include information of the configuration management system / repository used.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.4.4 Claim: partitioning of data (if relevant) is done without introducing bias.

A key step is to partition the data into independent training / validation and test sets.

Guidance note:

Evidence to support this claim may include comparisons of statistical properties of partitioned data sets (e.g. training / validation / test). This may be done statistically (mean, variance for numerical features/targets, value counts for categorical features/targets) and graphically (Q-Q plots for numerical features/targets, heatmaps for categorical features/targets).

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.4.5 Claim: thresholds on data quality for training / validation / testing are documented.

Guidance note:

Evidence to support this claim may include thresholds on data quality metrics such as completeness (DNVGL-RP-0497).

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.4.6 Claim: an appropriate model is designed, implemented and trained.

Guidance note:

Evidence to support this claim may include training logs and other information which show that the model training step executed successfully and a model is produced.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.4.7 Claim: the model is appropriately documented.

Guidance note:

Evidence to support this claim may include a summary of any model properties (size, complexity) not already covered by [4.5.4.2].

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.4.8 Claim: the model's expected sensitivity to data quality problems is documented.

Guidance note:

Evidence to support this claim may include a list describing how the model is expected to respond to common syntactic/semantic/pragmatic data quality problems (see DNVGL-RP-0497).

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.5 Model assessment

4.5.5.1 The model's performance is compared with the acceptance criteria defined in [4.2.5.1]. The claims in this section are primarily documented with statistical / argument-based evidence which in most cases is assembled manually. App.B describes steps towards direct assessment: automatic assessment of model performance.

4.5.5.2 Claim: the data used to validate / test the model is representative.

The performance of the model during validation / testing provides an indication of the performance to be expected when the model is deployed in the real world. If, for whatever reason, the data used for validation / testing is not representative, then the performance measured is no longer a good indication of real-world performance. It is therefore important to provide evidence that this data is representative.

Note that in some scenarios (e.g. anomaly detection for predictive maintenance) there may be insufficient validation / test data to cover all failure modes, implying that the validation / test data is not representative. In such cases it may be possible to use simulations to augment the available data set with data to represent the missing failure modes. The risk of using such simulated data should be evaluated and documented.

Guidance note:

Evidence to support this claim may be statistical and/or argument-based. For example, if a model is being developed to predict a structure's response to sea waves, evidence should be provided that the validation / test data sufficiently covers the height, frequency, intensity of waves expected in deployment. Evidence may also be provided that the quantity of data used for validating / testing the model is sufficient.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.5.3 Claim: The data used to validate / test the model is independent of the training data.

Guidance note:

If the validation / test data is independent of the training data, then model assessment is fair and the results of the assessment are likely to be transferrable to other data. If, on the other hand, leakage occurs from the training data to the validation / test data, then the assessment may be biased and show better than actual performance.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.5.4 Claim: The data used to validate / test the model has similar statistical properties to the training data.

Guidance note:

While [4.5.5.3] considers independence of the training and validation / test data, this claim concerns their statistical similarity. The statistical properties of the data sets may be characterized by distributions of the input features, distributions of target variables, and may also include data quality measures.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.5.5 Claim: the model is properly assessed, and necessary iterations performed until satisfactory performance results are obtained.

Guidance note:

Evidence should be provided that, during each iteration of development , the model has been assessed according to the project plan [4.2.5.1]. If a baseline model has been included, then the baseline performance should be compared with the performance of the main model.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.5.5.6 Claim: the model is better than the baseline.

Guidance note:

Evidence for this claim may include performance statistics on validation / test data which compares performance metrics measured for the model under development versus the baseline defined in [4.5.2.4].

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.6 Evaluation

4.6.1 General

The evaluation phase aims at evaluating two aspects: i) the application and whether it satisfies its goals, success criteria and requirements, and ii) the project itself.

Even if the success criteria are not satisfied, evaluation of the application with respect to the business objectives, and of the project, may provide useful insight which can be used in future iterations and/or other projects.

4.6.2 Evaluate results

4.6.2.1 In the modelling phase, the datadriven model is tested until satisfaction is reached. The evaluation presented in this section serves as a final step before the application (which includes the data-driven model) is approved for deployment.

The minimum action required is to assess if the application meets the business success criteria. With well-defined datadriven modelling success criteria, requirements, acceptance and evaluation criteria, this task should be to review earlier evidence and confirm that requirements are met.

4.6.2.2 Claim: the application is properly assessed with respect to business success criteria, data science success criteria and the requirements from the risk analysis (see [4.2.2] and [4.2.3]).

Guidance note:

The application performance should be explained in the context of the performance of a baseline [4.5.2.4]. For applications using regression models, performance may be measured by one or more performance metrics including: mean squared error (MSE), root mean squared error (RMSE), mean absolute percentage error (MAPE). For applications using classification models, performance may be measured by one or more metrics including: confusion matrices, area under curve (AUC), classifier precision, recall, classifier accuracy, sensitivity, F1-score. The business success criteria may specify which performance metrics shall be used. For example, for a predictive maintenance system the cost of a false negative (a component failure when the application predicted normal operations) can be much higher than the cost of a false positive (an inspection prompted by the application predicting a failure even when the machine was in fact OK).

		Observed	
		OK	Fault
Prediction	OK	True Negative	False Negative
	Fault	False Positive	True Positive

		Observed	
		OK	Fault
Prediction	OK	6520	48
	Fault	125	65

Figure 4-1 Example of a confusion matrix. Top: showing true/false positives/negatives. Bottom: showing example numbers for a experimental model.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.6.2.3 Claim: the application is tested in as realistic environment as possible.

Guidance note:

There is often a discrepancy between model performance observed in deployment and that observed during development. There are several reasons behind this discrepancy, which can influence the degree of such discrepancy, and careful modelling and testing can mitigate this tendency. One way to mitigate the risk of model drift is to periodically test the model on data collected post-deployment.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.6.2.4 Claim: the application's sensitivity to expected data quality problems is understood and quantified.

This claim supplements [4.5.4.8] which is concerned with the model's sensitivity to data quality problems. The current claim is concerned with the larger context of the application: the application may use redundancy and/or other mechanisms to handle model failure caused by data quality problems.

The data quality report ([4.3.5.1]) identifies known data quality problems in the available data. Some of these data quality problems may be explicitly dealt with through cleaning ([4.4.3]) but some may not. It is important to understand the impact of any unfixed data quality problems at prediction time. In extreme cases, missing data could cause a data-driven model to crash, which if not caught and handled at the application level could trigger other failures.

Guidance note:

It is important to understand the impact of any unfixed data quality issues at prediction time. In extreme cases, missing data can cause a datadriven application to crash, resulting in no prediction and potentially causing failures in other parts of the ecosystem.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.6.2.5 Claim: models in critical applications are tested with extended test sets based on context-specific objectives.

With increasing criticality, and wherever feasible, models should be tested in a realistic production environment and if possible with extended test data sets which are based on objectives such as robustness to data quality issues, degree of generalization, level of fairness/balance and profiling of the model's output domain from expected input data characteristics.

4.6.2.6 Claim: the application is approved for intended use.

The evaluation step ends with an approval of the application based on selection criteria and all assessment and evaluation results. Once approved the application may be deployed.

Guidance note:

Evidence for this claim may be a formal sign-off by the project owner.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.6.3 Review process

4.6.3.1 Claim: an appropriate process review has been performed and documented.

Guidance note:

Reviews are crucial for a learning organization. Since the use of data-driven applications in the real world is still very much in its infancy, reviews are even more important. Reviewing the process may include meetings with involved parties, memos, or lessons learned. Reviews might also be conducted by third party organizations, which is especially recommended for more critical applications.

In general, reviewing what is presented is relatively straightforward, while detecting what is missing is not. Special care should therefore be taken for defining methods which are suitable for detecting what is missing - e.g. brain-storming processes, usage of independent parties.

Note that this activity includes the review of the process leading to the application, while the project as such is reviewed in activity [4.6.5]. In small projects the two might be combined.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.6.4 Produce final report

4.6.4.1 The final report also represents an opportunity to describe the project in such a way to communicate quality issues and areas for improvement.

4.6.4.2 Claim: a final report has been issued.

A basic final report, compiling and organizing all relevant project outputs, should always be issued.

4.6.4.3 Claim: a presentation to relevant stakeholders has been held.

A presentation ensures that relevant stakeholders become aware of the existence of the application and receive some basic information about the project.

4.6.5 Review project

4.6.5.1 Claim: the experience from the project has been documented.

The total project is reviewed with a focus on the project success and the fulfilment of the overall objectives, building on the process review performed in phase [4.6.3] (review process).

4.6.6 Next steps

4.6.6.1 Claim: a list of possible next steps has been sufficiently identified.

Guidance note:

Through use of this assurance framework the project will deliver a suite of documentation about the application, the data and models used in its development. Additionally documenting possible next steps provides a useful starting point for following projects. Evaluations and reviews performed lay the foundation for deciding the next steps in the project. Are the results mature for deployment or should further iterations be done? This points back to [3.1] and the need for iterating the development process until satisfactory results are obtained. The step is an important milestone or decision gate for all stakeholders in the project who should then be included. The decision might be facilitated by a person outside the project.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.7 Deployment and operations

4.7.1 General

It is recommended to plan for deployment, even in the special case where the model is a one-time model intended purely for analysis. A minimum requirement is to document deployment considerations in the project report.

4.7.2 Plan deployment

4.7.2.1 Claim: a deployment plan has been issued.

For the application to be successfully set into use, a deployment plan shall be made. This typically involves some sort of operative use or environment (human operators, operative data sources, specific IT and hardware, etc.) and shall be sufficiently planned. The plan may also include information to users, problems to avoid, financial issues and foreseen monitoring and maintenance needs. The development of the deployment plan can be a valuable collaboration between stakeholders and end users, increasing the success rate. This can be drafted in advance, in parallel to the development process, giving valuable input and a fast deployment process.

4.7.3 Plan monitoring and maintenance

4.7.3.1 Claim: a monitoring and maintenance plan for the application has been issued.

For the application to be deployed and operational throughout its life-cycle, monitoring and maintenance plans shall be made. Claims in later sections [4.7.4] and [4.7.5] cover the details of operations context and monitoring. The current claim relates to the existence of a plan.

Guidance note:

Documentation and configuration control ensure that it is possible to step back in time, revert changes if necessary, and keep the deployed application running. Consideration should be given to the mechanism for making updates, which can be calendar-based, or triggered by changes to the operative environment, the application domain and/or changes in the business objectives. The use and performance of the application should also be monitored to determine when maintenance is needed.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.7.4 Operations context

4.7.4.1 To assure that use of the application will not lead to unwanted consequences, the operations context in which the application shall be used shall be carefully considered. The claims listed here complement those in [4.2.3] which focuses on Business understanding prior to and during modelling. Here the focus is on the behaviour of the deployable/deployed application in operations, that is, after modelling. Certain failure modes of the application may only apply in operations, and may only become apparent during testing of the application after model training is complete.

The evidence supporting these claims is an important part of the application documentation which enables those responsible for running operations to ensure smooth operations.

4.7.4.2 Claim: the assumptions underlying the application's operation are sufficiently documented.

If the assumptions under which the application was designed become invalid, then there is a risk that the application will not function as expected. Documentation of these assumptions is one way to minimize the risk of using the application in a situation where those assumptions do not hold.

Guidance note:

Evidence to support this claim may include a list of assumptions underlying the application's operation. In addition to assumptions concerning the data, evidence may also include assumptions concerning application use and user behaviour.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.7.4.3 Claim: the assumptions underlying the application's operation are valid.

Guidance note:

Evidence to support this claim may include details of how the validity of assumptions is evaluated, comparing the operations context (environment, intended use) with the assumptions documented in [4.7.4.2].

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.7.4.4 Claim: any contingency / redundancy / fallback mechanisms used in operations to handle wrong predictions or other application failure modes are sufficiently documented.

The application may include contingency for dealing with wrong predictions, such as data quality metrics to detect certain types of wrong predictions, redundancy in the application itself and/or fallback mechanisms. Documenting how these mechanisms are activated and what they do provides not only useful information to those integrating the application in other systems, but also enables management of risk.

Guidance note:

Evidence to support this claim may include details of the contingency / redundancy / fallback mechanisms, related to the failure modes described in [4.2.3.10].

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.7.5 Operations monitoring

4.7.5.1 Once the application is tested and deployed, its continued operation without unwanted consequences depends on a well-designed approach to monitoring performance.

The key to successful operation of data-driven applications is monitoring. If possible, monitoring should be continuous and real-time, easily available to the operations personnel and with information presented in a way which is easy to act on.

4.7.5.2 Claim: the validity of the assumptions underlying the application's operation are continuously / periodically monitored.

It is seldom sufficient to do a one-time check of assumptions: data-driven applications typically require periodic retraining, and the training data can drift over time, meaning that assumptions valid at one time are not valid at another. It is challenging to establish conclusively that previously good modelling assumptions still hold true, but measures of data drift can be sufficient to raise a warning when, for example, the current training data is statistically different to the original training data.

Guidance note:

Evidence to support this claim may include documentation about how, and how often, the validity of the assumptions is assessed, see [4.7.4.4].

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

4.7.5.3 Claim: the data quality of input data is continuously / periodically monitored.

The data quality of the application input should be monitored, since many of the data quality issues which can affect the application can in principle be detected before the data is used to make predictions.

4.7.5.4 Claim: the performance of the application is continuously / periodically monitored.

Monitoring of the application performance is important for detecting the problems not detected by [4.7.5.2] and [4.7.5.3].

4.7.5.5 Claim: objective performance requirements for the application in operation are sufficiently documented, and a plan exists for how to act if the application performance does not meet these requirements.

It makes good sense to decide before deployment how performance will be monitored, and what to do if the performance drops below an agreed threshold. There should be requirements about how to test (with what data?), how often to test, and how to measure the result.

4.7.5.6 Claim: the schedule / triggers for application retraining are defined.

A retraining plan should be established. The criteria which trigger retraining should be defined and documented. Retraining may occur at a fixed frequency (e.g. every 6 months), or triggered by performance or data metrics. What is appropriate for the application depends on the type of model, the data size and dynamics, and the expectations of the application performance.

APPENDIX A RISK ASSESSMENT OF USE OF APPLICATIONS WITH DATA-DRIVEN COMPONENTS

A.1 Risk assessment of an application comprising a stand-alone data-driven model

One way in which this RP can be used is to assess risk associated with an application which comprises a single data-driven model. This appendix provides a simple example of how such a risk assessment can be carried out.

The assessment analyzes the model development and deployment process to identify items of risk. The steps in the assessment, as shown in Figure A-1, are as follows:

1) Compile a checklist with questions derived from the claims in the RP. The set of questions together cover all the claims.
2) Prepare a risk matrix, define scales of probability / likelihood and consequence.
3) Together with the application owner/developer, fill out the checklist.
4) Use the responses to this checklist to populate a risk register.
5) For each item in the risk register identify one or more unwanted effects / risk items.
6) For each risk item, estimate the probability and consequence, and from these calculate the risk (risk = probability x consequence).
7) Grade these risks using the risk matrix.
8) (Optionally) Define mitigating actions for the highest risk items.

A.2 Risk assessment of a hybrid application containing multiple data-driven components

Many real-life applications are not purely data-driven, but instead combine data-driven components with components based on physics and/or heuristics or rules. The approach defined above for applications comprising single models can still be used, with modification, to perform a risk assessment of such hybrid systems.

As shown in Figure A-2 a checklist-based identification of risk can be carried out separately for each submodel, followed by an additional step which considers risk associated with combining predictions from the submodels. An application risk register is then populated and a risk matrix is used to estimate risk for each item in the register. Finally, mitigating actions can be defined and executed.

Figure A-1 Workflow for risk assessment of data-driven applications

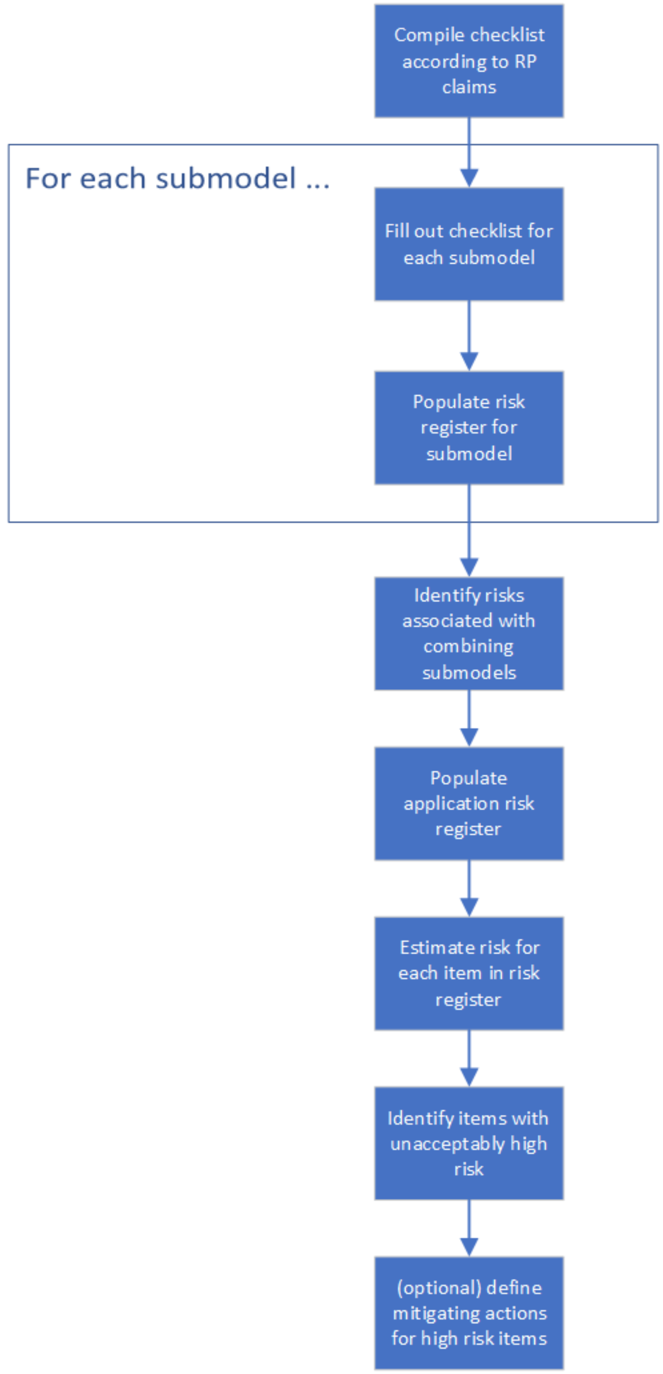

Figure A-2 Workflow for risk assessment of hybrid applications

APPENDIX B DIRECT ASSESSMENT

The assessment described in App.A assesses the risk of using a data-driven application by assessing the process by which that model was developed and deployed. It is also possible, though often more challenging, to assess the risk of using a data-driven application by analyzing the behaviour of the application itself. We call this direct assessment.

The current RP supports both indirect (assess the process) and direct (assess the application) approaches. For direct assessment closer access to the data-driven model and/or training and test data is required. There may be commercial and/or technical reasons why this is not possible, and so direct assessment is not a general requirement for assurance.

Approaches to explaining model behaviour can be divided into clear box and black box approaches. Clear box approaches require analytic insight into the model parameters to identify which parameters are the most important, and which are ignored. Black box approaches, in contrast, just take the model as is and infer which parameters are relevant by analysis of its behaviour.

In supervised learning, an assumption often implicitly made is that the real-life data seen in deployment has the same statistical properties as the training, and test data, seen during development. Evidence for this assumption is provided in claims [4.5.4.1] and [4.6.1.2]. If this assumption is not valid, then the real-life data seen in deployment is "out of distribution". In such cases there is an increased need for evidence that the application is behaving in a sensible manner. Tools for explainability / interpretability can provide such evidence.

For any particular model, the appropriate level of explainability / interpretability required to give a complete risk picture, and to give assurance that an application will perform as expected without unwanted consequences, depends on the intended use of the application and the criticality. This is a topic of ongoing research receiving much attention in the field of AI/ML, and it can be expected that a clearer picture regarding what is required and which techniques are appropriate to assess performance will emerge in the coming years.

The current RP is nevertheless still appropriate for applications which require direct assessment, the evidence provided for the claims related to data quality and evaluation can be evaluated numerically using data quality metrics (metrics such as completeness, within range) and evaluation metrics (on known test data, metrics such as AUC, accuracy, MSE). Risk assessment for these particular claims can be connected to the metrics such that this part of the risk assessment becomes objective (i.e. no longer based on the evaluation provided by an expert's opinion).

Making these parts of the risk assessment objectively measurable opens the possibility of providing assurance as a runtime service. The calculation of quality metrics can be provided by runtime services which monitor data quality of the input and output to the application. If/when the input/output data quality is low, these runtime services can detect that deviation, and automatic checks on this evidence can be used to flag the fact that the assurance case is weakened and ultimately that the model in its current state is not assured. This is an aspect which DNV GL expects to receive more attention in the future.

DNV GL recommends that a process risk assessment be the starting point for assurance, and that direct assessment and runtime monitoring are added at a later phase. At the current time DNV GL believes that direct assessment alone is insufficient to provide assurance that a given model will perform as required without unwanted consequences.

APPENDIX C USE CASE: APPLICATION TO PREDICTIVE MAINTENANCE

Predictive maintenance techniques are designed to help determine the condition of in-service machine or equipment (item) in order to predict when maintenance should be performed. This approach promises cost savings over routine or time-based preventive maintenance, because tasks are performed only when warranted.

The main premise of this maintenance is to allow convenient scheduling of corrective maintenance, and to prevent unexpected failures.

In the domain of equipment maintenance, interest in data-driven applications is primarily motivated by an ambition to reduce maintenance costs and improve equipment efficiency and availability. As illustrated in Figure C-1, by increasing the complexity of the analytics it is possible to move beyond descriptive analytics and condition monitoring (CM) to diagnostic analytics and beyond to predictive analytics and ultimately prescriptive analytics. Predictive analytics go beyond explaining past/current behaviour to making predictions about future failures and determine remaining useful life (RUL) of the equipment. These predictive analytics techniques include failure modeling techniques that are physicsbased, knowledge-based and/or data-driven.

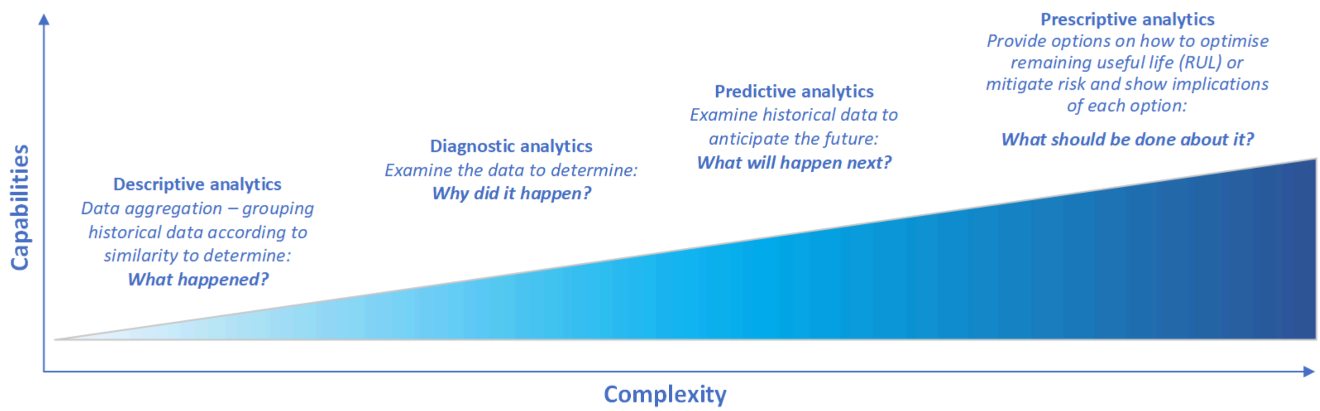

Figure C-1 Types of analytics

One approach adopted by the class society is described in DNVGL-CP-0484 that provides a common framework for approving a company to deliver a specific service such as condition-based maintenance (CBM) (see DNVGL-CP-0484 App.B). As part of this program it outlines how maintenance for items can be established or optimized focusing on condition-based maintenance (CBM), and including failure modeling techniques which are data-driven (such as machine learning).

A data-driven application used for CBM is normally part of a maintenance strategy for an item that can include both predetermined maintenance and predictive maintenance. In this context the data-driven application will normally be used to predict and/or to mitigate degradation related to a single or several failure modes to reduce probability of a functional failure.

To understand and determine performance requirements for a data-driven application it should be evaluated as part of a systematic process with the objective to identify the most effective and efficient preventive maintenance tasks to an item at optimal frequencies in order to retain its ability to perform its required functions over a given period of time.

A widely recognized approach by maintenance professionals is the reliability centered maintenance (RCM) analysis, see IEC 60300-3-11 /24/.

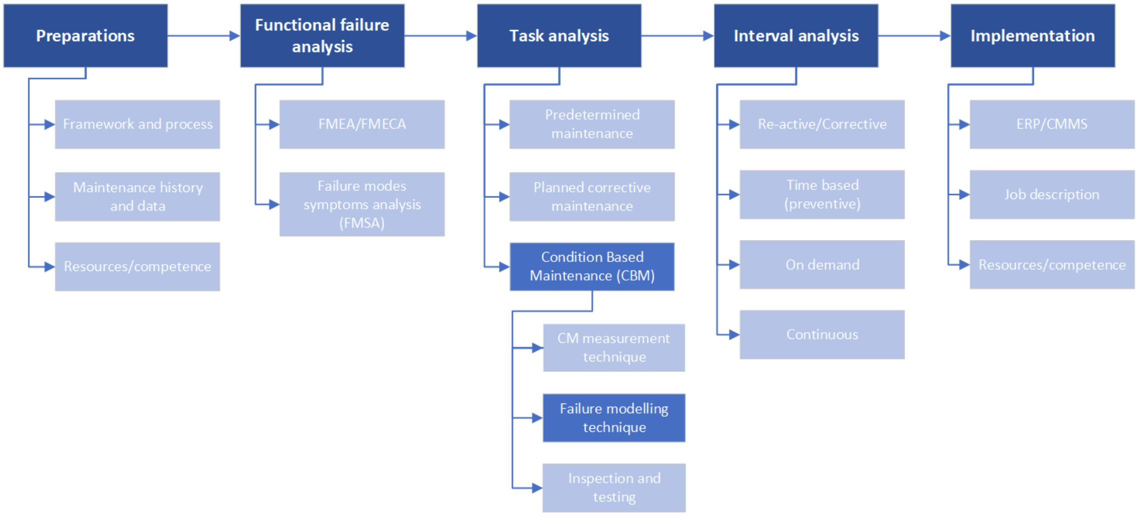

Figure C-2 Steps in RCM analysis

Preparations: the first step in the RCM analysis (process) starts with preparing the process and identifying applicable framework, get access to correct data and information and resources needed to perform the analysis.

The RCM analysis and the development of a failure modeling technique like a data-driven application for CBM will normally be split in two activities, but the two should be closely linked to get a common understanding and ensure the overall goal to identify the most efficient preventive maintenance for the item.

Functional failure analysis: the second step is to perform an evaluation of applicable functions with associated functional failures (failure modes). This evaluation starts with a failure mode, effects and criticality analysis (FMECA), see IEC 60812 /28/. This can further be extended with a failure mode and symptoms analysis (FMSA), see ISO 13379-1 /17/, that will provide a good basis for a comparative analysis of diagnostic models that can be used offering different levels of capabilities and results. With the RCM analysis it is the function that is being analyzed, not the item itself and the analysis should address all applicable functional failures such as:

— complete failure to perform a function
— under-performance (partial failure)
— over-performance (partial failure)
— unintended performance.

Task and interval analysis: the third step in the analysis is the maintenance tasks analysis where each functional failure is evaluated and where the most efficient preventive maintenance is determined. This process will normally be based on a predefined logic that will ask questions about the applicable failure mode performance requirements, risk (safety) level, type of degradations, symptoms, possibilities for detection and maintainability of the item. Based on this it will suggest the most effective and efficient maintenance approach with a focus on using CBM if applicable.

Implementation (deployment): in the final stage of the process, the data-driven application will be implemented as part of a maintenance strategy for an item. The implementation should ensure:

— an active monitoring to measure the performance of the data-driven application
— implementation of the related maintenance tasks in CMMS (typically planned corrective maintenance activated by the data-driven application)
— sufficient data management and data quality monitoring for parameters used as input by the data-driven application

— continuous improvement and validation of assumptions and operating conditions for the data-driven application.

C.1 Verification of data-driven applications

The approach adopted by DNVGL-CP-0484 sets requirements for the verification of modelling techniques used. The verification is performed by calculating a confidence metric /17/ for the modelling technique. This is a figure of merit essentially representing the cumulative effect of sources of error and uncertainty on the expected performance of the modelling technique. The overall confidence level is calculated as the weighted combination of confidence values estimated for each aspect of the system.

Table C-1 shows how these aspects are related to claims in the current RP, used for assurance (verification) of data-driven applications, with claims of high importance marked in **bold**. The table also provides the applicable requirements and suggests evidence (provided by the application owner) used for compliance/ documentation.

The mapping shown in Table C-1 divides requirements into the following aspects:

1) project execution
2) data basis
3) development choices
4) performance
5) deployment.

Table C-1 Mapping between requirements and claims for predictive maintenance applications which use data-driven models. Claims of particularly high importance are marked in bold.

Aspect	*Claim and topic*	*Requirement reference (DNVGL-CP-0484 App.B [5]) and details*	*Typical evidence (provided by the application owner)*
Project execution	[4.2.2.2] **Business context**	DNVGL-CP-0484 App.B [5.2.3.1], DNVGL-CP-0484 App.B [5.2.3.2], DNVGL-CP-0484 App.B [5.2.3.4] The operational context of the application shall be fully documented. It shall be clearly stated which equipment failure mode and/ or degradation the application is intended to predict together with a description of any interactions (automatic or manual).	Results of RCM analysis /24/ and/or task analysis (DNVGL-CP-0484 App.B [5.2.3.1]) comparing different strategies for preventive maintenance.
	[4.2.2.3] Business objectives	The business objectives shall be described in terms of one or more measurable KPIs.	Examples: — assess and/or mitigate degradation related to XX (one or more specific equipment failure modes of machine M) to reduce probability of a functional failure — reduce maintenance costs" or "reduce unscheduled downtime

Aspect	*Claim and topic*	*Requirement reference (DNVGL-CP-0484 App.B [5]) and details*	*Typical evidence (provided by the application owner)*
Project execution	[4.2.2.4] **Business success criteria**	No additional requirements.	Examples: — reduce 5-year maintenance costs by XX % — get XX% higher RUL from the item.
	[4.2.3.2] Key stakeholders		Identify owners, maintenance department, OEM, supplier, etc.
	[3.2.1.1] Organizational data science maturity	DNVGL-CP-0484 App.B [5.2.4] The organization's data management and data science maturity shall be documented.	Data management maturity assessment report as defined in DNVGL-RP-0497.
	[3.2.1.2] Team competence, tools	DNVGL-CP-0484 App.B [5.2.5.2], DNVGL-CP-0484 App.B [5.2.5.3] The team's competence shall be documented.	CVs and documentation of relevant experience in executing data science projects.
	[4.2.3.3] Data resources	DNVGL-CP-0484 App.B [5.2.4.1], DNVGL-CP-0484 App.B [5.2.4.3] Process for data management, data quality assessment shall be documented.	Examples: — data lineage and process descriptions for all processes related to input data — data profiling report, data quality assessment (DNVGL-RP-0497, ISO 8000-8 /18/).
	[4.2.3.4] Project requirements	No additional requirements.	
	[4.2.3.5] Project assumptions	DNVGL-CP-0484 App.B [5.2.3.1] No additional requirements.	
	[4.2.3.6] Project risk	No additional requirements.	
	[4.2.3.7] Terminology	DNVGL-CP-0484 App.B [5.1.6.2] Standard terminology shall be defined and used. Where no standard terminology is available this shall be indicated.	See EN 13306 /25/, ISO 8000-2 /26/, ISO 13372 /27/.

Aspect	*Claim and topic*	*Requirement reference (DNVGL-CP-0484 App.B [5]) and details*	*Typical evidence (provided by the application owner)*
Project execution	[4.2.3.8] Application design	DNVGL-CP-0484 App.B [5.2.3.2], DNVGL-CP-0484 App.B [5.2.3.3] Application design documentation shall include a description of all relevant model(s), sub-model(s) and parameters.	Fault tree with the equipment failure mode as the trigger event (top event). The fault tree may be split into sub-structures to capture all relevant cases.
	[4.2.3.9] Cost versus benefit	To document feasibility of the project it is recommended to perform an analysis considering lifecycle cost, cost of lost production time, consequential damage and warranty and insurance. This can be quantified by calculating today's value of expected cash flows discounted by today's value of cash invested in development, deployment and operation. The analysis should also evaluate the current maintenance cost (strategy) and estimate the effect(s) of the application measured with the business goals and success criteria and the effect (cost reduction) this represents over time (life cycle). Synergy effects from the application development and implementation should be considered.	

Aspect	*Claim and topic*	*Requirement reference (DNVGL-CP-0484 App.B [5]) and details*	*Typical evidence (provided by the application owner)*
Project execution	[4.2.3.10] **Application failure modes**	DNVGL-CP-0484 App.B [5.2.3.1], DNVGL-CP-0484 App.B [5.2.3.2], DNVGL-CP-0484 App.B [5.2.3.4] Failure modes for the application (application failures that will render unsuitable output not in accordance with the performance requirements) shall be determined and evaluated according to the consequence they will have for the functional failure mode related to out-of-service time, failure rate and mean time to failure, redundancy and secondary damage/fault. See also [4.7.3.3] for handling of inadequate or missing input / output. Acceptance criteria for the application related to risk are covered under claim [4.2.3.13].	Report describing the failure modes of the application, identifying how they are linked to the failure modes of the item (machinery / equipment). This is preferably done by updating the FMECA/ FMEA of the item.
	[4.2.3.11] Ethical considerations	No additional requirements.	
	[4.2.3.12] Legal considerations	Any change to the current maintenance approach shall be considered and evaluated according to applicable regulatory requirements and suppliers insurance or warranty.	Relevant regulators to be considered: class, shelf state, flag state, operator, etc.

Aspect	*Claim and topic*	*Requirement reference (DNVGL-CP-0484 App.B [5]) and details*	*Typical evidence (provided by the application owner)*
Project execution	[4.2.3.13] **Requirements**	DNVGL-CP-0484 App.B [5.2.3.7] Performance requirements on the application shall be based on machine reliability and availability and the risk level (safety critical, production critical and environmental critical) of the functional failure mode that the application aims to predict/prevent. Performance requirements shall be determined for model accuracy, prediction target and probability of detection. An applicable risk level (which is used to evaluate the application's confidence level under claim [4.6.1.1]) shall be defined.	Documentation may include: FMEA/FMECA, Reliability block diagram, RCM task analysis defining requirements for remaining useful life (RUL) or estimated time to failure (ETTF), probability of detection, accuracy (ISO 5725-1 /10/) or other documented methods that quantify prediction performance. Documented applicable safety level.
	[4.2.5.1] Project plan	No additional requirements.	
	[4.2.5.2] Choice of tools	No additional requirements.	
	[3.3.1.2] Version control of evidence	DNVGL-CP-0484 App.B [5.2.3.9] No additional requirements.	
	[3.3.1.3] Version control of requirements	DNVGL-CP-0484 App.B [5.2.3.9] No additional requirements.	
	[3.3.1.4] Version control of data	DNVGL-CP-0484 App.B [5.2.3.9] No additional requirements.	
	[3.3.1.5] Version control of features	DNVGL-CP-0484 App.B [5.2.3.9] No additional requirements.	
	[3.3.1.6] Version control of models	DNVGL-CP-0484 App.B [5.2.3.9] No additional requirements.	
	[3.3.1.7] Version control of test data	DNVGL-CP-0484 App.B [5.2.3.9] No additional requirements.	

Aspect	*Claim and topic*	*Requirement reference (DNVGL-CP-0484 App.B [5]) and details*	*Typical evidence (provided by the application owner)*
Project execution	[3.3.2.2] Sufficient iterations for performance	No additional requirements.	
	[3.3.2.3] Sufficient iterations for risk	No additional requirements.	
	[3.3.3.2] Infrastructure	DNVGL-CP-0484 App.B [5.2.4.3] No additional requirements.	
	[3.3.3.3] Coding	No additional requirements.	
	[3.3.3.4] Data management	DNVGL-CP-0484 App.B [5.2.4] All parameters (sensor data) collected and used as input for the application, shall be quality assured by a sufficient level of data management to ensure that the data is protected and that the data acquisition do not introduce data security threats/breaches to the asset network. Sufficient data management shall be documented.	See also [3.3.3.2], [4.3.5.1] and data management maturity assessment as defined in DNVGL-RP-0497. Data management may be documented by: data flow diagram showing all relevant infrastructure and nodes (components) in the system, data quality assessment process, topology diagram showing system architecture of the infrastructure source system(s), intermediary components and target system, security system specification and diagram showing zones and conduits for all systems and interconnections and description of responsibilities and competence requirements related to data management.
	[4.6.1.5] Application approval	DNVGL-CP-0484 App.B [5.2.3.5], DNVGL-CP-0484 App.B [5.2.3.9] No additional requirements.	
	[4.6.2.1] Process review	DNVGL-CP-0484 App.B [5.2.5.4] No additional requirements.	
	[4.6.3.1] Final report	DNVGL-CP-0484 App.B [5.2.3.9] No additional requirements.	
	[4.6.3.2] Final presentation	DNVGL-CP-0484 App.B [5.2.3.9] No additional requirements.	

Aspect	*Claim and topic*	*Requirement reference (DNVGL-CP-0484 App.B [5]) and details*	*Typical evidence (provided by the application owner)*
Project execution	[4.6.4.1] Lessons learned	DNVGL-CP-0484 App.B [5.2.3.9] No additional requirements.	
	[4.6.5.1] Next steps	DNVGL-CP-0484 App.B [5.2.3.9] No additional requirements.	
Data basis	[4.3.2.1] Data collection	DNVGL-CP-0484 App.B [5.2.3.9] No additional requirements.	
	[4.3.3.1] Data characteristics	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	
	[4.3.4.1] Data exploration	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	
	[4.3.5.1] **Data quality**	No additional requirements.	
	[4.4.1.1] Data selection	DNVGL-CP-0484 App.B [5.2.3.5] Evidence shall be provided that the data is representative. See also [4.5.4.1].	Evidence that the data contains sufficient cases of the equipment failure mode or equipment degradation state of interest; that the data spans sufficient variation in i) time (for equipment known to exhibit seasonal variation), ii) locations and environments iii) operating patterns (vessel- or operator-specific) and/or iv) maintenance tasks and/or intervals.
	[4.4.2.1] Missing records	DNVGL-CP-0484 App.B [5.2.4.1] No additional requirements.	The number of missing records in main data set is N (p % of total). These are ignored.
	[4.4.2.2] Missing fields	DNVGL-CP-0484 App.B [5.2.4.1] No additional requirements.	Table of data fields, number of missing values, how interpolated / imputed
	[4.4.2.3] Data cleaning	DNVGL-CP-0484 App.B [5.2.4.1] No additional requirements.	
	[4.4.2.4] Data consistency	DNVGL-CP-0484 App.B [5.2.4.1] No additional requirements.	

Aspect	*Claim and topic*	*Requirement reference (DNVGL-CP-0484 App.B [5]) and details*	*Typical evidence (provided by the application owner)*
Data basis	[4.4.3.1] Data sufficiency	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	
	[4.4.4.1] Data format	DNVGL-CP-0484 App.B [5.2.4.1] No additional requirements.	
Development choices	[4.2.4.2] Prediction target	DNVGL-CP-0484 App.B [5.2.3.1], DNVGL-CP-0484 App.B [5.2.3.2], DNVGL-CP-0484 App.B [5.2.3.4] The prediction target of the application shall be explicitly defined. The target may be a condition/state of an equipment failure mode or prognostics to provide a quantified expected RUL or ETTF with a satisfactory level of confidence.	A prediction target may be a condition, abnormality or parameter limit associated with an alert or alarm. It may also be more specific stating: predict ETTF at least 1 week before failure will happen.
	[4.2.4.3] Technical success criteria	DNVGL-CP-0484 App.B [5.2.3.6], DNVGL-CP-0484 App.B [5.2.3.7] Acceptance criteria for the model shall be explicitly defined.	Acceptance criteria may be specified as a probability of detection to be predict 95% of equipment failure with a false alarm rate < 1%.
	[4.5.1.1] Modelling technique	DNVGL-CP-0484 App.B [5.2.3.4] No additional requirements.	
	[4.5.1.2] Modelling assumptions	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	
	[4.5.1.3] Model transparency	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	
	[4.5.1.4] Model baseline	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	The baseline may be another (simple) data-driven model, but may also other type of model(s) such as physics-based, knowledge-based or from inspection and testing (condition assessment).
	[4.5.2.1] Test design	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	

Aspect	*Claim and topic*	*Requirement reference (DNVGL-CP-0484 App.B [5]) and details*	*Typical evidence (provided by the application owner)*
Development choices	[4.5.3.1] Training documentation	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	
	[4.5.3.2] Configuration management	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	
	[4.5.3.3] Data partitioning	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	
	[4.5.3.4] Data quality requirements	DNVGL-CP-0484 App.B [5.2.4.1] No additional requirements.	
	[4.5.3.6] Model documentation	DNVGL-CP-0484 App.B [5.2.3.4] No additional requirements.	
Performance	[4.5.3.5] **Model performance**	DNVGL-CP-0484 App.B [5.2.3.8] No additional requirements.	
	[4.5.3.7] Model robustness	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	
	[4.5.4.1] **Representative test data**	DNVGL-CP-0484 App.B [5.2.3.5] Test data shall represent the scenarios (limitations and operation conditions) under which the model will be used.	Data set covers a set of N machines operating on M vessels. Operation is known to exhibit seasonal variation, and data records for each machine span at least 2 years.
	[4.5.4.2] **Independent test data**	DNVGL-CP-0484 App.B [5.2.3.5] It shall be documented that the test data is independent of the training data. Test data shall be composed of independent samples from different experiments, rather than multiple samples from the same experiment, to ensure a reliable indication of the expected performance of the machine/system in real conditions.	Training data is taken from years 1-5, test data from year 6. Training data spans 20 machines on 10 vessels, test data is from different vessels, and spans 4 machines on 2 vessels.
	[4.5.4.3] Test data characteristics	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	

Aspect	*Claim and topic*	*Requirement reference (DNVGL-CP-0484 App.B [5]) and details*	*Typical evidence (provided by the application owner)*
Performance	[4.5.4.4] Assessment	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	
	[4.5.4.5] Model baseline	DNVGL-CP-0484 App.B [5.2.3.5] It shall be documented that the model is better than the baseline.	The model has classification accuracy of 99.5% compared to baseline model accuracy of 98%. Model precision is 99% compared to baseline model precision 97%.
	[4.6.1.1] **Application assessment**	DNVGL-CP-0484 App.B [5.2.3.6], DNVGL-CP-0484 App.B [5.2.3.7] It shall be documented that the application satisfies the acceptance criteria for both performance and risk defined under [4.2.3.13]. The confidence level (DNVGL-CP-0484) of the application shall be compared to the applicable risk level of the equipment failure mode that the application aims to predict/prevent.	Report describing application performance with reference to requirements.
	[4.6.1.2] Application test is realistic	DNVGL-CP-0484 App.B [5.2.3.5] No additional requirements.	
	[4.6.1.3] **Application robustness**	[DNVGL-CP-0484 App.B [5.2.3.5] It shall be documented how well the application responds to data quality issues (including missing sensor records, missing values from sensors, incorrect sensor data).	Report showing model performance under a set of simulated data quality problems.
	[4.6.1.4] Critical applications	DNVGL-CP-0484 App.B [5.2.3.1], DNVGL-CP-0484 App.B [5.2.3.2], DNVGL-CP-0484 App.B [5.2.3.4] No additional requirements.	
Deployment	[4.7.1.1] Deployment plan	No additional requirements.	
	[4.7.2.1] Maintenance plan	No additional requirements.	

Aspect	*Claim and topic*	*Requirement reference (DNVGL-CP-0484 App.B [5]) and details*	*Typical evidence (provided by the application owner)*
Deployment	[4.7.3.1] Assumptions in operation	DNVGL-CP-0484 App.B [5.2.3.4] No additional requirements.	
	[4.7.3.2] Assumptions in operation are valid	DNVGL-CP-0484 App.B [5.2.3.4] No additional requirements.	
	[4.7.3.3] Contingency / redundancy documented	DNVGL-CP-0484 App.B [5.2.3.1], DNVGL-CP-0484 App.B [5.2.3.2]and DNVGL-CP-0484 App.B [5.2.3.4] Contingency / redundancy / fallback mechanisms used in operations to handle wrong predictions or other application failure modes shall by documented.	RCM analysis documenting each application failure mode, its failure mode risk level and the countermeasures / maintenance tasks used to handle the failure. See also [4.5.1.4].
	[4.7.4.1] Monitoring of assumption validity	DNVGL-CP-0484 App.B [5.2.3.4] No additional requirements.	
	[4.7.4.2] Monitoring of data quality	DNVGL-CP-0484 App.B [5.2.4] No additional requirements.	
	[4.7.4.3] Monitoring of performance	DNVGL-CP-0484 App.B [5.2.3.4], DNVGL-CP-0484 App.B [5.2.3.6] No additional requirements.	
	[4.7.4.4] Handling of performance problems in operation	DNVGL-CP-0484 App.B [5.2.3.1], DNVGL-CP-0484 App.B [5.2.3.2]and DNVGL-CP-0484 App.B [5.2.3.4] No additional requirements.	
	[4.7.4.5] Triggers for retraining in operation	DNVGL-CP-0484 App.B [5.2.3.4] No additional requirements.	

C.2 Terminology differences

Most predictive maintenance applications fall under the umbrella of anomaly detection: in anomaly detection problems there is a large quantity of data from the normal or no-fail state, but a relatively small (or zero) quantity of data from the fail state. Applications for supervised learning in such scenarios focus on characterizing the normal state, and detecting deviations from that normal state. As the target of such applications is to predict which of the states fail / no-fail the system is in, this is a classification problem, and performance of classifiers for anomaly detection can be measured using the same metrics as for classifiers: for example classifier accuracy, classifier precision and recall.

Guidance note:

Note that there is a potential point of terminology confusion here: accuracy and precision are used both as performance metrics for classifiers and as performance metrics for measurements. For classifiers, accuracy is defined as the proportion of test cases that are correctly predicted (both true negatives and true positives), and precision is defined as the proportion of positives that are true positives. In standards relating to measurement (e.g. /10/), a measurement has high accuracy if it has both high trueness (the measurement average is close to the real value) and high precision (the standard deviation of the measurements is sufficiently small). In the current RP the performance metrics for classifiers are referred to as classifier accuracy and classifier precision to avoid any such confusion.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

Guidance note:

A further potential terminology confusion arises with the use of baseline. In data science / machine learning terminology, as used in this RP, a baseline model is a simple, interpretable model, and comparison of model performance to baseline performance provides a simple check that the model is doing something useful. See [4.5.1.4] for more information. In predictive maintenance a baseline is often understood to be an existing model, or a physics-based model. But in both fields, a baseline provides a reference which the model under development shall improve upon.

---e-n-d---o-f---g-u-i-d-a-n-c-e---n-o-t-e---

CHANGES – HISTORIC

There are currently no historical changes for this document.

About DNV GL
DNV GL is a global quality assurance and risk management company. Driven by our purpose of safeguarding life, property and the environment, we enable our customers to advance the safety and sustainability of their business. We provide classification, technical assurance, software and independent expert advisory services to the maritime, oil & gas, power and renewables industries. We also provide certification, supply chain and data management services to customers across a wide range of industries. Operating in more than 100 countries, our experts are dedicated to helping customers make the world safer, smarter and greener.

SAFER, SMARTER, GREENER