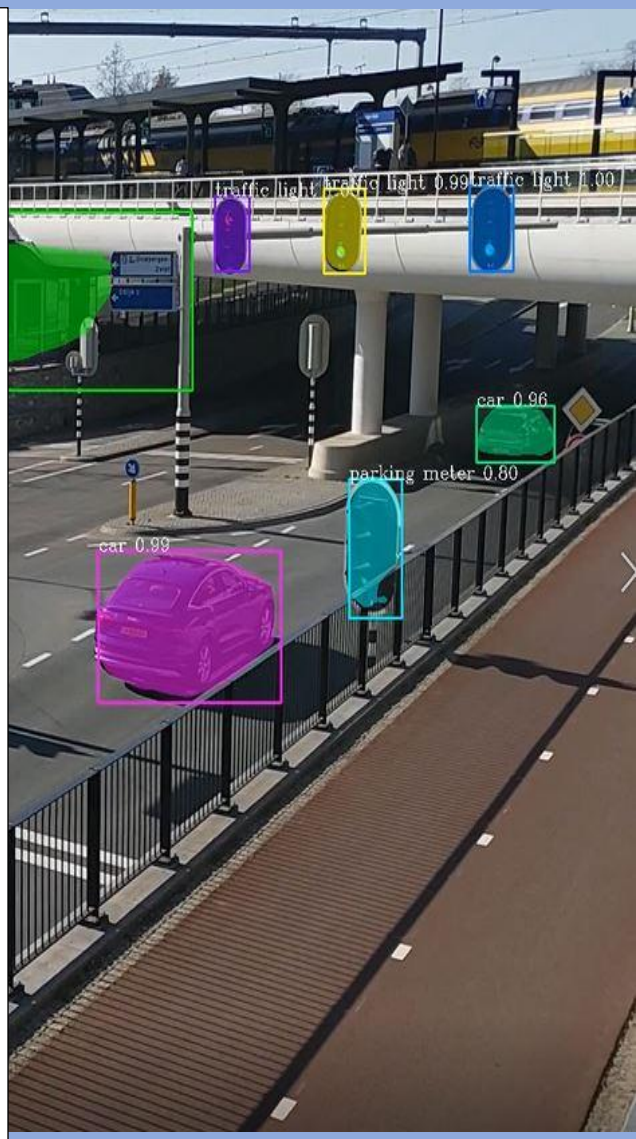
Quality and Testing

of

Artificial Intelligence



September 2021

Sander Mol
Peter Collewijn
Hannie van Kooten

Working group Testing and AI

TEST NET

Version: September 2021

Written by Sander Mol and Peter Collewijn, edited by Hannie van Kooten, with contributions from Rik Marselis and Mariëlle van der Sluys and the cooperation of all fellow members of the working group Testing and AI: Bram van den Reijen, Gerline van Lieburg, Johannes Sim, Marco Verhoeven, Mariëlle van der Sluys, Martin van Helden and Richard van Emmerik.

We would like to thank these members and also Aleksandr Novolokov, Artur Donaldson, Ellie Williams, Evelyn Bongers, Fatemeh Ramezani, Joanne Boerstoel, Kimberly Bittler, Kristy Parhiala, Reinier Zwiep and Ufuk Anil Calimli for their reviews and suggestions for the English version.

Comments and suggestions for improvement on this document are appreciated. Send these responses to the working group by mail: ai.workgroup.testnet@gmail.com.

The working group would like to get in touch with other groups or organizations involved with this topic. Do you know such a group or are you part of such a group and are you also looking for this collaboration, please contact us via email: ai.workgroup.testnet@gmail.com.

Introduction

Working group on Testing and AI

This white paper was written by the working group on Testing and AI, part of the Dutch testing community TestNet. Our group was founded in January 2018 and initially focused on testing using artificial intelligence (AI). After the publication of the white paper on this topic in 2019, we moved on to the topic “testing of AI”. The working group regularly gives presentations on the subject of testing with or of AI at TestNet conferences or elsewhere.

By writing this white paper, we tried to give structure to the many ideas that the working group members had about Quality and Testing of AI. With a subject that is still in its infancy, there are no beaten paths or best practices. We have gradually come to the current structure and it has also been a voyage of discovery for us.

Purpose of this white paper

There are two main goals of this paper. The first is to give fellow testers enough information and confidence to recognize the risks associated with an AI implementation, enabling them to shape the AI test process with their own knowledge and skills.

The second goal is to stimulate further general developments in AI quality and testing. These developments could be the long-term development of best practices and the development of new methods of testing where necessary. We would therefore like to invite readers to share experiences and respond to what we have written. Contact details are on the last page.

Resources

In recent years, several books have been written about the risks and testing of AI and several courses have been developed. Additionally, guidelines have been drawn up and/or are being developed by various governments, sector organizations and other partnerships. The source list can be found in the appendix. We discussed these sources within the working group and then translated them into this white paper.

We realize that this publication is a snapshot. Nevertheless, we are convinced that this white paper is a good starting point for AI testing, with insights that will be useful in the testing profession for a long time to come.

Summary and reading guide

This paper is divided into eight chapters. The topics of these are described below. The paper also contains appendices which include further in-depth descriptions and explanations, sources of material and the glossary. The glossary can be referred to for explanations on specific terminology used in the text.

Chapter 1 deals with algorithms. About algorithms to calculate the price of a house in 'ordinary' software and about Machine Learning algorithms. What makes them so different and what exactly is Machine Learning?

Chapter 2 discusses the risks of Machine Learning. We have chosen to identify, in our view, the most important five general risks. There are many more, but we have limited ourselves to key risks, like dependence on data and limited explainability.

Chapter 3 discusses the different categories of Machine Learning, which we have called “appearances of AI”. These appearances are, for example, image recognition and speech generation. For each appearance we discuss the risks that are the most typical or the most clarifying. Most of the general risks discussed in chapter 2 apply to all the appearances to some extent. Whether these risks are large or small varies for each individual application.

In chapter 4 we took the degree of autonomy of the AI application as a starting point. This can vary considerably. On a more basic side, the ML application can edit pre-processed information, after which you personally review the edits made. On the more autonomous side, the ML application can obtain information completely independently and make independent decisions, such as a self-driving car.

Chapter 5 focuses on ethics. This is an interesting topic that is directly related to the testing of the AI application, discussing questions such as: what regulations are being developed in the EU to achieve robust and secure AI? Which aspects, such as justice, privacy, fairness, and transparency are important?

Chapter 6 discusses the non-functional aspects of a Machine Learning application in more detail. These aspects are currently described in the ISO 25010 standard for software product quality. However, these appear to be too limited. Therefore, this chapter discusses several initiatives to add new features and attributes specifically for ML supported software.

In chapter 7 we arrive at AI testing. How do we achieve high-quality AI and which tools can we use for this? In this chapter, many well-known test methods such as A/B testing are discussed alongside lesser-known ones such as Metamorphic testing.

Chapter 8 looks at testing in practice. This chapter discusses what roles there are within an AI project and who takes responsibility for the quality. It also lists the necessary skills and knowledge areas for testers who want to develop further in the world of AI.

Finally, we repeat our call to share knowledge and work together on to develop best practices and to strengthen the positioning of testers in an AI team.

Table of contents

1. Artificial Intelligence

The terms Artificial Intelligence, Machine Learning and Deep Learning are often used interchangeably. This paper also uses the term 'AI' for recognizability, while we mainly focus on the subcategories Machine Learning and Deep Learning. This chapter indicates the differences, but also the interdependence of the three concepts.

1.1 What is Artificial Intelligence (AI)

AI or Artificial Intelligence is a container term for everything that we consider intelligent and originates from a computer or machine instead of a human being. Artificial Intelligence has become a buzzword in the last ten years. Concepts such as Machine Learning (ML), Unsupervised and Supervised Learning, Deep Learning (DL) are regularly used, and the news often talks about 'algorithms'.

Artificial Intelligence plays an increasingly important role in our daily lives. The AI component of Machine Learning in particular has taken off, because the computing power of computers and the amount of data available has increased drastically, while the costs for this computing power and accessing data have fallen drastically. The technology is now within reach for everyone who is interested and the number of applications will continue to increase sharply in the coming years.

1.2 What is Machine Learning (ML)

In traditionally programmed software we find intelligence in the form of programmed rules that we humans find logical. For this we use terms such as decision rules, formulas, or algorithms. An example of these types of rules is the calculation of a predicted house price:

The predicted house price in euros = 700 * living space +
500 * plot area +
8,000 * number of bedrooms +
score based on the zip code

Then we can easily calculate the predicted price of a new house, as long as we know the input variables of this house.

An important starting point for programmed rules is that they are well thought out in advance, in this case by people who know the housing market very well. We also know that if we have any doubts about whether this rule is (still) correct, we can turn to these experts. There is therefore a fixed way to determine whether a test has passed, also known as a 'test oracle'.

This is different with ML algorithms. The starting point isn't a well-thought-out rule, but a set of collected examples. These examples consist of input variables and an associated output variable, which together lead to an algorithm or a model. In an image it looks like this:

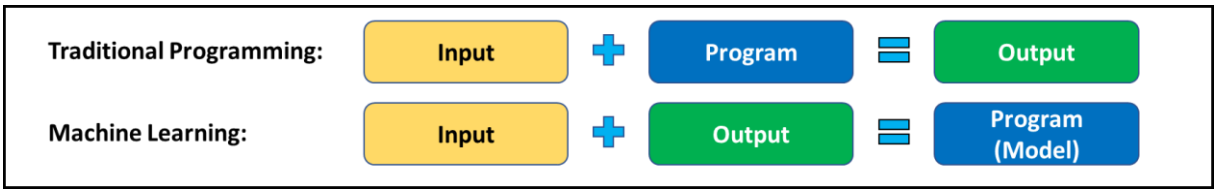


Figure 1: Traditional Programming - Machine Learning

In house prices, these could be the first three examples:

Living area	Plot area	Bedrooms	Zip code	Price
110	173	4	3311 AA	400.000
104	145	6	3351 ES	375.000
122	211	5	3352 VA	450.000
...	...	...	...	...

Table 1: Characteristics of houses - house price

The ML tool is then instructed to devise the best possible algorithm for predicting the price, also for new houses. Again, the input variables are the basis for the price prediction. In an image it looks like this: The extent to which the input variables are taken into account is still unknown. We show them as variables, w (weight) 1 to 4.

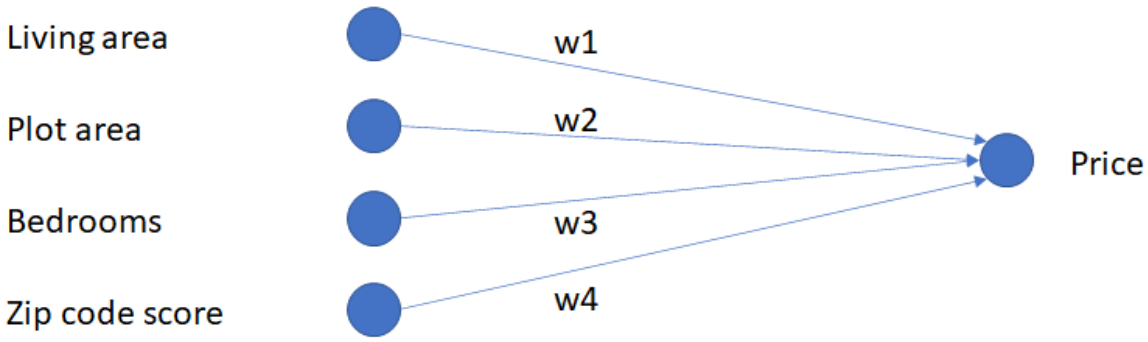


Figure 2: Weights of properties of houses for house price determination

Then we come to the 'magic' of ML, the independent search for the correct weights. ML uses algorithms based on statistical techniques to determine these weights. Again, a kind of formula (or algorithm or model) is created similar to the one shown at the beginning of this chapter. It is important to realize that the factors (or weights) have not been established based on human expertise. It is also possible that by applying a different algorithm, a different result is obtained. However, it can never be fully determined whether the result is completely correct; the 'test oracle' is missing.

Yet, often ML will offer sufficient certainty to be applied in practice. The ease of using data to learn from and ultimately arrive at a production-worthy model is very attractive. Moreover, with ML it is possible to develop applications that would not be possible with programmed software, such as recognizing images or speech. In addition, it is possible to adapt or detail previously trained models to your own needs. For example, it is possible to adapt a model that has been trained to recognize cars to recognize certain makes or classes. This significantly reduces the time required to train the model.

1.3 What is Deep Learning (DL)

Often the predictive value of ML models is too limited if only the direct relationship between an input variable and the output variable is considered. The relationships between two input variables could also say something about the output variable. For example, a combination of a small living space and a large plot area could indicate a luxurious, detached villa and therefore a higher price.

We therefore need an extension of the previously used image to be able to find the connections just mentioned. Specifically for living space and plot area, this looks like the following:

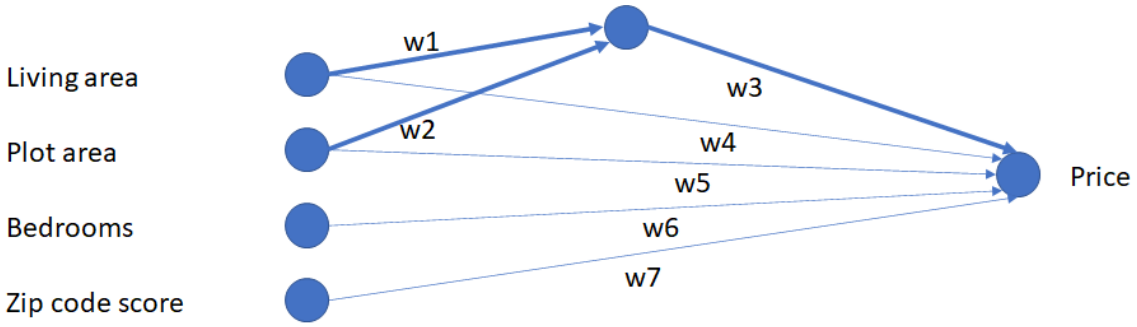


Figure 3: Weights of properties of houses for house pricing advanced

Instead of using an algorithm, it now starts randomly assigning values to the weights. The predicted price based on these weights will probably be completely wrong and the error with the actual outcome is large. The goal is to make this error as small as possible. The DL tool therefore adjusts the weights a little bit and determines whether the predicted price has become a little bit better or a little bit worse. In the case of an improvement, the new weights are used as a starting point for the next minor adjustment of these weights. This iterative process is repeated until the model has an accurate predictive value or no further improvement occurs.

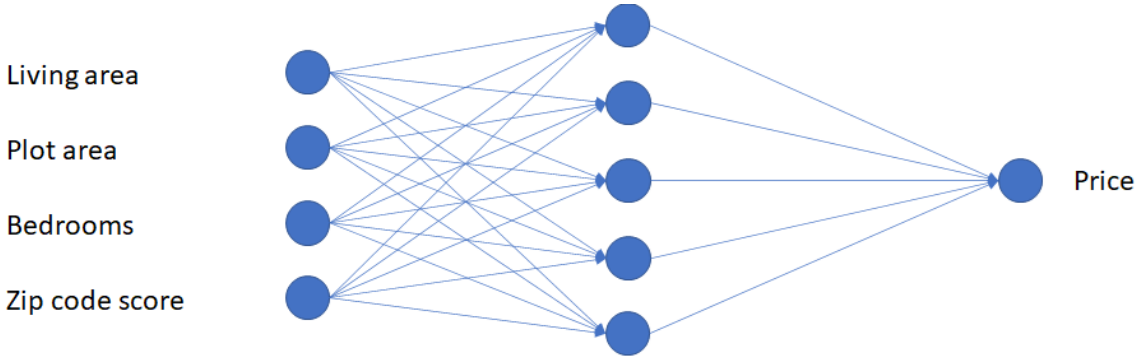


Image 4: Weights of properties of houses for house price determination with intermediate layer

To be able to make all of these combinations using the image above, a weight is required for all 25 lines. These weights are given an arbitrary value in advance and are slightly improved with each training round.

It is important to realize that the model above has been given an extra 'layer' of 5 nodes, where these nodes have no fixed meaning while the input variables do. Instead of 5, we could have chosen, for example, 3 or 20 nodes in this extra layer.

Even at 20 nodes, the model would still be very small. In practice, dozens of input variables, dozens of nodes per layer and dozens of layers are no exception. Because of this, this type of model is often called a neural network, because it is reminiscent of the construction of brains with neurons. This increases the number of weights that can be adjusted to arrive at a good prediction. The addition of all these nodes with no fixed meaning also reduces the explainability of the model.

1.4 AI and risk-based testing

When testing software, we as testers have been looking at product risks for decades to determine how much time we need for testing and what we want to spend that time on. Risks consist of the impact of a particular negative event and the probability of that event.

Of course, this also applies to software that has learned the rules through data-based training. If your face on the camera cannot be matched with the photo on your passport at the airport, you can still fly after a manual verification. This is a low impact. However, being recognized as a criminal may have a high impact. That depends on how one handles this result; you may have to go through security and miss your flight. In the worst case, you will be handcuffed, and you may spend the night in a cell.

This immediately shows that the risk of AI predictions can go both ways; an incorrect recognition (false positive) or incorrectly not recognizing something (false negative). In both cases it is good to think about the impact beforehand.

Risk-based testing therefore remains important. We have noticed, however, that the risks of AI-supported software generally look different than with programmed software. The next chapter deals with these risks.

2. The general risks of AI

Every software application has its own specific risks. Therefore, it is always necessary to consider these risks when developing software. As a TestNet working group, we believe there are general risks that apply exclusively or relatively often to AI-supported software. By focusing on these risks, a faster and more complete risk analysis of AI-enabled software can be conducted.

In this chapter we elaborate on the most important general risks.

2.1 Uncertain outcomes

When calculating house prices it is relatively easy to determine the expected outcomes with traditional, rule-driven, programmed algorithms. However, the use of ML leads to an application of which we do not know the exact rules in advance. Here are a few more examples:

- Based on the text the customer has typed in the chat, the algorithm predicts that he wants to report damage to his car.
- Based on the pixels of a particular image, the algorithm predicts that there will be a dog on this image.

In all these cases, the algorithm does not give an absolutely certain answer, but an answer with a certain degree of certainty or accuracy. For example, in the image with the dog, the algorithm predicts a dog with 97% certainty, but also a wolf with 82% certainty and a cat with 56% certainty. The question that must be asked for any software application is to what extent you want to build on these uncertain results. With the image of the dog, is it enough to recognize it with 90% certainty? And what if both the dog and the wolf are recognized with 95% certainty in one image? And if we want to recognize a total of 100 animals, how do we know that we are correctly handling all possible combinations of correct and incorrect predictions?

These are all uncertainties that contribute to the risk. When predicting animals, this is still fairly harmless, but when predicting diseases, it quickly has a high impact, both with false positives (care costs and incorrect treatment) and false negatives (lack of treatment). There are many more examples where the risk of uncertain outcomes is very high.

Because this topic is so important and may seem complex, we have included a detailed explanation of accuracy percentages in Appendix C.

2.2 Dependency on data

As indicated in the previous chapter, ML is the part of AI that learns from collected data to make predictions about new data. Any AI application can only build models or make predictions based on the data that it was trained on. Therefore, the quality of this data is very important and there are many risks involved in collecting and using this data.

Below are several points that affect the quality of the data and thus ultimately the quality of the ML model.

- **Sources used for the data**. These sources must be representative and in proper proportion to reality. Which source is used to determine a house; the land registry, the municipality, or the

price of a house that has just been sold in the same street? Or a combination of these sources?

- **Selection of the data**. It is impossible to use all available data and so filtering will take place. This creates the risk that the dataset is chosen too broad or too narrow or becomes biased. The surface area and the number of rooms are certainly important when determining a house price, but data on whether the garden has grass or a stone terrace may not always be accurate, even though it could improve the model.
- **The purpose of the data**. Data may have been collected for a purpose other than training ML models. Data may be missing because it was not needed for the original purpose or because it was not checked during collection. Inaccuracy may also have occurred due to the omission of the unit of measure.
- **Completeness of the data**. The way in which the data is collected. Are all fields correctly filled in the same way, do these fields also exist in all data sources? With a web form it is possible to make fields mandatory or to provide them with fixed values by means of selection fields. With mail, chat and telephone calls this is not possible. Remember that the collected data is only a subset of reality. There is a very real chance that the collected data does not represent reality well and even that some real-life cases are not available to train a model.
- **Clarity of the data**. It is possible that fields contain no value or multiple values that could mean the same thing. For example, Gender: Male and M or Woman and W. A choice will be made to provide these fields with unambiguous content through data conditioning. For example, the gender field could be conditioned to only contain the value M and F for analysis. Missing values could be provided with a default value, interpolated based on the most similar data on other metrics, or an average.
- **Personal preferences in data collection.** Data collections can vary depending on the person who collects it, or who sets up the collection. The question and answer options can lead respondents to certain outcomes. The recording can be selective, both when entering records into a database and when creating and collecting images, texts, etcetera. These systemic biases in the underlying data can also be reflected in any model built, resulting in a model that does not accurately reflect reality, but rather our data collection method instead.
- **Time of data collection**. The moment of collecting, including date and time are important. Does data collection only take place at a certain time of the day, or are only working hours included? Trends observed may be related to the time of the week or to an event. For example, a peak of service desk calls after the weekend or after the launch of a software update. Another important consideration is time related autocorrelation. For example, data taken a few seconds apart are likely more similar than data taken several years apart. When data is closely related in time (or space), it can make models appear to be more accurate than they truly are. A third consideration is the extent to which historical data reflects the same conditions that you are modeling now. For example, how far does one go back in time for the house price; a month, a year or five years?
- **Format and source of the data.** The integration of the data sources. Is the format of the source data the same? A date field can be day-month-year in one system and year-month-day in another system. There is also a chance that in numbers the notations of the dot and the comma deviate (for example in the USA decimals are delineated with a period, while in Europe they are delineated with a comma). In addition, data may differ in proportion or unit.

Suppose one house price system works with length and width in centimeters and another with meters, that is a factor of 100 difference. The frequency or timespan can also make a difference. One system reports the number of sales per hour and the other system the number of sales per day. This must be considered when integrating this data.

- **Labeling data.** Giving a label to an object seems simple, but in practice it is often a challenge. See the example below on labeling cats. Mislabeled data creates misinformation. Labeling is a subjective process. For example, the condition of a house can be bad, mediocre, good or excellent. But what is the difference between good and excellent? This will vary from person to person.
- **Outliers in data and bandwidth.** Outliers are values that are outside the usual bandwidth and do not follow the expected pattern of a given data set. For example, a house with 12 rooms, while all other houses have 2, 3, 4 or 5 rooms. If these outliers do not occur regularly and do not contribute to the result of the model, they can be removed. In practice it is good to look at the cause of these outliers in the data because they can be the result of special cases such as fraud, hacking attacks or malfunctions. If the cause is not removed, recurrence of the outliers can also occur in the future. It is good to build in a check on the input side of a model to detect and avoid including these outliers in the model. The result can be that a data variable is given a certain bandwidth, both on the input and prediction side. If a bandwidth is used when compiling the dataset for training the model, this bandwidth also applies when using the model in the production environment.

Despite the broad enumeration, this list is not yet exhaustive. This again shows the dependence on data and the associated risks on the final model.

How selective the collection can be is nicely illustrated by Cassy Kozyrkov's cat selection, where people can indicate whether they think it is a cat or not.

Figure 5: Selection of cats

The first five images are easy to describe, but in the sixth example we need to tighten up our definition of 'cat'. Is it about felines, or is it just about domestic cats? This is a clarification that is not always done, especially in practice where the differences are even less clear.

In addition to completeness and correctness of data, the quality is also influenced by the selection and interpretation of that data. Low data quality means that we draw varying or downright wrong conclusions via ML. This means that the results of the ML model will not match reality very well when used in production, limiting the practical usefulness.

2.3 Limited explainability

There are several ways to perform Machine Learning. A well-known variant is a decision tree (see the left side of the image below). The decision tree can be followed from the top (the trunk) down for each example, while with each step down a choice is made based on the characteristics of the example. For example: people over 36 years old take the left path, people 36 years old or younger take the right path. The AI model tries to distinguish between two or more outcomes that we want to predict on the basis of the available input variables. This tree can be constructed with ML. Again, the model will learn by making a small adjustment each time and assessing whether this gives a better or worse prediction. The model that gives the best result after all small adjustments is used to make a prediction for new situations.

The advantage of this decision tree is that we can indicate why a certain prediction is made in all trained and new situations. After all, you can follow the decision tree from top to bottom until you get to the prediction.

The right side of the image below shows the Neural Network approach, which is most used in AI applications. These models are particularly complex, as many combinations of data and structures of analysis are explored and selected by the algorithm. And although they can give very accurate predictions, we as humans can hardly or no longer explain how the prediction comes about due to higher complexity.

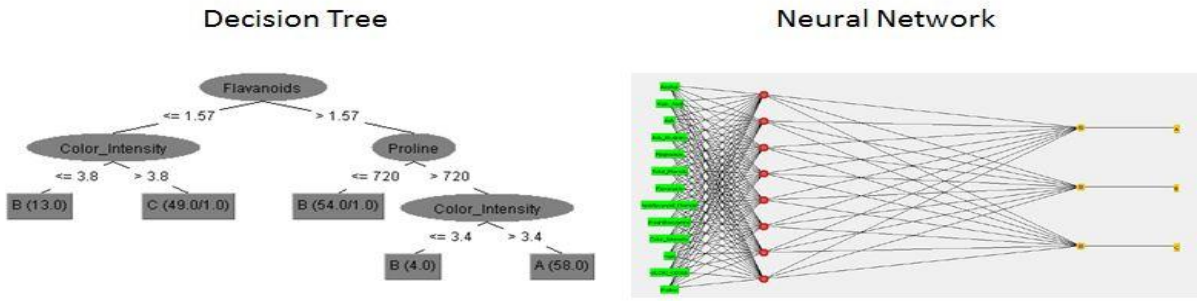


Figure 6: Decision Tree - Neural Network

One of the techniques where a prediction via Deep Learning (DL) can sometimes be explained is in the recognition of images. The neural network below was trained to distinguish standard passenger cars from off-road rally cars driving in the Paris - Dakar race. Even though the model had an accuracy of 95% on the test images, the analysis showed that the model mainly relied on the background; sand from the desert or no sand.

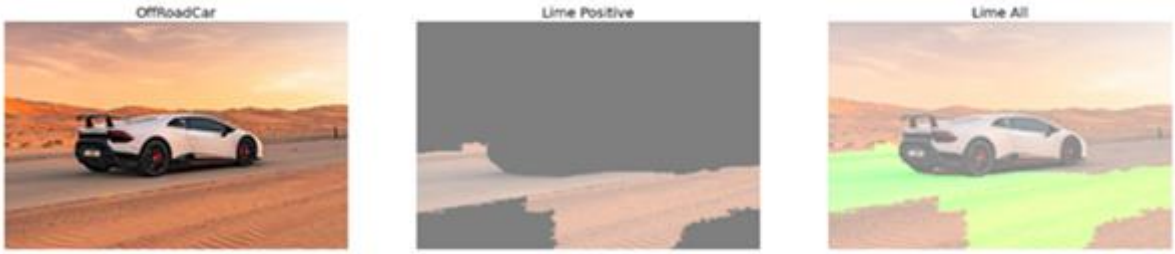


Figure 7: Erroneous recognition of a car based on the environment

A person would have looked at the car itself for his or her prediction, for example at the wheels and not at the background as in the above example. So, we are quick to think that a machine uses the same reasoning, with a significant risk that this assumption is incorrect. It is therefore important that the way in which an ML model arrives at a prediction is transparent to be able to assess this 'reasoning'.

2.4 Changing reality or need

Software development is never completed. There are always new requirements, user requests or insights that need to be responded to. IT solutions with an AI component are no exception. Regular adjustment is especially important in ML applications, where the data is the source.

Recall the example of house prices; these change continuously and sometimes very strongly. The chance that the model is still usable a day after it has been trained is quite high. The chance that this will still be the case after a month is reduced and after a year the chance is almost negligible. There will have to be a chosen moment to train a new model.

Furthermore, a conscious trade-off must be made in the amount of history used to train the model. There is no point in retraining the house price model with data from a year ago. Ideally, you should only use the most current data, for example that of the past week. But then there is a real chance that insufficient or only limited representative data is available.

However, if you want to use the history of the past month, then there is a chance that not all data was collected in the same way. Perhaps it was agreed last week that basements should or should no longer be included in the number of bedrooms. How do you deal with those differences? And this is just one example that can still be solved unequivocally, provided it is known which houses have a basement. It is also possible that the way data is collected has a major impact on the usability of historical data, or that it is not even known that the data has been collected differently. As an example, we may have recently completely adjusted the zip code score, or the basements have been removed or added to the count without the project team's knowledge.

It requires vigilance, effort and creativity to continue to connect reality with an AI model. There is certainly a risk that this will receive little or no attention while the value of the model degrades over time.

2.5 General fear of AI

New technological developments are usually received with some mistrust. That was the case with the advent of computers, the advent of the Internet, and so on. This is understandable. For example, who could have predicted what the impact of computers and the internet would be on society? The possible impact of developments such as blockchain, internet of things, quantum computing and of course AI cannot yet be foreseen.

Part of the general fear of AI comes from people not knowing what AI is and what it can do. Stories about superintelligence, in which computers are smarter than humans, add to this. However, the current development of ML is separate from self-thinking robots. Formulating actual goals, then devising all the steps towards that goal and executing them in the right context are impossible with current ML techniques. However, it is what the term Artificial Intelligence suggests. The prejudices surrounding AI will therefore continue to exist for a long time to come, especially because the technology behind ML is also difficult to explain to everyone.

Another way people fear AI is based on experience with how AI can be misused or lead to unforeseen negative consequences. There is still plenty of experimentation with AI, and things still regularly go wrong in this phase of technology development. There are countless examples of this; chatbots learning abusive language, cops being sent to the same neighborhood over and over based on expected crime, men being selected as managers more often than women, and so on.

Finally, there is also a fear of the unlimited use of data to reach harmful conclusions. Personal data is collected everywhere, and some organizations may use this data for applications that do not benefit individuals or societies. For example, using browsing history to raise prices when shopping online. This is especially true for government organizations, where you as a citizen have no choice to hand over your data.

As a nuance, the context in which an AI application appears is a major factor. If AI is used in a well-known place, it's a well-known application and you can freely choose to use it or not, it's usually not a problem. For example, think of your Apple Carplay or your (face editing) applications on your tablet. Risk assessments remain tailor-made.

2.6 New challenges for testers

The general risks in this chapter show that it is important to be careful when implementing or considering using AI. Testers are used to thinking in terms of risks. Since the arrival of the first computer programs, testing has grown into a fully-fledged profession. With the advent of AI, the testing profession will change again.

However, the development of AI models is also continuing at a rapid pace. This development is largely done through practical applications. It is becoming increasingly easier to train a model. A model can already be trained with 10 lines of code. Techniques such as AutoML make this even more accessible; the training application itself looks for the AI model and the associated parameters that probably best fit the data.

The gap between coming up with an idea for an AI application and its actual implementation is therefore many times smaller than with a rule-driven application. Still, all quality and testing activities are also required for AI applications. The testing of these kind of applications cannot be left behind. Not only does there remain a need to test the functionality, but other aspects such as ethics with explainability, accountability and non-discrimination are also important. If this does not grow, trust in AI applications will quickly disappear as soon as users or general opinion are regularly confronted with mistakes. It can be argued that for the further growth and acceptance of AI, further growth in the testing of AI applications is necessary.

3. Appearances of AI applications

This chapter discusses the different appearances of AI, such as image recognition and speech generation. A selection has been made of the risks from the previous chapter. Not all risks are discussed with every appearance, although almost all of them apply. Some risks appeal more to the imagination or are easier to explain in one appearance type than in another. The selection is based on that.

3.1 Pattern recognition in datasets

Companies and governments are increasingly using pattern recognition in datasets. Over the years they have built up large datasets with a lot of information about their customers or citizens. With the arrival of AI, they see the opportunity to derive added value from this.

The most visible result of pattern recognition is interactions with customers that predict which product the customer would like based on the customer's previous purchases and interests. AI is also used to find social media posts that the organization would like to respond to, or to recognize which customer has the greatest chance of leaving. In addition, there are even more internal processes for which AI is used. Think, for example, of assessing a credit application, detecting fraud, predicting required maintenance on vehicles or simply categorizing emails based on content.

Recognizing patterns in large data sets is so complex that this is no longer humanly possible. With the help of ML it is possible to visualize these patterns and to predict a result or category. This leads to new useful information, but has a downside, because it is difficult to explain how this prediction came about.

This limited explainability may have consequences. For example, if a person is accused of fraud while the reason for it cannot be explained, that may not be legally accepted.

> *Example*
>
> Limited explainability was a reason for the court to ban the Dutch SyRI system. The SyRI system was an initiative of the Ministry of Social Affairs and was used in the cities of Rotterdam, Eindhoven, Haarlem and Capelle aan den IJssel. Its purpose was to prevent fraud in taxes, allowances and benefits. A large amount of data was shared and analyzed in order to subsequently make risk reports. These are reports linking someone to possible fraud.
>
> The Dutch court found that the state should use new technological possibilities to prevent and fight fraud, but that there should be a right balance between the benefits and the right to respect for private life. How data is processed and analyzed is not transparent, the court said. This created a risk of "unintentionally discriminatory and/or stigmatizing effects".

The general risks from the previous chapter play a role in pattern recognition in datasets. Due to the size of the datasets, there is a great deal of dependence on this data, there is a great chance that not all data were collected under the same circumstances (since the reality is changing) and there is limited explainability. Moreover, this data is used for training AI models, while it was not collected for that purpose. Customers may find this objectionable, especially if their data is used for a prediction that is detrimental to them.

3.2 Image Recognition

Image recognition is an appearance that most appeals to the imagination. This creates a great deal of creativity in devising applications. Everyone knows the usual examples by now, such as recognizing your face to unlock the phone, recognizing objects by self-driving cars, and recognizing abnormalities and diseases on medical photos. However, the creativity extends much further, recognizing the amount of garbage bags along the road (for more efficient collection), mapping the number of potholes in the road and their size and depth, determining the condition of fruit, signaling dangerous situations on a bridge and assessing damage to cars, homes and greenhouses.

Image recognition can be regarded as a specialization of pattern recognition. After all, it concerns a pattern of pixels that make up the image. In recent years, specific techniques have been developed for image recognition. One of these techniques is called Convolutional Neural Network (CNN), the operation of which can be seen in the image below.

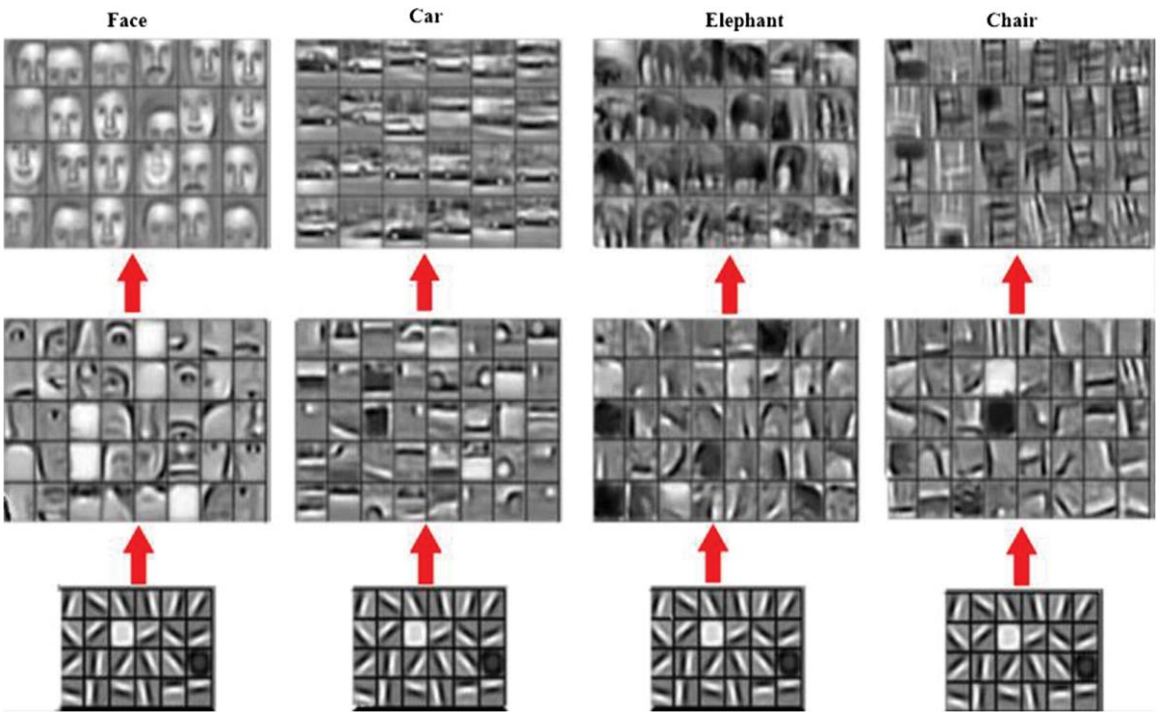


Figure 8: The layers of an image recognition model

The CNN first looks for basic shapes and basic colors and combines these basic shapes into increasingly complex shapes, until the complex shapes together predict a result like a cat or a car.

The image shows well that the simplest shapes only need to be trained once and can be reused again and again. Models for recognizing common objects such as human, dog, cat, bicycle, and car are therefore easy to find, free to use and usually have an accuracy of above 90%.

For most image recognition applications, a general model with common objects is not sufficient. Specific examples must first be collected to achieve the desired goal. As a result, the recognition percentages can be a lot lower, especially if you have just started training.

In addition, investments must be made to match each sample image to the correct outcome. This may affect the final use of the model in practice, depending on the impact of a misjudgment. Recognizing a garbage bag too much or too little is at most annoying, recognizing a disease too much or too little can be fatal. That is why people in the medical sector are reluctant to use AI. During

training, the trained models proved to have a high accuracy, but in practice they are not often applied because the risk of failure is still too high.

Risks can also arise over time because of changes in understanding.

> *Example*
>
> Tesla cars are a prime example. They have advanced ML systems to enable self-driving and are equipped with cameras. To improve the ML models, Tesla has the option of collecting additional information via these cars, for example about specific traffic situations or traffic signs. However, in theory this could also be used to collect information from other objects or people. This possibility is a reason for the Chinese armed forces to ban Tesla cars from their military complexes.

The risk associated with this appearance depends enormously on the area of application.

When recognizing dangerous situations on a bridge, it is usually good to point out to the bridge operator that there is 'something' different from the expected image, whether it is a person or a garbage bag. In a self-driving car, this distinction is important again; it can be a reason for the car to brake independently on a busy road.

Depending on the application, the uncertainty in image recognition plays a limited or greater role. The uncertainty is increased because we humans also do not always have an unambiguous expectation, as we saw with the question whether a tiger is a cat. AI can be used to reveal these kinds of ambiguities during the training phase. In practice, however, you want to have the biggest doubts addressed and that will require a conscious effort.

3.3 Sequence Recognition

Sequence recognition involves patterns in which order is important. Sequences can take various forms.

An example that can be quickly visualized is analyzing patterns in transactions to find unusual patterns and possibly fraud. Or the order in which people watch videos, to suggest the most suitable next video. On a slightly larger scale, one can find a pattern of seasonal influences on, for example, agriculture.

Another, perhaps more subtle example, is analyzing text, video, speech, or other sounds. Text is a sequence of letters and words; video is a sequence of images while speech and sounds are sequences of sound waves.

In sequence recognition, determining the time window is important. This refers to the interval between different times to recognize the pattern. That can range from seconds to minutes or even days. This depends on the subject; with a phone call it will be in seconds rather than minutes.

When looking for sequence patterns, the context should always be considered. A person with a different native language may take longer to answer a call center employee's question. An ice cream seller will have smaller sized transactions than a car salesman. Patterns can also undergo temporary changes. Think, for example, of the consequences that the Covid pandemic has on patterns of spending behavior that banks might record. Many AI models will have been retrained to take this into account. It is therefore important to continuously monitor the data pattern on which the AI model is trained and to ensure that it corresponds to reality. This depends on the type of AI model; it is unlikely that image recognition software in cars will need to learn new road signs or new strollers every week.

3.4 Regression

This appearance can cause confusion in the testing world since testing for unexpected changes between software versions is also called regression testing. Regression as an ML appearance is a very well-known form, examples of which are the prediction of house prices or stock prices.

Many ML applications make a prediction for a certain value at a certain moment based on the discovered pattern. This prediction is not 100% certain. In contrast to the previous appearances where the outcome was predicted with a probability, is the outcome for this appearance a value with a range of uncertainty or bandwidth. For instance, an image of a cat is predicted with a probability of 90%, however for stock price predictions the value of a share is not predicted to be worth € 10.00 next week with a 90% probability, but with a bandwidth of let's say € 0.50. The smaller the bandwidth, the smaller the range of possible outcomes, the higher the probability of the predicted outcome.

As mentioned before, to arrive at a certain price, the ML model must be trained with data. The risks associated with data have been described in detail in the previous chapter. A new risk arises in this category, namely extrapolation. Suppose the house price model is based on houses with 2 and 4 rooms in the training data, can you predict a price for a house with 3 rooms? This could still be done based on a weighted average. When predicting a house with 6 rooms, the uncertainty about whether the prediction is correct increases.

The house price is a simple model. For models with more abstract data or many more variables, it is difficult for us as humans to determine what the correct predictions are and to what extent the result is based on extrapolation.

3.5 Text generation

Almost everyone comes across this appearance daily. It can be through the spell checker, a chatbot or when translating a text into another language. This white paper has also been published in Dutch and English with the help of Google Translate (and human translators). Even the Dutch reader runs a good chance that there are sentences generated by Google Translate. Experience has shown that translating a document from Dutch to English and then back again often results in more readable sentences.

> *Example*
>
> A current development is the "Generative Pre Trained Transformer 3" (GPT-3) model. This is a language model developed by the OpenAI organization and is capable of generating code and even poetry. The model consists of 175 billion parameters and 499 billion tokens were used for training. It has been estimated that this required 700 GB of memory and 3.14E23 FLOPS of computing power. In short, an average PC will take thousands of years to do this. The cost for this training is estimated at $4.6 million. This is also an example of the fact that there are only a limited number of parties that can perform this, both with regard to the investment and with regard to the required computer power. In addition, at this time, only Microsoft has been granted the exclusive license to use and source code.

However, formulating grammatically correct sentences is only part of the challenge. The application generating the texts will have to approach the user in a consistent manner. A distinction must be made whether it concerns a formal text, for example a text that must be fully legally correct, or an informal text, for example an email to a colleague. Depending on the type of conversation that arises, the application should be informative, calming or, on the contrary, activating. In short, the application must have its own personality and style to ensure smooth communication.

For example, a very enthusiastic customer should not be made less enthusiastic by cool official answers. It is important that a chatbot can recognize this and can advise to pass the conversation on to a call center employee or to end the conversation. Chatbots have made discriminatory comments or started swearing. This effect is not only bad for the image of the company concerned, but also for the acceptance of this technique in general.

3.6 Speech Generation

Speech generation has been in the works for decades. It has evolved from saying words to reading sentences aloud, allowing for as good a human conversation as possible. The speech quality is now almost the same as how a real person would pronounce it. It is therefore good to look at speech generation from an ethical standpoint.

It is usually not a problem if the user has chosen the source of the spoken text himself, for example with the reading function on websites or with a navigation system. Even if the source of the text comes from a familiar environment, there are hardly any objections. Think of Siri, Alexa, or Google Home. Or talking robots for users suffering from dementia. In all these cases, the user has deliberately chosen this application.

It becomes ethically more difficult if the user has not chosen it or is not even aware of the application of speech generation. A clear example is the demo of the Google Assistant that makes an appointment by phone at the hairdresser. As a side note, it must be mentioned that this demo uses a chain of AI applications, in which speech generation is only the end station.

This way, ML makes a significant step into the real world to the extent that it becomes unclear when and how we are dealing with an AI. The ethical question is whether this is really a step too far or whether we as a society have to get used to this technology. It is clear that we as humans at least want to know if an AI communicates with us. This can be done via an announcement and possible agreement from the person concerned, but perhaps also by making the quality of the speech a lot less human.

3.7 Image Generation

As testers, we have little experience with image generation. However, it is regularly in the news, especially in the form of 'deep fakes'; images and videos that appear lifelike but are generated by an AI model. It can cast doubt on the authenticity of these images and videos, or it can be used unnoticed to influence public opinion or individual thoughts and actions. In the latter case, you can speak of manipulation or blackmail. In that sense, there is great fear of image generation. There are still many ethical, legal, and social risks to address.

Nevertheless, this technique will slowly but surely find its way to regular, generally accepted applications. Think of applications for enhancing or coloring old photos or films. An overview of the environment can also be generated for self-driving cars. An AI model can then be used to change environments, for example by generating snow, dusk, or fog, on which the self-driving car is then trained. The question is whether it is desirable for AI models to learn from each other. After all, to what extent do you still prepare the model for actual reality?

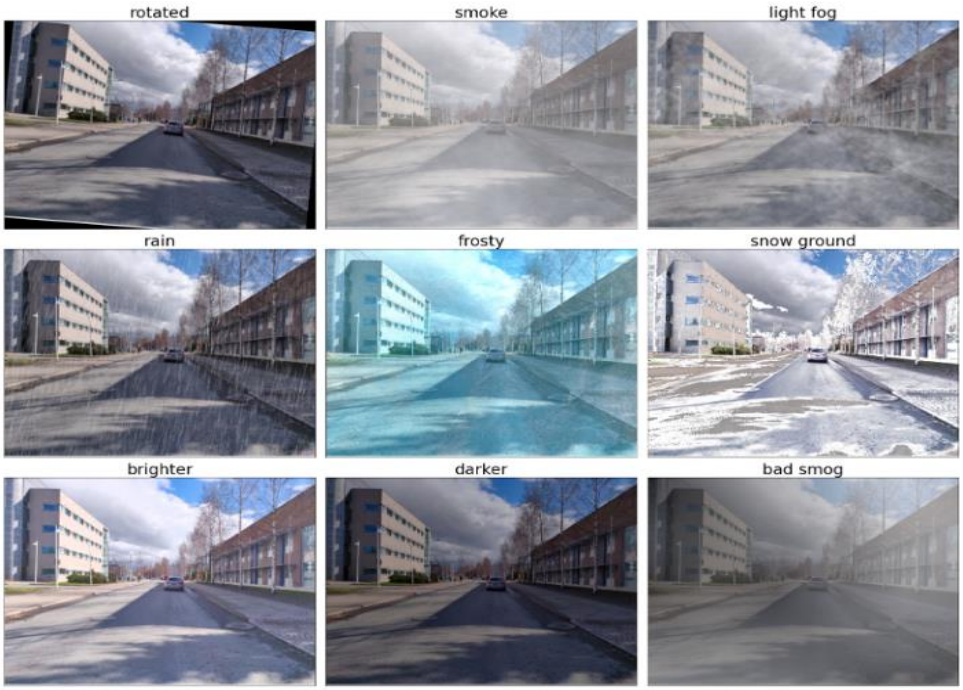


Figure 9: View of the street with varying environmental conditions (image: Teemu Kanstrén)

As long as this is merely a handy application and a check with the original or reality is possible, this application has social added value. In addition, as with speech generation, it is important that the user is informed when AI-generated images or videos are shown.

4. Different degrees of autonomy

Not all AI models operate completely independently from the moment they are put into use. This largely depends on the trust in the model, but also on the risk if errors occur. Some models will never be deployed autonomously. This chapter lists the different degrees of autonomy.

4.1 Manual input and control

In the simplest form of AI applications, someone manually offers one or more examples into a trained AI model and one or more people check the result. This is the most cautious variant, where human eyes look at the quality of the input data and the usefulness of the prediction. This caution can be due to both elements of risk:

- The probability that the data or prediction is incorrect. For example, because little experience has been gained with the prediction. Very diverse input data, complexity of the prediction, or multiple prediction goals can also be reasons for this. For example, a chatbot of an insurance company that must handle both car window damage and car theft.
- The impact it has when wrong predictions are used in practice. Think of loss of (healthy) life, high costs, reputational damage and so on.

Manual input and control is an 'expensive' variant to implement AI. The work that goes into all these checks drives up the cost of using the AI model. To be able to work more efficiently, it is customary to draw up a fixed set of checks on the basis of experience. These checks can be on the input values as well as on the predictions.

With input checks, this experience comes from experts in the field, or from experience with preprocessing data that has already been submitted to the model. The optimal end result of these input checks is that 'any' future input can be presented, with the model handling them well in all scenarios without human intervention.

Efficiency can also be achieved when manually checking predictions by the model. Think of a random check instead of a full check. Clusters can be created from related input and prediction values. In that case, the probability of incorrect predictions can be examined per cluster. Furthermore, attention can be shifted to borderline cases. For example: the model predicts with 51% certainty that the glass damage can be compensated. Or just for the extremely certain predictions, for example 99% certainty, where usually 80% is the maximum certainty of the model.

In this variant, the human still has control over the process and the AI only gives a suggestion. This can also be reversed, especially if one is still learning how the prediction works; the human gives the prediction (whether or not to accept a claim, whether or not a disease is present) and the AI checks whether this deviates from its own prediction. This is a useful application if you want to prevent humans from being guided by the AI model.

4.2 Autonomous input or processing

Autonomy of AI is a very sensitive topic in society, after all, there are many examples where an AI model was not robust enough in practice. This can be robustness for all variations of input data, or for all prediction variations and combinations it could provide. The autonomous input of data into an AI model and/or the autonomous processing of the prediction usually only takes place if:

1. There is sufficient insight into the quality of the input data and into all variations of input data that can be offered to AI now and in the near future.
2. The impact of wrong outcomes has been sufficiently assessed, so that we think we can handle all the right, the questionable and the wrong predictions that the model can give.

It is important to check at which point 'sufficient' knowledge has been gained about the impact of incorrect predictions. When recommending a product in an online store, the effort of determining the impact of all variations can be less extensive than for a surgical robot or self-driving car. Yet there are also degrees of impact with product proposals; showing an 18+ product to a 6-year-old child, or recommending medical products has a higher impact than recommending a tent when looking at products for animals. In all cases it is a risk assessment.

Autonomy requires a well-thought-out, well-balanced, and well-tested set of controls on inputs and outputs. The samples mentioned in the previous section are still possible as an ex-post quality control, but they no longer have a risk-limiting influence on the use in practice. After all, the processing has already taken place.

Safety nets are always needed in autonomous use to prevent the most important unwanted predictions. Autonomous use means that the AI model is part of a larger whole, a chain. In this smallest sized chain, the input is received from a user or other system, this input is converted so that the model can make a prediction and the prediction is converted into an action.

Testing chains is nothing new. Software development projects have given us decades of experience with this through standards such as TMAP. Still, development projects with an AI component can sometimes get bogged down, both on a technical level and on the agreements that apply within organizations about the use of data. This is especially true for projects that were started to see 'what AI can do'.

4.3 A multitude of calculations

In the previous section, it was assumed that an AI model made a prediction and thus actually made a decision. Limiting rules or controls can be used to capture illogical or unconvincing predictions.

Linking multiple AI predictions and/or programmed rules together makes things even more complex. Think of a system that answers spoken questions with spoken text. This includes several subsystems such as speech-to-text, interpretation of text, generating the answer, and text-to-speech. Or something more complex: ask Siri or Google Assistant for the nearest supermarket, which should also include geolocation and a map application.

The more complex the decision, the greater the importance of properly testing all individual components. In other words, First consider which variants are possible per component and establish that all checks work properly, before the component is activated and produces a result for the next step in the chain. After all components have been tested, it is time to test the chain as a whole. The disadvantage of a large number of components is that the chain can 'fail' in a large number of ways and the handling thereof must be done in the right way. Fortunately, the testing components and chains, the test standards mentioned in the previous section can be used.

4.4 Controlling machines with AI

When adding physical machines to a chain of algorithms, some new considerations arise. The whole of algorithms and machines is referred to as a robot. This can consist of several layers, for example the AI that makes the decisions, intermediate systems that translate the decisions into actions and

the physical hardware that performs the actual actions. When testing, each of these layers must be taken into account.

A machine can often perform many operations. Of all these actions, an AI model predicts whether this is the right one, with the highest score usually 'wins'. Here too, however, we will have to simulate different scenarios. Suppose two actions have almost equal value, which one should win? Should one of these actions even be chosen in that case? And what if the highest predicted action only scores 35% certainty?

These kinds of considerations are also reflected in AI models and in chains that do not control physical machines. Yet there is an important difference; with a piece of software there is usually a 'plan B' available; if you can't recommend a good movie, you can always recommend the most watched movie. If you do not know whether an insurance claim can be paid out, you can indicate that an employee will look at it. These kinds of secure options are often difficult with physical machines. The machines are sometimes forced to make a choice. If this choice affects one or more people, it may also have ethical implications. It is important here that the choices made when using ML are often not transparent or explainable. That's why with these kinds of ethical choices we are reluctant to let an AI algorithm decide. Ethics is therefore a very important topic within Quality and Testing of AI.

5. Ethical Guidelines and Regulations

The relationship between AI and ethics has been mentioned in this paper before. The risks of limited explainability and fear of AI partly have an ethical background. When developing and testing AI applications, the topic of ethics can provide essential new points of view.

Several organizations from both the public and private sectors have developed ethical guidelines in recent years. In this chapter, the ethical guidelines and draft regulations of the EU are used as a starting point. This chapter should be seen as a plain summary of these regulations and the underlying principles.

Since ethics is a new field for many, we start with a short section on AI and ethics. After reading this section, the subsequent sections will be easier to interpret.

5.1 AI and ethics

This section is based on the paper: 'AI no longer has a plug' by Rudy van Belkom. This paper was chosen because it discusses the relationship between AI and ethics based on a study of many sources, while still being easy to read.

5.1.1 Ethics

What 'the good thing' is and how to 'do the right thing' are questions that have occupied mankind for centuries. Ethics is a branch of philosophy that deals with this issue. Each and everyone have their own opinion when it comes to ethical dilemmas. These differences come from the fact that everyone has different values and the influence a certain context can have on those values.

Within ethics there are different movements and views that are sometimes diametrically opposed to each other. In some cases, the emphasis is on the action itself (Principle ethics) and in other cases more on the consequences that the action has (Consequential ethics). In yet another view, the emphasis is precisely on the intention of the person performing the act (Virtue ethics).

5.1.2 Ethics in relation to AI

In the past, ethics mainly concerned human actions. With the advent of AI, a new player is added to the game, namely Machine Learning technology. As technology becomes more autonomous, new ethical issues arise. Consider the issues of explainability, privacy and bias. AI technology contributes to decisions, but the decision rules of this technology are difficult to trace. This is new territory for humans and logically creates feelings of fear.

AI is already better than humans in some areas, especially when it comes to highly specialized applications. For example, algorithms are already more capable of recognizing cancers on lung x-rays than doctors. The question can be asked whether it is still justified in these situations to have certain tasks performed by people. And also whether in these situations people should be given the choice to choose an AI over humans.

5.1.3 Transparency and Fairness

In all discussions about ethics in relation to AI, two points keep coming up: transparency and fairness.

- As AI can make increasingly autonomous decisions, it is important that it is transparent how the decisions of such systems are made.

- When it comes to fairness, it is basically about the equality of people. People should be treated equally and given equal opportunities. This does not mean that there cannot or should not be differences between people. But when people are treated differently, there must be a clearly identifiable reason to justify the difference.

AI applications are used in various domains, such as business, government, security and the medical world. This makes it difficult to draw up clear and overarching ethical frameworks. What is 'good' or 'correct' always depends on the specific context. In addition, the differences in ethical movements and views do not make things any easier. It remains a democratic issue to reach consensus about what is the best decision for society. The discussion to arrive at this consensus is often based on practical examples, such as the example below from Uber.

> *Example of transparency in the algorithms*
>
> Several drivers accused Uber of using algorithms to determine which drivers should be fired. The drivers were said to have been fired after Uber's algorithms found that they were guilty of 'fraudulent activities', while they themselves denied these activities.
>
> Automated dismissal of people is not allowed, the drivers said. According to the British and European privacy law, a person must make such a drastic decision and it should not depend on the outcome of an algorithm. Uber denied that automated decisions were made about which drivers should leave, but according to the drivers there was no "meaningful human intervention". It is required that a human being reaches a decision and must do more than follow the advice of an algorithm.
>
> In February 2021, Uber was ordered in absentia by the court of Amsterdam to allow a driver onto the platform again. Because it concerned a judgment in absentia - which means that Uber was not present at the hearing - the judge did not investigate the substance of the case.

In another case, drivers wanted to know how Uber's algorithms judge them and what effect that had on how rides are allocated between drivers. Based on European privacy legislation, they have a right to receive this type of information. The GDPR privacy law allows residents of EU countries to ask what companies and authorities know about them, but also what happens with that data. For example, they must be able to see how their data is treated and whether decisions are taken automatically based on that data. This example shows that transparent and explainable use of AI is an important precondition for proper discussions about ethics.

5.2 EU Ethical Guidelines

Within Europe, the High-Level Expert Group on Artificial Intelligence is concerned with ethics. Their document “Ethics Guidelines for Trustworthy AI” states the following:

> “AI systems should focus on people and strive to use these systems in the service of humanity and the common good, with the aim of improving human well-being and freedom.”

This shows that the EU recognizes the possibilities of AI systems and wants to stimulate them. The aim is to optimize the benefits of using these systems while minimizing the risks associated with their use. Trust and reliability are preconditions for achieving this goal and the expert group has put together a framework that must be met throughout the entire life cycle of the system. This framework consists of the following three components:

5.2.1 Robust AI

Robustness is a well-known quality characteristic that is also included in the ISO 25010 standard. In the context of AI, this characteristic is given a broader interpretation. It is argued that an AI system must be robust, not only from a technical point of view, but also from a social point of view. This is described in this way because AI systems can unwittingly cause damage, even when the intentions are good. For this reason, precautions should be taken to avoid unintended negative consequences.

5.2.2 Lawful AI

The AI system must be legal, by ensuring that all applicable laws and regulations are met. The legislation contains both positive and negative obligations. This means that it must be interpreted not only in relation to what is not allowed, but also in relation to what must be done. It is based on a number of fundamental rights, such as:

- Respect for human dignity
- Freedom of the individual
- Respect for democracy, justice, and the rule of law
- Equality, non-discrimination, and solidarity

This component will be supported by additional legislation in the future.

5.2.3 Ethical AI

The AI system must be ethical, by ensuring that ethical principles and values are adhered to. These ethical principles stem from the above-mentioned fundamental rights and are named as follows:

The principle of respect for human autonomy

People who work with AI systems must be able to maintain their full and effective self-determination. AI systems should not unreasonably subjugate, coerce, or mislead people.

The principle of damage prevention

AI systems should not cause or increase damage or otherwise have negative consequences for humans. They must be technically robust, and care must be taken to ensure that they do not allow for malicious use.

The principle of justice

Individuals and groups must be free from bias, discrimination, and stigma. In addition, the use of AI systems must never lead to the users being misled or limited in their freedom of choice. Also under this dimension is the possibility to challenge decisions made by AI systems.

The principle of accountability

Accountability is critical to creating and maintaining user trust in AI systems. This means that processes must be transparent, the purpose of the system must be communicated, and decisions must, as far as possible, be explainable to those who are directly or indirectly affected by them.

5.3 The EU's draft regulation on AI

Many AI systems have a limited to negligible risk and can be used to solve many societal challenges. However, certain AI systems create risks that need to be addressed to avoid unwanted outcomes. For

example, the opacity of algorithms can lead to uncertainty and can hinder the effective enforcement of existing legislation on security and fundamental rights.

For companies, this can lead to legal uncertainty and delay the convergence of AI technologies due to the lack of trust. In April 2021, the European Commission presented proposals for legal regulations to combat the misuse of AI. This is a follow-up to the Ethical Guidelines of the EU. This draft regulation also comes with a comprehensive assessment to identify the risks and identify the requirements.

The Commission proposes a risk-based approach with four levels of risk, namely:

5.3.1 Unacceptable risk:

This includes all AI systems that violate EU values because they violate fundamental rights. Systems that are considered a clear threat to the security, livelihoods and rights of people will be banned.

5.3.2 High-risk:

AI systems classified as high-risk include AI technology used in:

- Critical infrastructures (e.g., transport) that can endanger the life and health of citizens
- Educational or vocational training, which can determine access to education and the professional course of one's life (e.g., scoring exams)
- Safety components of products (e.g., AI application in surgery supported by robots)
- Work, personnel management, and access to self-employment (e.g., CV sorting software for recruitment procedures)
- Essential private and public services (e.g., a credit score that denies citizens the opportunity to get a loan)
- Law enforcement that may infringe on people's fundamental rights (e.g., evaluation of the reliability of evidence)
- Migration, asylum, and border control management (e.g., verification of authenticity of travel documents)
- Administration of justice and democratic processes (e.g., application of the law to a concrete set of facts)

Binding requirements are proposed for the AI systems that fall into this category before they can be placed on the market. These requirements relate to the quality of:

- datasets used
- documentation and data
- registration transparency and provision of information to users
- human supervision
- robustness
- accuracy
- cybersecurity

5.3.3 Limited risk:

The AI systems that fall into this category have an obligation to transparency. When using AI systems such as chatbots, users must be aware that they are interacting with a machine.

5.3.4 Minimal risk:

AI systems that fall into this category can be developed and used in accordance with existing legislation without additional legal obligations.

The vast majority of AI systems currently in use in the EU, such as AI-based video games or spam filters, belong to the latter category with minimal risk. Providers of those systems can choose to apply the requirements for trusted AI and adhere to non-binding codes of conduct. The committee will encourage industry associations and other representative organizations to establish voluntary codes of conduct.

5.4 The Creation of Trustworthy AI

A checklist and an online assessment tool are available for checking reliable AI systems. A reference to these resources is included in Appendix A. These can help identify potential risks posed by AI systems and determine whether, and what kind of measures should be taken to mitigate those risks. A number of methods, including testing and validation, are also described.

The European Commission's expert group argues that testing and validating a system should be done as early as possible to ensure that the system behaves as intended throughout its entire life cycle, especially after installation. Due to the nature of AI systems, traditional testing is not enough. Therefore, to verify and validate the data processing, the underlying model must be carefully monitored during both training and installation for stability, robustness, and operation within clear and predictable limits. Care must be taken to ensure that the outcome of the planning process is consistent with the inputs and that decisions are made in such a way that the underlying process can be validated.

It must include all components of an AI system, including data, pre-trained models, environments, and the behavior of the system as a whole. The system must be designed and implemented by as diverse a group of people as possible, so that sufficient attention is paid to the various possible angles and backgrounds during training.

Multiple metrics need to be developed for the categories being tested from different points of view. This concerns, for example, the use of ethical hackers. This testing from other angles is known as adversarial testing. Finally, it must be ensured that the results or actions are consistent with the results of the previous processes, by comparing them with the previously established policy, so that this is not violated.

The above description of the method testing, and validation is taken from the EU guidelines. This makes it clear that the EU also sees the importance of thorough testing, considering the ethical aspects and from different backgrounds and angles. The guidelines provide a good platform for quality and testing of AI.

6. Quality attributes

Quality attributes can help users and stakeholders to determine which attributes of quality are important for software systems. This also goes for AI systems.

The ISO 25010 is a well-known standard for quality attributes. The attributes of this standard can also be used for AI systems, but do not yet cover all aspects of AI. Various parties have thought about, or are still thinking about, adding (sub) attributes to get a standard that also covers AI systems.

6.1 The current ISO 25010 standard

Not only functionality is important for the correct operation of software. Other aspects such as performance, usability, stability, and maintainability are also important.

The ISO 25010 standard brought together a number of these aspects, the so-called quality attributes, in a framework.

This standard recognizes the following attributes regarding product quality:

Attribute	Description
Functionality	The extent to which stated and assumed needs are met.
Performance	The performance relative to the number of resources used under the stated conditions.
Compatibility	The extent to which a system can exchange information or perform in other environments.
Usability	The extent to which a system can be used to achieve the specified goals.
Reliability	The extent to which a system performs functions over a long period of time and in different situations.
Security	The degree to which a product or system protects information and data.
Maintainability	The extent to which a system can be changed.
Portability	The extent to which a system can be ported to another environment.

Table 2: The ISO 25010 standard

Each attribute also contains several sub-attributes.

This paper will not go into further detail on these aspects, there are plenty of other sources for this. However, it can be safely stated that these quality attributes, both for traditional software systems and for AI systems, can be applied to provide insight into the quality of the system. Yet it is not easy

for ML systems to give substance to a number of sub-attributes. Below are some considerations regarding these challenges.

Functionality		Comment
Completeness	The extent to which the set of functionalities supports all specified tasks and goals for users.	Completeness will never be 100%.
Correctness	The extent to which a software product or computer system provides the correct results with the required accuracy.	What accuracy is acceptable?
Maintainability		
Modularity	The extent to which a system or computer program is built up in separate components so that changes to one component have minimal impact on other components.	In the event of regression, the impact will usually be a 100% retrain.
Reusability	The extent to which an existing part can be used in more than one system or when building a new part.	This is only possible if the exact same data structure and ratio is used.
Analyzability	The extent to which it is possible to effectively and efficiently assess the impact of a planned change of one or more parts on a product or system, to determine deviations and/or fault causes of a product or to identify those that need to be changed.	This is not easy, especially with the ML variant Deep Learning.
Modifiability	The extent to which a product or system can be changed effectively and efficiently without resulting in errors or reduction in quality.	See again the point of regression.
Testability	The extent to which test criteria can be effectively and efficiently established for a system, product or component and to which tests can be performed to determine whether those criteria have been met.	See the risks mentioned in Chapters 3 and 4 and the introduction to Chapter 7.

Table 3: Challenges for ISO 25010 attributes in relation to AI attributes

For AI systems, and especially ML systems, some of the current quality attributes are not easy to define or verify. This makes testing an ML system different from a traditionally programmed software product. In addition, it appears that some of the attributes do not cover all characteristics of AI systems.

6.2 Additional quality attributes from 3 different sources

In the following paragraphs we provide an overview of new, additional quality attributes. These can be used to describe and assess the quality of an AI system. This overview has been compiled from several sources, whereby there is also a certain overlap with each other. These resources also contain ethics related attributes.

6.2.1 Source 1: Testing in the digital age

The book “Testing in the digital age - AI makes the difference” describes an extension of the quality attributes of the ISO 25010 standard. In this book three new main groups are added.

Attribute - sub-attribute	Description
Intelligent behavior	Intelligent behavior is the ability to understand. It is in fact a combination of reasoning, memory, imagination and judgment. All of these faculties are dependent on the others.
● Ability to learn	The ability to learn is the ability to understand and benefit from experience.
● Improvisation	Improvisation is the power of the intelligent system to make the right decisions in new situations.
● Transparency of choices	Understanding how the system comes to a decision.
● Collaboration	The extent to which the system responds to changes in human behavior.
● Natural interaction	This type of interaction is important in verbal and non-verbal communication. Especially with social robots, it is important that the way people interact with a robot is natural, the way they interact with humans. But also with a search engine, for example: ‘Should I bring an umbrella tomorrow?’
Morality	See chapter 5.
● Ethics	See chapter 5.
● Privacy	See chapter 5.
● Human friendliness	This has to do with safety and security.

Personality	A personality is the combination of characteristics or qualities that make up the distinctive character of an individual.
● Mood	A temporary state of mind or feeling.
● Empathy	The ability to understand and share another person's feelings.
● Humor	The quality of being funny or comical.
● Charisma	The compelling attractiveness or charm that can inspire others to devotion.

Table 4: Additional quality attributes described in the book Testing in the Digital Age

6.2.2 Source 2: ISO/CEN 5059 / ISO/IEC WO 5059

This new ISO standard is still being developed. The following attributes have been mentioned in several webinars:

Attribute – sub-attribute	Description
Ability to learn	The ability of the system to learn from the use of the system itself, or the data and events to which the system is exposed.
Ability to generalize	The ability of the system to be applied successfully to different and never-before-seen scenarios to the system.
Trustworthiness	The extent to which the system can be trusted by stakeholders.
Robustness	The extent to which the system or an application is sensitive to external interference.
Controllability	The extent to which an external agent can intervene in the functioning of the AI system.
Explainability	The extent to which important factors influencing the AI system (e.g., algorithms, model) can be expressed in a way that people can understand.

Table 5: Concept of additional quality ISO/CEN 5059 / ISO/IEC WO 5059

There are also various ethical attributes:

● Explicability and accountability ● Respect for democracy, justice, and the rule of law ● Responsibility ● Privacy	● Fairness and non-discrimination ● Transparency ● Reinforcement of existing bias ● Consistency ● Free from bias

Table 6: Concept of additional ethical quality ISO/CEN 5059 / ISO/IEC WO 5059

As indicated earlier, many of these attributes are also mentioned in some form in the previous source or in the ethics chapter. For this reason, we do not elaborate on these ethical attributes here. Taken together, this is a confirmation that these characteristics are important.

6.2.3 Source 3: DIN SPEC 92001-1 AI, Life Cycle Processes and Quality Requirements

The German standards institute DIN has developed a specification according to the PAS procedure (Publicly Available Specification) and is freely downloadable - see Appendix A for the reference. A DIN SPEC can be used as the basis for a future standard. This paper proposes an approach to analyze AI-related software quality aspects.

Ensuring high quality of certain AI modules is a difficult task. Especially in ML, because of the unpredictable response to unforeseen input and in DL because of the lack of transparency. Addressing these challenges in a structured manner is a foundation for the successful development and integration of robust, secure, and reliable AI modules.

To make this possible, this document describes a model with the following quality characteristics:

Quality Attribute	Description
Functionality & Performance	These characteristics indicate the extent to which an AI module can fulfill its intended task under certain conditions.
Robustness	Robustness indicates that an AI module can deal with erroneous, polluted, unknown and hostile input data. Due to the complexity of the AI module's environment, robustness is an important AI quality attribute.
Comprehensibility	Machine Learning models are mostly not transparent, making mapping from input to output largely incomprehensible to stakeholders. This means that the AI component must be transparent and interpretable. Legislation is a possible external constraint.

Table 7: Quality attributes DIN SPEC 92001-1

Note that these characteristics are not completely independent from each other, but this classification facilitates a structured enumeration of specific quality requirements.

6.3 In conclusion

The most important concepts from the three sources summarized: an addition to the current quality attributes is desirable. Several initiatives have been launched to search for these quality attributes, from different parties and with different perspectives. When the initiatives are compared side by side, there are differences, but the similarities are striking.

In general, attention needs to be paid to autonomy and additional expectations regarding consistency, robustness, and self-reliance. Attention should also be paid to the intelligence and humanity of non-human systems; considering learning and the ability to generalize, transparency and trust, and social interaction.

It seems obvious that there will be consensus about the formation of new or adapted quality attributes. This will create a good and widely supported set of quality attributes to specify and test the quality of AI solutions.

7. Testing of AI

The previous chapters have shown various perspectives on the application of AI and the associated risks. After reading those chapters it will be clear that the risks and focus area to a large extent, are different than with traditionally programmed solutions. In this chapter, all possible risks and focus areas are translated to test activities.

The test activities are deduced from the risks as described in Chapter 2. An explicit link between risk and test activity is made in Appendix D.

7.1 Static Testing

When an ML project is not successful, it often turns out that several quality problems could have been anticipated prior to the development of the model. This can be prevented by static testing. Consider checklists, reviews, and assessments. These kinds of tests can not only be performed at the start of a project, but also during the project when a certain milestone has been reached.

7.1.1 Checklists

Checklists which are common in the market, or that have been made for the project in consultation with the stakeholders, can be used. Points included in such a checklist could be:

- Is the goal clear?
- Is the timeline clear?
- Can expectations of the stakeholders be reached?
- Is the data available?
- Is there sufficient data and is its ratio in balance?
- Is the solution future-proof?
- Has there been a similar project or academic research?
- Are ethical issues anticipated?

For each implementation, the way to measure success has to be defined as well as minimum viable AI solution. The latter is certainly important, because it is not possible to work with an explicitly achieved end result. This makes it hard to have detailed test cases, but checklists and test guidelines can provide guidance and a degree of consistency.

7.1.2 Reviews

Various forms of reviews are possible, from formal inspections to peer reviews. Reviews provide insight and valuable feedback regarding a certain aspect. Reviews can take place prior to, during and after the development process. Some areas that can be considered for a review at the start of the project are research into the available data, research into the proposed ML algorithm, a review of the goal and the means to achieve this goal, including ethical considerations and how the success of the model is measured. For example, during or after the construction of the model, the project team can consider a technical review of the code or parameter setting, a process review with lessons learned including recommendations for improvements.

The information obtained from these static tests is very suitable for creating test cases and test scenarios.

7.2 Testing the data

In the preceding chapters the importance of good data has already been emphasized. This leads to the conclusion that testing this data is an important step in the development process. ML learns based on the available data. If this data is not complete or incorrect, the delivered model will not perform properly. For this reason, the dependency on data is one of the most important risks mentioned in Chapter 2. In the explanation of this risk, a number of points are mentioned which influence the quality of a model and can serve as a foundation for testing. With data, a distinction should be made between the data that is used as input value, for example the surface area of a house, or the number of rooms, and what this collection of data ultimately represents, for example the selling price of a house. As indicated in the example about recognizing a cat this distinction is often subjective.

Obtaining data and making it suitable for further processing, the so-called preprocessing of the data, is a specialism called Data Engineering. This includes supplementing missing data or transforming data to a different format. This activity can be performed manually or automatically. To monitor the quality of this process, it is important to describe the steps of how the data is obtained, and the preprocessing steps through which this process can be tested.

When obtaining the data, it must be tested to what extent this data corresponds to reality. In this way it can be tested whether the male/female ratio is correct, and whether the age distribution corresponds to the target group.

If the data comes from multiple sources, it can also affect how the model works. Questions asked on the phone are often different from those on a company's website. In addition, it is plausible that other types of people seek contact via social media than contact via telephone. For this reason, data cannot be merged without checking.

Data must also be tested when using the model in production. In part, this is an input validation that needs to be tested, such as:

- Is the format correct?
- Does the data fall within the ranges used for learning?

It is important to compare the pattern of input data to the pattern of data with which the model has been trained. If there has been a shift in this pattern, this may be an indication that the model needs to be optimized. Suppose when making a house price model that 80% of the houses have 3 or 4 rooms, over time this share falls to 50%. This can affect the model. For this reason, it must be examined to what extent the model still corresponds to reality. Which emphasizes the importance of proper recording and monitoring for the distribution of data.

7.3 Testing the model

Creating a model requires a limited amount of code or can sometimes be done with only a parameter configuration (so-called Auto-ML). Despite this limited arrangement, a relatively limited adjustment of a parameter can have a major impact on the result. Testing this configuration of the model is project specific, but issues that need attention are:

- Was the correct algorithm used?
- Are the parameters of this algorithm entered correctly?
- Is the input and output format correct?
- How long does it take to train the model?

In practice, when one speaks of testing a model, it does not mean the above configuration test, it is about evaluating the model. In other words, how good are the model's predictions?

This is a standard step in the development process and an iterative process, the outcome of this evaluation is used to tune the parameters to get a better result. This continues until there is no more improvement. The performance of a model is measured by various indicators, such as accuracy. For more information, see Appendix C.

Testing boils down to comparing the predicted values with the actual values. In principle, the data used for testing is another set of data than the dataset used for training the model.

It is common to divide the available data, one part to train the model and one part to test the model. Because this test data has not been used to train the model and has not been seen by the model before, this data is similar to data in the production environment and is therefore suitable for assessing the model. This also indicates that the distribution must be done carefully, because both datasets must have the same characteristics. For example, the same mean, distribution and number of groups. This must be analyzed prior to the start of testing a model, otherwise no judgment can be given about the functionality of the model.

Besides being able to test a model on the result obtained, i.e., how good is the obtained prediction, you can also test a model by investigating how the model arrived at this prediction. Explaining a model is known as Explainable AI. A comment should be made immediately; ML models based on DL techniques are difficult to explain in practice. Instead of actual explaining, 'interpreting the behavior' is a better description. Interpretation is the extent to which a person can understand the reason for a decision. For example, it is possible to investigate which input variables contribute the most to the final result, or to examine how an individual result has been established.

With Image Recognition it is possible to check which pixels contribute the most to the prediction result. This can be made visible with a so-called heatmap; the more a pixel contributes to the result, the warmer, the more red this pixel is projected on the original photo.

See the images below as an example:

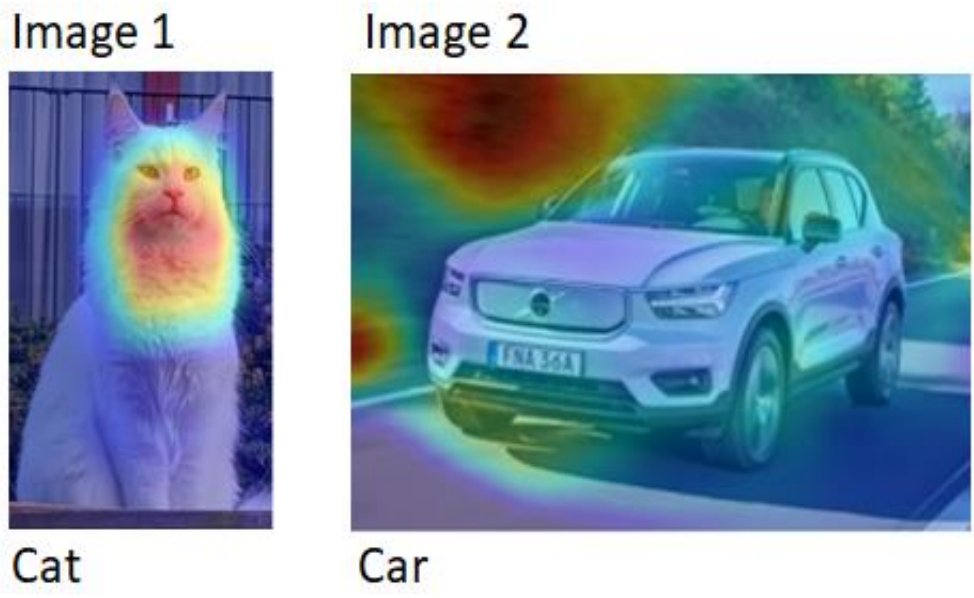


Image 10 Heatmap of Cat and Car

In image 1, the cat, most of the red pixels on the heatmap are in the face of the cat. This indeed is a typical characteristic of a cat and gives a good indication that the decision to classify this image as a cat was made on the right grounds.

In image 2, the car, most of the red area on the heatmap is off the car. This means that the most important area on this image used to classify the representation as a car is not the car. This immediately indicates that the prediction that this image represents a car cannot be worth much.

7.4 Testing the functionality of the model

When testing an ML model, it is not expected that each individual test data record leads to a perfect prediction. It is about the overall performance of the model. It is quite possible for a test to pass with an accuracy of 80%, a single wrongly predicted result is not a bug. The question is not whether a model is correct, but how well the model solves a problem.

Determining the quality of a model requires more than its general accuracy. To give an opinion on the obtained functionality of the model, it can be good to also examine the various sub-areas, such as boundary cases and target groups. A number of test techniques are described below:

7.4.1 A/B testing

In an A/B test, different model versions are compared. A selected experiment group (the B group) will receive the modified version and the control group (the A group) will receive the unmodified version. The differences in results or behavior are analyzed. This test passes the test oracle problem. This test is not about how good a model is, but whether a model is 'statistically significantly' better than another version of the model. That is why this test is widely used when testing ML systems.

The A/B test can be done in the development stage as well as in the production stage. Sometimes it is difficult to determine to what extent a version is better. Think of adding new functionality when there is only a limited amount of data to test this new functionality, or the expectation that the behavior of the user will change.

This test technique has several variants. For example, by having the A and B groups use the same data, the differences between the model versions can be analyzed. Another variant is to divide the test data in an equal way and use it to test the model, a so-called A/A test. Since it is the same model version, you would not expect any differences.

These types of tests give an assessment of the performance of the model as a whole. Comparing the performance between multiple models often does not provide direct insight into the model behavior itself. An example can be that a model can be better on average, but it can then happen that a certain group (class) from the data or user population is favored or disadvantaged in the updated version. To assess this, other test methods are needed: methods that test the behavior of the model and deliver an understandable and predictable result.

7.4.2 Equivalence Partitioning

The tests described above mainly focus on optimizing the overall quality. This may be too general. Models that achieve high overall performance can produce unacceptable failures on critical parts of the data, such as the detection of vulnerable cyclists during an autonomous car journey.

Equivalence Partitioning divides the data into partitions (classes) in such a way that all members of a given partition are expected to be processed in the same way and produce a similar result. So, it's about consistent behavior; in similar cases, similar results are expected. Since ML systems process large amounts of data, there are tools to support this. For example, pairwise testing could be applied

here. A combination of variables (or partitions) is tested simultaneously, so that reasonable coverage is still possible with a limited subset of combinations. Another variant is to ask domain experts about the most relevant partitions. With a technique such as the Data Combination Test, a number of input fields can be combined in order to determine the correct coverage.

Using these tests, it is possible to make a more substantive comparison between model versions. A new version of a model is not expected to differ in the result of a partition, unless this was a conscious choice when creating this new version. Therefore, good version management is essential for both the test data and the results per test.

7.4.3 Boundary Value Analysis

This is a test that can be performed after the Equivalence Partitioning test. The minimum and maximum values (or first and last values of a data set) of a partition are its boundary values. The behavior around the boundaries of the partitions is more likely to be different than the behavior within the partitions. Using Machine Learning, this needs to be interpreted a little more broadly. Consider, for example, an application for image recognition. During the daylight hours and during the night there are expected to be few differences in recognizing an object, but in the boundary between these two partitions, dusk and dawn with varying luminosity and changing colors, there are many possible light conditions. The behavior of the model must be tested for this.

The Corner Test can be used as a variant of this test. With Corner Testing, the extreme values of all input variables (features) are taken. A distinction can also be made between the minimum and maximum value of this variable in the (test) dataset, or the possible minimum or maximum value that can serve as input to the model.

7.4.4 Metamorphic Testing

This is a test method to reduce the test-oracle problem. The idea is simple: even if we don't know what the correct output (the result) should be from a single input, we can still know the relationships between the output of input, especially if the inputs themselves are related. We can check the model for these relationships: the so-called metamorphic relationship. If the test shows that the relation is not kept, then this is certainly a signal to investigate further.

The example from Chapter 1, predicting the value of a house, can serve as an illustration. It is impossible to determine the 'correct' value of a house. However, it can be assumed that the house price increases in value as soon as a house has more surface area. In this example, a metamorphic relationship can be made between the input values and output values, namely if the surface area increases, the house price will rise.

This test level can be used in combination with the partitions as described in the Equivalence Partitioning test method. We do not expect a significant change in the output from the input values within a partition. This is also known as an invariance test. These tests allow us to describe a series of changes that we should be able to apply to the input without affecting the output of the model. We can use these perturbations to produce pairs of input samples (original and perturbed) and to check for consistency in the model's predictions. The desired result must remain the same within the bandwidth.

This technique can also be used in the opposite direction, it is then called a Directional Expectation Test. By means of change on the input, we expect a predictable effect on the output of the model. It should result in a deliberate difference outside the bandwidth.

7.4.5 User Story Testing – Use Case Testing

The description of the desired functionality, desired behavior and requirements can be specified in the form of User Stories and/or Use Cases. When using this technique, the preconditions, conditions, and acceptance criteria are specified. The compilation of these tests generally takes place in dialogue with the stakeholders and/or domain experts. This test method is widely used in traditional software development but can also be used well when testing ML applications.

7.4.6 Expert Panel Testing

By using the knowledge and skills of domain experts and other experts, the result of an ML application can be compared with the results expected by these experts. Input test data sets are compiled based on domain analysis. Both the ML application and the experts determine the likely results based on these test data sets. This allows the quality of the ML application to be assessed.

When using experts, it is important to take the following points into account:

- Human experts vary in competence, so the experts involved must be representative in number, area of expertise and level of knowledge.
- Experts may not agree with each other even when given the same information. In many cases, the correctness of the test is perceived differently by different individual users. For example, what may seem like a complaint to some may come across as a neutral statement to others.
- Human experts may be biased for or against automation, see the risk General fear of AI.

Despite these points, these types of tests can quickly provide insight into the quality of a model and contribute to its acceptance in production.

7.4.7 Experience-based Testing

In experience-based testing techniques, the test cases are derived from the skills and intuition of testers and the experience with similar applications. These techniques can be helpful in identifying tests that are not easy to describe. This situation will certainly apply to a number of ML applications.

Experience-based tests are:

Exploratory Testing

In Exploratory Testing, the tests are not described in advance, but the evaluation takes place during the execution of the test. An exploratory test looks at a certain theme, which is described in a test charter. This ensures that you document what you have tested, which in turn helps with trust and acceptance. Test charters can focus on, for example, capturing abuse of the model, finding underrepresented examples and recognizing bias. This can be done both for the analysis of the data and for the development of a model.

When developing ML models, it is difficult to estimate in advance what the achievable specifications are and the requirements are often only briefly specified. Therefore, this test approach is useful and usable. Exploratory testing is often used in addition to other more formal testing techniques, or when there is significant time pressure on the testing process.

Error Guessing

Based on domain knowledge and knowledge of the process, the data set is enhanced with situations for which problems can be expected. These can be rare combinations across extremes. But also:

- How the ML model has worked in the past

- Flaws, weaknesses, inherent in the chosen model
- Mistakes that data engineers can make
- Errors that have occurred in other applications

7.4.8 Testing using Personas

Personas are fictional characters representing stakeholder or user groups. This method starts with identifying and compiling the user profiles. In the profiles, matters that may be important for the (AI) application under test are specified, such as education level, reason for using the application and the like. The advantage of using personas is that an attempt is made to put yourself in someone else's shoes and view the (AI) application from these eyes.

It is a test method that is suitable for detecting specific prejudices. As indicated in the chapter on ethics, a model should not be prejudiced. Unfortunately, this phenomenon is very likely to occur. In some systems, for example, a dual nationality is registered, and different nationalities might be processed in different ways. Since a model is trained on data from the past, the behavior of the model may also have been influenced by this. Creating a persona with dual nationality will make sure that this concern will be covered.

7.5 Testing for Drift

Generally, the creation of the first ML model is not that complicated compared to the first version of programmed software. However, maintaining a model requires more attention, due to the need to respond to changes in the market and availability of data. The results of a model will change over time. It is possible that the pattern of input data has changed, but it is also possible that the result of the model is valued in a different way. The result is that the ML model becomes less effective. This phenomenon is called drift and in some cases can occur within a few days. Causes are, for example, changes in user preferences, marketing campaigns, seasonal influence or adjustments at a competitor. In practice, this means that there is almost never a definitive version of an ML model.

For these reasons, a model in production should always be monitored to detect these types of situations. The model will have to be retrained to be able to function again in this changed world. Since the process of detecting and the action of adapting the model is crucial for the effectiveness, this process also needs to be tested. This is a guarantee for the consistent quality of the model.

7.6 Regression testing

As with traditional software, an AI model will be updated regularly. This can have several reasons:

1. The model has to be retrained because the original goals are no longer being achieved or because reality has changed and the model does not match any more. We explained this concept in the previous section as a concept of drift.
2. New or modified goals are being pursued with the model, while some original goals are still applicable. Think of recognizing new questions by a chatbot or a new product category in product proposals.

Therefore, it is important that the requirements of the model are clear and that the model is monitored to check if it complies to the requirements.

An update of the model requires some caution. In traditional software development, the risk of regression errors is usually manageable because only a limited amount of code is modified with a new release. This is different with Machine Learning, as the model is retrained it may behave entirely

differently. So even with a limited change, it is possible that the model changes on crucial points. It may also be the case that the overall performance of the model is improved, but an unwanted negative effect in a data group (such as boundary values, data partitions and persona's profiles) still occurs. This fact justifies the importance of periodically performing good regression tests in Machine Learning.

With each update, it will have to be tested to what extent there has been a shift in the result. This can be done by means of the discussed A/B tests. It is important to keep the original test dataset including the associated results in order to recognize the growth and change of the model. In addition, a test dataset with current data is important to test the current performance. Maintaining old and new test cases and test results for all model versions is needed to assess the quality of the current and previous models.

7.7 In conclusion

We believe that the described test methods and quality characteristics provide insight into the quality of a model. Of course, such an overview is never complete and one test will be more suitable than the other per situation and per type of model. In addition, it is quite possible to combine several test methods in a test, as testers have been doing for many years. It will also have to be examined per test which risks, quality characteristics and ethical aspects need to be covered. The test methods we described here show the specifics when testing an ML model. Integration testing, security testing and user acceptance testing have not been mentioned here because these tests do not differ fundamentally for an ML model.

It's much easier for people to trust technologies over which they can exercise complete, constant control. This is one of the biggest challenges in AI and ML. By testing for consistent and predictable behavior and the use of documented versions and test results to explain unexpected behavior when a model has been retrained, it will increase the level of confidence in the operation of ML systems.

8. Testing AI in practice

The role of testers and quality assurance specialists in AI processes is relatively new. If you are amongst these professionals, this offers many opportunities to shape this role yourself and to contribute to the successful implementation of an AI application. In this chapter we take a closer look at projects, roles, and skills.

8.1 The course of an AI project

Every AI initiative (or specifically: ML initiative) will want to use data to achieve a certain goal. Sometimes this will arise from the curiosity or need of one of the employees. If the data is available, an experiment can be started. A first concept model can be quickly set up and trained.

Another starting point is a business case, often to speed up a process within the organization or to better serve the customer. These kinds of initiatives are more likely to be set up as a full-fledged project, but here too it usually starts with an idea from an employee. The starting point where the board or management indicates that 'we should do something with AI' considerably reduces the chance of a successful implementation.

An AI project is a constant exploration. Team members learn which data is available, what quality it has and to what extent it helps to achieve a certain goal. In the meantime, various models will be trained. Each iteration provides new insights and new challenges to solve.

The following activities will play a role in an iteration, although the focus in the first iterations will mainly be on the first activities and in the last iterations it will be on the last activities.

- Idea elaboration. This is a phase in which the balance between ambition and feasibility is examined. It is also a good time to ask critical questions, for example whether ML is indeed the best choice for the problem, whether the right knowledge and resources are available, whether the data is available and usable, what general requirements there are around topics such as ethics, explainability, privacy, and so on.
- Data collection and processing. This is retrieving internal or external data sources, assessing consistency, completeness, and usability. Selecting, editing and supplementing also takes place here. Of course this should be done according to clear, logical and recorded choices.
- Modeling, training and validation. Here the model variant is chosen, possibly on the basis of tests with several variants. The models are configured, data is fed into the model, and model scores are produced. These scores are assessed based on a separate dataset.
- Testing. This is the substantive, manual assessment of the model. This involves looking at individual examples, such as limit values and outliers. The aim is to establish that the scores look logical by looking at them critically, preferably with the help of domain experts. In addition, the technical tests will also be carried out here to see the model at work in a possible chain.
- Accepting. This is the acceptance test, in which all stakeholders must be convinced that the AI model is useful and pleasant to work with. Stakeholders are representatives of customers or other end users, or their own employees who will be involved, but also a representative of the client who will want to validate the business case. In addition, the lawyers and the security department of the organization will want to give their opinion.

- Implementing and using. Here the organization is prepared to use the model in practice. For example, customer service may have questions that need to be answered. The scores of the model must also be monitored in the production environment and it must be possible to implement a new version of the model and any additional control measure as a safety net when this proves necessary.

8.2 Roles in an AI project

A commonly used division of roles in current software development is the following.

- A designer or architect who thinks about what needs to be developed.
- A programmer who works out the plan in detail and then writes it out in code.
- A tester who, based on risks, checks whether there are errors or ambiguities.
- An acceptor, who checks whether the whole is workable and adds value.
- A project leader or scrum master, who ensures progress and collaboration.

Working according to the iterative agile or devops method fits very well with the way AI models are developed and managed. AI development is pre-eminently a joint exploration process to extract predictive value from data, in which everyone contributes ideas and asks questions from their own specialism. The development of an AI application is also an iterative process. In addition, an AI model will always have to be closely monitored after it has been put into use and it will have to be retrained on a regular basis.

We see new roles emerging in relation to agile and devops:

- An ML data scientist, who ensures that data is collected and made usable.
- An ML engineer, who builds and tests a model based on usable data.
- An AI Ethics officer, similar to the Data Privacy Officers after the GDPR responsible for achieving and monitoring compliance.

Another difference with agile is that the acceptor is involved much earlier in the project. He or she is actively involved in making the data useful, based on knowledge from practice. So there is already an active collaboration between the data specialist and the acceptor. Then, as soon as a first version of the model has been trained, the acceptor will start interpreting the emerging trends and the most striking outcomes. There will be joint follow up to create the next iteration of the model.

Structured collaboration in an AI project is important. The ML engineer provides factual data and the acceptor interprets the results. However, there are many more activities that contribute to improving quality. This is where testers or QA specialists can play a role, so that sufficient certainty about coverage and unmitigated risks is ultimately obtained. Participating in an AI project offers testers opportunities to show their added value and to develop further in a new field.

8.3 Knowledge and skills

Skills needed in an AI project include traditional skills of testing associated with quality assurance, such as finding errors and exceptions, making connections and trying out unexpected combinations, and estimating risks. The most suitable profile of the tester or QA specialist will be different per project. Sometimes more coordinating skills will be required, sometimes the emphasis is more on technology.

When testing in AI projects, there may be a need for new or more in-depth knowledge and skills such as:

- Extensive proficiency in using black box testing techniques, especially in AI development where deep learning is used and it is impossible to assess the internal workings of the model.
- Knowledge of ethical considerations that must be made, based on guidelines from the government or from the own organization. It is also important to be able to draw attention to this at the right times and in the right way.
- Knowledge of data and skills to assess it. How is it set up, how should it be interpreted, what problems usually exist with data, what ways are there to solve this.
- Basic knowledge of AI / ML / DL, knowledge of different ways of modeling. Which models are relevant, of course, depends on the project. Image recognition is different from pattern recognition in datasets, models that generate something are also different.
- Knowledge of programming languages and frameworks commonly used in AI. These can be tools for dealing with data, but also tools for modelling. Sometimes this is based on Python or R programming languages, sometimes with the help of libraries such as TensorFlow, Pytorch, Keras, Pandas and Scikit-learn. Possibly supplemented with knowledge of platforms that offer modeling or even AutoML, such as Microsoft Azure, Google Cloud, Amazon AWS and IBM Watson.

The role of 'tester' is therefore a very flexible concept. If one still thinks of the traditional test role when thinking of 'tester' in an AI project, it can occur that quality assurance and asking the really critical questions are considered too late in the project. This white paper is deliberately called 'Quality and Testing of AI', not just 'Testing of AI'. Moreover, we deliberately use the term QA specialist next to the term 'tester'.

A new skill can be developed when you position yourself as a tester on projects while AI is being developed. At the moment it is not yet self-evident that testers will become involved (in time). But that doesn't mean it can't be done!

8.4 Fulfilling the quality role together

Using testers and QA specialists in AI projects is still in its infancy. Because it is relatively new, it offers opportunities to develop yourself. It helps if you already have experience in previous AI projects. But if you are at the beginning of such a project, the knowledge of others can help you. We consider the role and place of tester or QA specialist in an AI project to be of added value.

And as a group, testers and QA specialists can jointly fulfill the quality role within AI, which can include the use of new tools and techniques, testing ethics, developing best practices and courses for further training. This approach makes it possible to jointly emphasize the quality and testing in an AI project, which will speed up the adoption of AI solutions.

This white paper is therefore a call to all testers and QA specialists to find each other, share successes, and learn from mistakes. We like to seek out cooperation through communities such as TestNet and EuroSTAR. If you have ideas about AI testing or if you already have experience with it, we cordially invite you to share your insights and feedback with us. Contact details can be found on the last page.

Appendix A: Resources

General

- Human Compatible- Stuart Russell, 2019
  https://en.wikipedia.org/wiki/Human_Compatible
- The Next Decade in AI - Gary Marcus, 2020
  https://arxiv.org/ftp/arxiv/papers/2002/ 2002.06177.pdf
- Rebooting AI - Gary Marcus and Ernest Davis, 2019
  http://rebooting.ai/
- What makes AI testing different - Peter Collewijn (TestNet), 2020
  https://www.testnet.org/testnet/download/common/2020 -11-what-makes-ai-testing-different.pdf
- Hidden Technical Debt in Machine Learning Systems, D. Sculley, 2015
- The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction, Eric Breck, 2017
- Testing machine learning based systems: a systematic mapping, Vincenzo Riccio, 2020
- Why Should I Trust You?", Explaining the Predictions of Any Classifier, Marco Tulio Ribeiro, 2016
- Combinatorial Testing for Deep Learning Systems, Lei Ma, 2018
  https://arxiv.org/abs/1806.07723
- Effective testing for machine learning systems, Jeremy Jordan, 2020
  https: //www.jeremyjordan.me/testing-ml/
- A Software Testing View on Machine Learning Model Quality, Christian Kästner, 2020
  https://ckaestne.medium.com/a-software-testing-view-on-machine-learning- model-quality-d508cb9e20a6
- Continuous Delivery for Machine Learning, Danilo Sato, 2019
  https://martinfowler.com/articles/cd4ml.html
- Snorkel Intro Tutorial: Data Slicing
  https://www.snorkel.org/use-cases/03- spam-data-slicing-tutorial
- Towards Robust and Verified AI: Specification Testing, Robust Training, and Formal Verification, Deepmind, 2019
  https://deepmind.com/blog/article/robust-and-verified-ai
- Interpretable Machine Learning, Christoph Molnar, 2020
  https://christophm.github.io/book/
- Metamorphic Testing of Machine-Learning Based Systems, Teemu Kanstrén, 2020
  https://towardsdatascience.com/metamorphic-testing-of-machine-learning-based-systems-e1fe13baf048
- Discrimination, AI and Algorithmic Decision-Making, Frederik Zuiderveen Borgesius, 2018, https://rm.coe.int/discrimination-artificial-intelligence-and-algorithmic-decision-making/1680925d73
- Cat selection by Cassy Kozyrkov
  https://towardsdatascience.com/in-ai-the-objective-is-subjective-4614795d179b
- Decision trees versus neural networks
  http:// www.312analytics.com/decision-trees-vs-neural-networks/

Quality standards

- Testing in the digital age, Tom van de Ven, Rik Marselis & Humayun Shaukat, Sogeti Nederland BV, 2018, ISBN: 978-90-75414-87-5
- Quality and AI-based Systems with Adam Leon Smith (ISO 25059) https://youtu.be/OadJbNeTmiY
- Artificial Intelligence – Life Cycle Processes and Quality Requirements (DIN SPEC 92001-1)

Ethics

- AI no longer has a plug, Rudy van Belkom https://detoekomstvanai.nl/artikelen/ai-heeft-geen-stekker-meer/
- Ethics guidelines for trustworthy AI https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy- ai
- Regulatory framework proposal on Artificial Intelligence https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai

Checklist

- Checklist for Data Science Research Review https://medium.com/@ptannor/checklist- for-data-science-research-review-8a817b50697b
- Checklist for Artificial Intelligence in Medical Imaging (CLAIM) https://pubs.rsna.org/d oi/10.1148/ryai.2020200029
- AI Checklist Cards https://www.tmforum.org/resources/reference/ai-checklist-cards/
- Chatbot test https://chatbottest.com/

Assessments

- EU: https://futurium.ec .europa.eu/en/european-ai-alliance/pages/altai-assessment-list-trustworthy-artificial-intelligence
- ECP: https://ecp.nl/publicatie/artificial-intelligence-impact-assessment-nieuwe-versie/ https://ecp.nl/publicatie/artificial-intelligence-impact-assessment-english-version/
- Pair: https://pair.withgoogle.com/

Courses

- Ai United Certified Tester in AI (CTAI) https://www. ai-united.org/
- A4Q, AI and Software Testing https://www.alliance4qualification.info/a4q-ai-and-software-testing
- KSTQB & CSTQB, Certified Tester AI Testing (Testing AI-Based Systems) https:// imbus.cn/upFile/Uploadfiles/AI%20Testing_Testing%20AI-Based%20System%20Syllabus%20v1.3.pdf
- Datascience Academy, Testing & Monitoring ML Deployments https://data-science-academy3.teachable.com/p/testing -monitoring-machine-learning-model-deployments

News Articles

- Uber Must account driver recovering after accusations of fraud
  https://nos.nl/artikel/2376697-uber-moet-account-chauffeur-herstellen-na-beschuldiging-fraude.html
- Tesla's banned by Chinese military bases because of concerns about espionage
  https: / /www.nu.nl/tech/6122885/teslas-geweerd-bij-chinese-legerbases-wegens-bezorgdheid-over-spionage.html
- Report: British police face fails in four of five cases
  https://tweakers.net/nieuws /154870/report-facial-recognition-british-police-failure-in-four-of-five-cases.html
- Smokescreen around fraud system Nissewaard
  https://webcache.googleusercontent.com/search?q=cache:VHqc2uuOkTMJ:https://www.binnenlandsbestuur.nl/sociaal/nieuws/rookscherm-rondom-fraudesysteem-nissewaard.13026523.lynkx+&cd=4&hl=nl&ct=clnk&gl=nl
- Microsoft speaks of coordinated Tweet attack against chatbot
  https://webcache.googleusercontent.com/search?q=cache:8f1teYguutsJ:https://tweakers.net/nieuws/109711/microsoft-speaks-of-coordinated-tweet-attack -tegen-chatbot.html+&cd=7&hl=nl&ct=clnk&gl=nl
- Google just gave a stunning demo of Assistant making an actual phone call
  https://www.theverge.com/2018/5/8/17332070/google-assistant -makes-phone-call-demo-duplex-io-2018recognition-een
- The impact of facial recognition: a face as evidence of crime
  https://www.nu.nl/tech-background/6121506/de-impact-van-face-recognition-een-face-as-proof-for-criminality.html
- Anti-fraud system SyRI must be off the table, government invades private life
  https://nos.nl/artikel/2321704-anti-fraudesysteem-syri-moet-van-tafel-overheid-maakt-inbreuk-op-priveleven.html
- Search algorithms for welfare fraudsters, what role does ethnic profiling play
  https://nos.nl/artikel/2366962-algorithms-search-for-bijstand-fraudeurs-which-rol-plays-ethnic-profileren.html
- GPT-3 Facts
  https://en.wikipedia.org/wiki/GPT-3
- Amazon scraps secret AI recruiting tool that showed bias against women
  https://www.reuters.com/article/us-amazon-com -jobs-automation-insight-idUSKCN1MK08G
- Top 10 Reasons Why 87% of Machine Learning Projects Fail
  https://dzone.com/articles/top-10-reasons-why-87-of-the-machine-learning-proj
- Why So Many Data Science Projects Fail to Deliver
  https://sloanreview.mit.edu/article/why-so-many-data-science-projects-fail-to-deliver/?utm_source=ONTRAPORT-email-broadcast&utm_medium=ONTRAPORT-email-broadcast&utm_term=&utm_content=Data+Science+Insider%3A+March+5th%2C+2021&utm_campaign=06032021
- Google assistant phones hairdresser
  https://www.youtube.com/watch?v=JvbHu_bVa_g

Appendix B: Glossary

Term	Explanation with links to text in the chapters
A/B testing	See Chapter Testing AI
Acceptor	See Chapter Testing AI in practice
Adversarial testing	See Chapter Ethics
Amazon AWS	Cloud environment of Amazon
AutoML	See General risks of challenges for testers
Bias	See appendix C
Bias Trade off	See Appendix C
Boundary Value	See Boundary Value Analysis
Configuration	See Testing the model
Consequence Ethics	See Chapter Ethics
Convolutional Neural Network (CNN)	See Chapter 3 Image recognition
Corner test	See Boundary Value Analysis
Data combination test	See Equivalence Partitioning
Data scientist	See Roles in an AI project
Deep fake	See Image generation
Deep learning	See Chapter Artificial Intelligence
DIN SPEC 92001-1	See Chapter Quality attributes
Directional Expectation Test	See Chapter Testing of AI, Metamorphic testing
Drift	See Chapter Testing of AI, Testing for Drift
Equivalence Partitioning	See Chapter Testing of AI, Equivalence Partitioning
Expert panel test	See Chapter Testing of AI, Expert Panel Testing

Term	Explanation with links to text in the chapters
Explainable AI	See Appendix D
Exploratory testing	See Chapter Testing of AI, Experience-based Testing
Extrapolation	See appearance Regression
FLOPS	See Text Generation FLOPS is a unit used to indicate the computing power of CPUs. https://en.wikipedia.org/wiki/FLOPS
GAN	https://en.wikipedia.org/wiki/Generative_adversarial_network
Google Cloud	Cloud environment of Google
Heatmap	see Chapter Testing of AI, Testing the model see appearance Image Recognition
High-Level Expert Group on Artificial Intelligence	See Chapter Ethics
IBM Watson	Cloud environment of IBM with AI applications and tools
Input variables	See Chapter Artificial Intelligence also called input variables
Invariance Test	See Chapter Testing of AI, Metamorphic testing
ISO 25010	See Chapter Quality attributes. More information: https://iso25000.com/ index.php/en/
Keras	Software library with frequently used ML functions
Linear regression	See Appendix C
Metamorphic testing	See Chapter Testing ai, Metamorphic testing
Microsoft Azure	Cloud environment Microsoft
ML engineer	See Testing AI in practice
Neural Network	See Chapter Artificial Intelligence
OpenAI	See Text generation
Outliers	See Dependence on data

Term	Explanation with links to text in the chapters
Overfitting	See Appendix C
Pairwise testing	See Chapter Testing AI, Equivalence Partitioning
Persona	A persona is a type of a user, or a characterization of a certain type of user. See Testing using personas
Polynomial regression	See Appendix C
Preprocessing	See Testing the Data
Principle Ethics	See Chapter Ethics
Python	Programming language often used for ML development
Pythorch	Deep Learning Framework for configuring DL models
Quality attributes Quality features	See Chapter Quality attributes
Quality standards	See Appendix A: Resources
R or Rstudio	Programming language often used for ML development especially in an academic environment
Regression	See appearance Regression
Regression testing	See Chapter Testing of AI, Regression testing
Robustness	See Chapter Ethics and Chapter Quality attributes
Scikit-learn	Software library with frequently used ML functions
Sequence recognition	See chapter Appearances.
TensorFlow	Deep Learning Framework for configuring DL models
Test data	See Appendix C
Test oracle	See Chapter 1
Testing a model	See Chapter Testing of AI
Testing from personas	See Chapter Testing functionality
Testing on Drift	See Chapter Testing AI,Drift

Term	Explanation with links to text in the chapters
TMAP	See Chapter Degrees of Autonomy and www.TMAP.net
Token	See Text Generation
Training	See Appendix A
Training Data	See Appendix C
Transfer Learning	Transfer Learning is the reuse of a pre-trained model on a new problem. It's currently very popular in deep learning because it can train deep neural networks with comparatively little data. https://builtin.com/data-science/transfer-learning
Underfitting	See Appendix C
User Story test Use Case test	User Story testing – Use Case Testing
Variance	See Appendix C
Virtue Ethics	See Chapter Ethics

Appendix C: A piece of technology

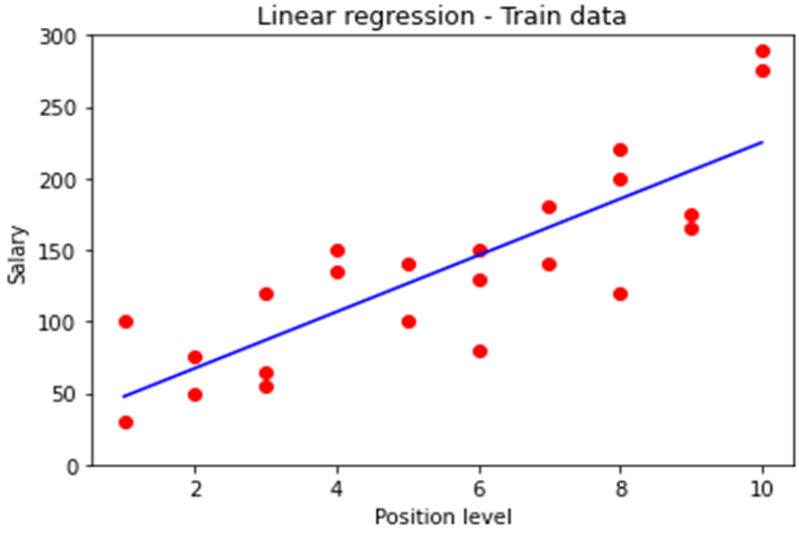
A Machine Learning algorithm is based on examples, data that has resulted in a certain result in the past. Those examples are, by definition, only part of reality. To develop a Machine Learning model, you need (a lot of) data. The more data, the better the chance that an algorithm can be made that makes a good prediction of the examples that we have provided the model with.

In the accompanying example, we use the salary compared to the salary scale as data.

It is common to divide this data into two parts:

- A part that is used to train the model (red)
- A part that is used to test the model (green)

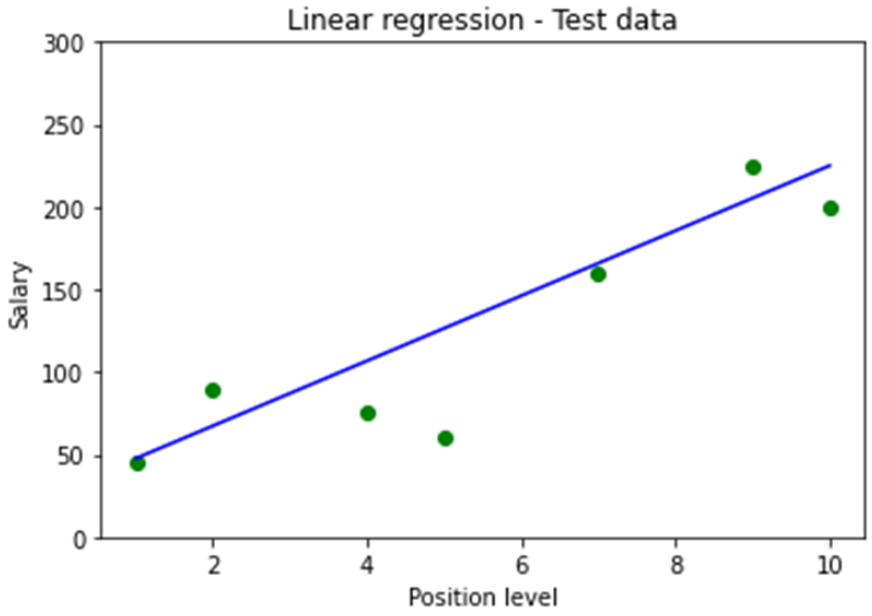
After training the model, it is tested how well, how accurate, the model is. This is a kind of unit test on the model performed by an ML engineer. In our example, the model produces a straight line and based on the training data, the red dots, we get the following graph:



This line predicts approximately 70% accurately the pattern of the training data.

However, this data is based on training data, the model already knows the outcome of this. It is therefore important to know how well this model performs based on the test data. After all, that is an indication of how well the model will work in a production environment.

In the following graph, the model, the line, is plotted against the test data.



The model appears to perform even better on the test data. After calculation, the model appears to accurately predict the pattern of the test data for 78%.

Of course, the goal is to get as close to 100% as possible. The difference between this 100% and the actual accuracy (see this as the error) is called **Bias**.

The aim is to reduce this bias as much as possible and thus to make the difference between an arbitrary prediction and the correct prediction of our examples as small as possible. With this we limit **Underfitting** (i.e., we ensure that the model fits the examples better).

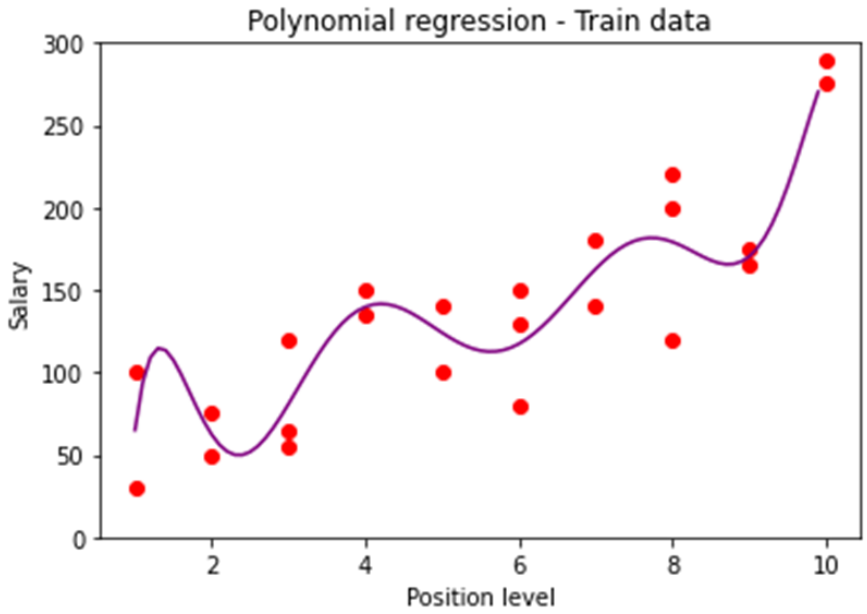
So, in our example there is a 22% bias (error) on the test data, and 30% bias on the training data.

The difference in bias between these two data sets is called **Variance**. In our example, the variance is 30% - 22% = 8%.

As indicated, the aim is to develop a model that comes as close as possible to 100% accuracy. The fact that the model already performs better with the test data is also an indication that there is room for improvement. The model is now **Underfit**.

We can try another training method.

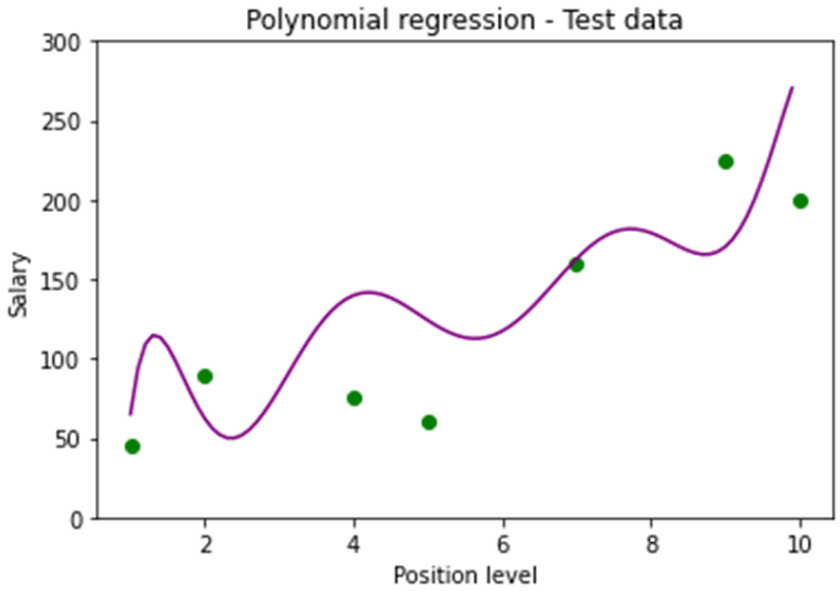
Due to this different approach, the model now predicts a line that better follows the pattern of the training data. See the graph of the training dates below.



This model has an accuracy of 84%. This is a big improvement compared to the 70% of the first model.

However, the question is again: how well does the model perform on the test data?

That is unfortunately disappointing, see the proof in the graph below:



The accuracy is now only 38%.

In a situation such as in this example:

- the model performs well with the training data, but
- the model performs poorly with the test data,

one speaks of a model that is **Overfit**.

Compare it to memorizing all the answers of a mock exam. That is usually also not a good learning method to pass the actual exam.

The variance of this model is also very large. The difference is now 84% - 38% = 46%.

So, we must be careful not to shape the algorithm too much after the examples we have. After all, we want the algorithm to generalize well so that it also scores well on the cases that we have not used as an example. This reduces the variance (the difference between the prediction of the chosen examples and the prediction of the complete reality) and we limit overfitting (the model mainly predicts the examples well, but is less good at predicting the complete reality).

Conclusion:

We are therefore looking for a model with the highest possible accuracy. In other words, the smallest possible bias.

Also, we want this accuracy to be generic, so that there is little difference in this accuracy across different data sets. So, the smallest possible variance.

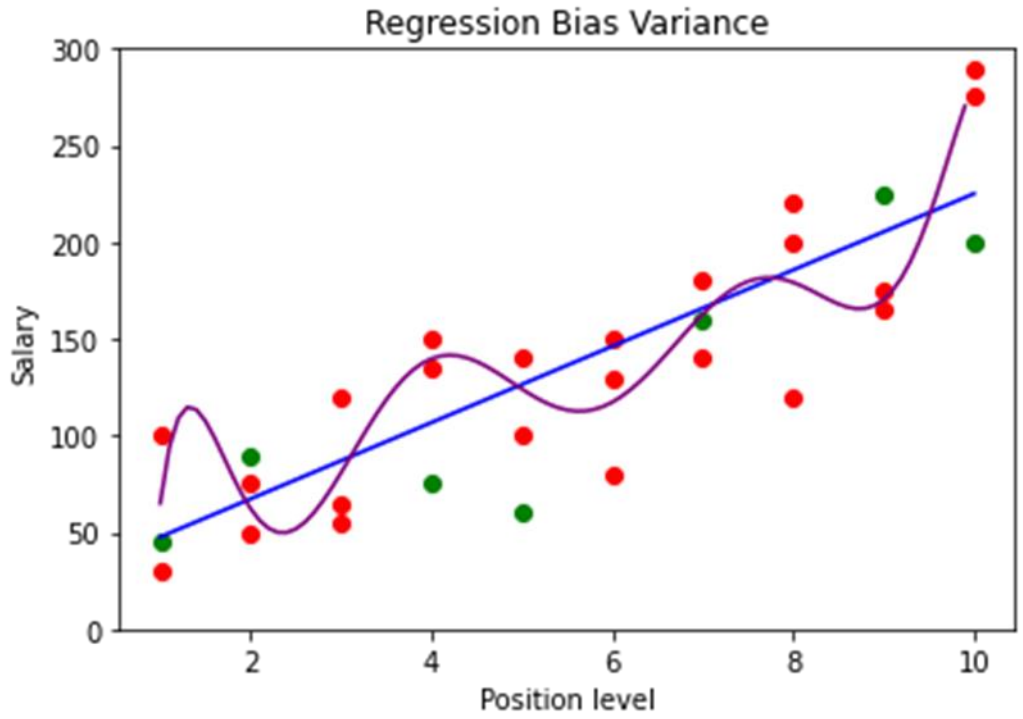
If you train a model better or longer, it is possible that the accuracy of the training data will improve. The bias of the model on these data decreases.

However, often the difference with the accuracy of the test data becomes greater. So, the variance gets bigger.

This phenomenon is called the **Bias-Variance Trade Off**.

The goal is therefore to make a model with the highest possible accuracy and that has a comparable accuracy with both the training and test data.

After all, that is a good indication of how the model will perform in production.



Appendix D: Risks and testing activities

The risks and points of attention of AI development have been discussed in various chapters. In the chapter ‘Testing of AI', it is indicated step-by-step which quality and testing activities contribute to removing the risks. This appendix describes a link between risks and testing activities.

Uncertain outcomes

Getting a grip on uncertain outcomes can partly be done with existing techniques such as Equivalence Partitioning and Boundary Value Analysis, which can be used to gain clarity about grey areas in predictions. This enables domain experts to determine which results the model should give in each of the defined situations. A persona's test can also provide insight into the extent to which the outcome depends on the group the persona represents.

Dependency on data

Testing data remains one of the key points in AI testing. Data can be assessed in the way it was collected, selected, edited and fed into the model. Ethical considerations are often of great importance in this regard. The white paper has devoted a chapter to ethics, in the expectation that this will help you to translate the ethical considerations into your own practice.

It is also good to realize that Equivalence Partitioning, Boundary Value Analysis and other techniques for data analysis can also be applied to the collected (input) data. Here too, experts can help to assess this data for correctness in terms of content and mutual relationship.

Limited explainability

Machine Learning (ML) models, in particular the Deep Learning (DL) models, will always prove difficult to explain, but there is always the possibility to partially interpret them. Explaining AI, or eXplainable AI (XAI), is an area that is still developing. From a traditional test perspective, it is obvious to apply at least a number of black box test design techniques and approaches such as use case testing, exploratory testing or personas. Metamorphic testing is also possible, because monitoring logical relationships between input and output is very important for comprehensibility and thus explainability. Whitebox test design techniques will surely add value too, for as far as it is possible to use them. Consider the example in this white paper about the Dakar car in the sand.

This type of testing increases confidence that the AI model functions comprehensibly and ensures acceptance of its use.

Changing reality or need

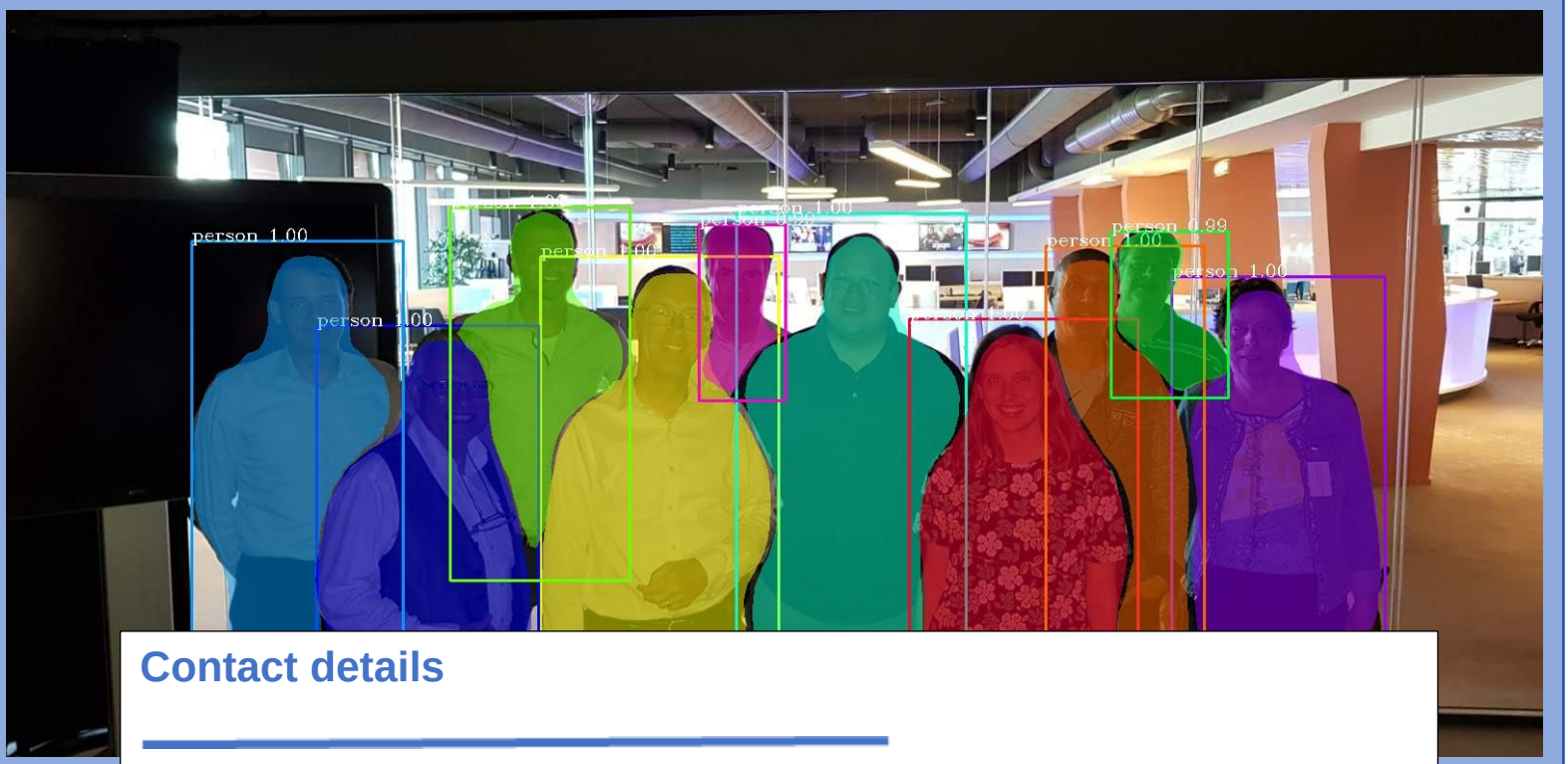
The changing reality or need is known as drift and is briefly described in the chapter 'Testing of AI'. With the help of A/B testing it is possible to gain insight into this risk of changing behavior.

These tests involve monitoring, such as a change in the ratio of the input data or the accuracy of the prediction. This has been discussed in limited detail in this white paper, but it goes without saying that monitoring ensures that the AI model remains acceptable in practice. Since different groups of stakeholders find different scores important, testing with personas can help to keep all stakeholder groups happy.

A/B testing can also be used to check an AI model in practice for changing needs. This is of course also a form of behavior, but in this case it shows that the interest in general has shifted, for example if people suddenly click on a certain category of films on a video platform. These are subtle changes in user needs. If a major change in needs arises from within the organization, for example the above example about recognizing other types of window damage, then regression testing is an important activity.

General fear of AI

New techniques almost always lead to feelings of fear in the beginning. The most important activity to take away fear is to give the right insights. One of the EU's goals is to encourage the use of AI while allaying fears by proposing ethical guidelines and eventually making them mandatory. To gain insight into how the model works, checklists and assessments have been drawn up for these guidelines. Completing these checklists and assessments will at least provide some comfort. Exploratory testing and testing with personas can also play a role in gaining insight into how an AI model works. In fact, every test that is performed contributes to increasing this insight and thus contributes to reducing the risk of fear of AI.



Contact details

Working group Testing and AI

Sander Mol
Peter Collewijn
Hannie van Kooten

email: ai.workgroup.testnet@gmail.com
site: https://www.testnet.org/testnet/p000610/werkgroepen/werkgroep-testen-en-ai

TestNet

site: https://www.testnet.org/testnet/home