



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Home Affairs FDHA
Federal Statistical Office FSO

The Data Innovation Project “NOGAuto”

SWISS FEDERAL STATISTICAL OFFICE

EXPERIMENTAL STATISTICS



Neuchâtel, 2021

Published by: Federal Statistical Office (FSO)
Information: cindia.ducsfez@bfs.admin.ch, tel. +41 58 463 63 37
Editors: Business Register Data section (BRD)
Contents: Lorenz Helbling, Prof. Dr. Diego Kuonen
Topic: 00 Statistical Basis and Overviews
Original text: English

Layout concept: Section DIAM
Downloads: www.experimental.bfs.admin.ch
Copyright: FSO, Neuchâtel 2021
Reproduction with mention of source authorised
(except for commercial purposes)



The Data Innovation Project “NOGAuto”

SWISS FEDERAL STATISTICAL OFFICE

Helbling Lorenz

GSEM Supervisor: Prof. Dr. Diego Kuonen

May 24th, 2021

Executive Summary

The work is a Master's thesis done by Lorenz Helbling in partial fulfillment of the Master of Science in Business Analytics at the Geneva School of Economics and Management (GSEM) of the University of Geneva. This Master's thesis was supervised by Prof. Dr. Diego Kuonen (Statoo Consulting & GSEM, University of Geneva). This document describes how the “NOGAuto” machine learning project was conducted during the ten months of internship at the Swiss Federal Statistical Office (FSO). It highlights the importance of applying a well-defined process and a proper understanding of the business context. Furthermore, the different machine learning methods will be described to understand how the related prediction system was created.

Table of contents

1. Introduction	1
1.1. Context Presentation.....	1
1.2. Project Description.....	1
2. Data Innovation as a Process.....	2
2.1. Theoretical Fundamentals	2
2.1.1. Prototype (CRISP-DM).....	3
<i>Business understanding</i>	<i>3</i>
<i>Data understanding</i>	<i>5</i>
<i>Data preparation</i>	<i>5</i>
<i>Modeling</i>	<i>5</i>
<i>Validation</i>	<i>6</i>
2.1.2. Minimal viable product (MVP).....	6
<i>Realising</i>	<i>6</i>
<i>Continuous testing</i>	<i>7</i>
<i>Architecture adjustment & design adjustment</i>	<i>7</i>
2.1.3. Production.....	7
<i>Release.....</i>	<i>8</i>
<i>Operating activity</i>	<i>8</i>
<i>Monitoring</i>	<i>8</i>
2.2. Applied to the “NOGAuto” Project	9
2.2.1. Machine learning prediction system	10
<i>Predictive accuracy.....</i>	<i>10</i>
<i>Documentation.....</i>	<i>10</i>
<i>Implementation of a monitoring</i>	<i>11</i>
2.2.2. Domain user.....	11
<i>Interface</i>	<i>11</i>
<i>Performance</i>	<i>12</i>
<i>Daily use</i>	<i>12</i>
<i>Added value for the user.....</i>	<i>12</i>
3. “NOGAuto” Machine Learning System	13
3.1. Data Collection.....	13
3.2. Data Preparation	16
3.2.1. Language detection.....	17
3.2.1.1 R packages	17
3.2.1.2 Root cause analysis	18
<i>Recurrent words</i>	<i>19</i>
<i>Association rules.....</i>	<i>19</i>
<i>Mosaic plots.....</i>	<i>20</i>
<i>Correspondence analysis</i>	<i>21</i>
3.2.2. Text mining & natural language processing.....	23
3.2.2.1 Data cleaning	24

3.2.2.2 Document representation	24
<i>One-hot encoding</i>	25
<i>Word2Vec</i>	25
<i>Text2Vec</i>	26
3.2.2.3 Evaluation and validation	27
<i>t-SNE</i>	27
<i>Word cloud</i>	29
3.3. Modeling	29
3.3.1. Training	30
<i>caret</i>	31
<i>Hyperparameters determination</i>	31
<i>GBM</i>	31
3.3.2. Prediction	32
4. Lessons Learnt	33
5. Appendix	36

1. Introduction

1.1. Context Presentation

The Swiss Federal Statistical Office (FSO), as part of the Federal Department of Home Affairs, is the main office providing national statistics on population, economy, society, education, research, territory and environment. This valuable information provide the population and the government with a more accurate overview of the situation and development in Switzerland. In addition, these statistical indicators can help and guide various decisions in politics and other diverse areas.

In 2017, the FSO released its data innovation strategy to ensure to keep up with time and technology. The “NOGAuto” project, which has been submitted by the Business Register Data section (BRD), was one of the main five projects, which has been retained in line with FSO’s data innovation strategy with the goal to augment and/or complement the existing basic official statistical production at the office. The “NOGAuto” project consists in the creation of a prediction system that predicts companies’ NOGA codes¹ (categories) using machine learning methods.

1.2. Project Description

The General Classification of Economic Activities (NOGA) is an essential element of statistical production at the FSO. The quality of the NOGA codification of enterprises recorded in the Business and Enterprise Register (BER) has a direct impact on the results of structural, cyclical and synthetic statistics concerning enterprises. These depend on a stable, controlled and quality conscious NOGA coding of the units in the BER. In the interest of discharging enterprises and continuously improving the coding of enterprises, the “NOGAuto” project aims to automate the assignment or, more precisely, the prediction of economic activity codes to enterprises, based initially on the information already available within the FSO.

Currently, the allocation and control of NOGA codes is executed manually, which leads to discrepancies in the interpretation (“human factors”) of the economic activities of enterprises and thus creating instability in the codification quality of NOGA codes. This may cause a decrease in the overall quality of the various statistical outputs.

The creation of such a “prediction system”, that can automatically classify the NOGA codes of enterprises registered in the BER using machine learning techniques, should have a coding quality equal to or better than the manual coding currently conducted.

Through the “NOGAuto” project, the Business Register Data (BRD) and BER sections as well as all the FSO statistical users could benefit from:

- a “tool” which automatically verifies the NOGA codes of enterprises in the register;
- a standardisation of the codification;

¹ A NOGA code enables the statistical units to classify “enterprises” and “establishments” by their main economic activity and categories them into coherent groups. This classification allows, for example, a comparability of our economic structure at national and international level.

- a minimisation of the interpretation factor;
- an optimisation of the time spent on codification;
- a quality improvement of the NOGA coding and consequently of all enterprise statistics.

To have a better understanding on how the “NOGAuto” project evolved during the ten months of internship, the first part will explain the importance of applying a well-defined process to the project by showing the importance of understanding the business. The second part will describe the different steps that were carried out to arrive at a working prediction system. Finally, in the “lessons learnt” chapter, some of the highlights of the internship will be discussed such as the presentation of the “NOGAuto” project at the European New Techniques & Technologies for Statistics (NTTS) 2021 conference. In addition to this, the possible next steps of the project will also be explained.

2. Data Innovation as a Process

Nowadays, organisations launch data science projects to improve strategic perspectives such as reducing costs, optimising processes or even trying to conquer new business areas. However, most projects fail to be deployed in some organisations because the “right” questions are not asked on the “right” problems at the beginning of the project. Therefore, a data science project can not only be focused on the collection of data and the application of machine learning and artificial intelligence methods, but on its overall impact in the organisation.

Thus, it is essential to adopt a process that embraces all the challenges that a project may face during its implementation. The data innovation process, which has been applied to the “NOGAuto” project, will be detailed first from a theoretical point of view followed by how it has been implemented by the team, composed of data scientists, business specialists and coders.

2.1. Theoretical Fundamentals

First, to understand how the project was conducted, it is essential to subdivide each phase and each step contained in it. By doing this, all the involved collaborators from different backgrounds will be well informed about how the project will progress and through which phases it will pass. This will have a consequential benefit on communication and coordination between collaborators, as they will have a transparent and similar project deployment knowledge. This is especially true when this process is applied for the first time.

Below, in Figure 1, we can see the overall data innovation process and its workflow, which is divided into three different phases:

- “Phase 1”: “Prototype (CRISP-DM)”;
- “Phase 2”: “Minimal Viable Product (MVP)”;

- “Phase 3”: “Production”.

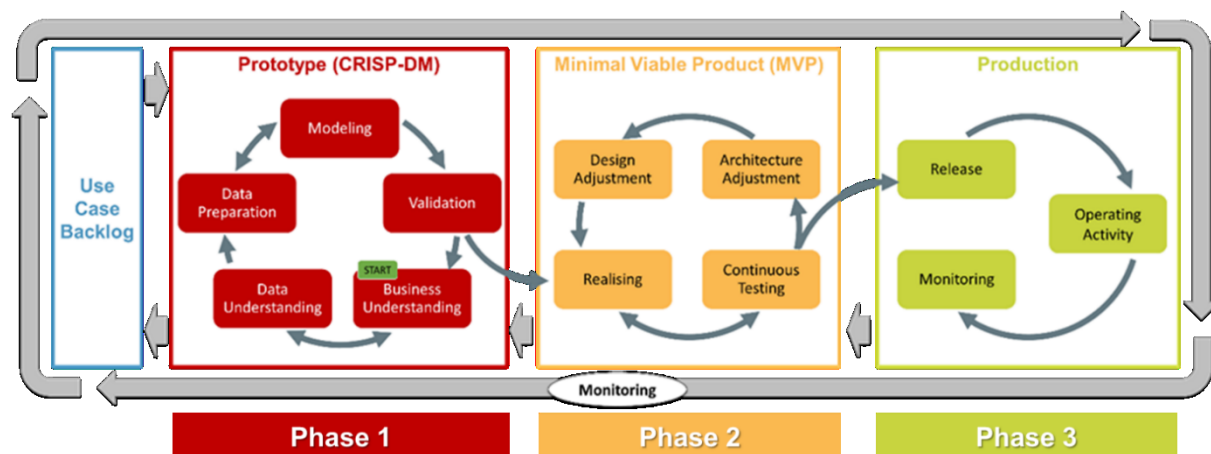
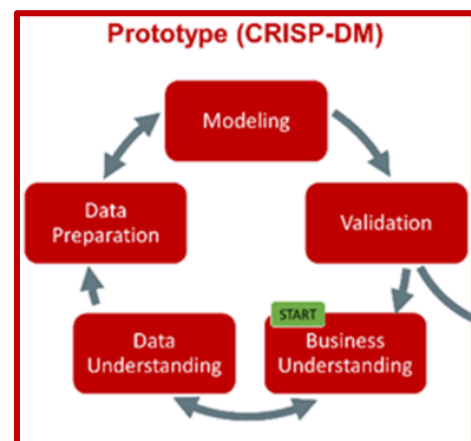


Figure 1: Statoo Consulting, Switzerland

It is important to note that the three phases are interconnected allowing a continuous improvement of the project spread over time with the help of a monitoring. However, for the project to proceed to the next phase, corresponding success criteria must be determined already at the beginning of the project. If these are fulfilled, the project can then progress to the next phase. At this point, the “NOGAuto” project is at the MVP stage. Indeed, because a whole feasibility work of the project has been performed and has shown that a prediction system of NOGA codes is possible.

2.1.1. Prototype (CRISP-DM)

The project begins with the “Prototype (CRISP-DM)” phase which uses the CRoss Industry Standard Process for Data Mining (CRISP-DM) framework. “*This specific framework was created in 1996 by European data mining experts to set up some standards to projects subject to big data problematics*”² (Kuonen, 2018; Slide 81). In the beginning of this process, the project will follow five steps that ensure the good understanding of the business context and problem as well as “*consistency, repeatability and objectivity*”³ (Kuonen, 2018; Slide 81) of this one.



Business understanding

As a first step in the “Phase 1” process, the business understanding step is probably the most important of all because it represents the heart and the essence of the project. Without a clear identification of the problem, the project may face unforeseen obstacles and may never be completed. Here, the team will already need to think about what the product might look like in

² Kuonen, D. (2018). An Introduction to Analytics, Data Science, Machine Learning and Supervised Learning. Course slides of the training on May 1, 2018, for data innovation pilot teams at the FSO.

³ Ibid.

the MVP (“Phase 2”) and production (“Phase 3”) phases. In addition, it is also necessary to think about the type of data and machine learning and artificial intelligence methods they will need throughout the project.

However, to better determine and assess the problematic within the organisation, the team members must follow the clear structure showed in Figure 2.

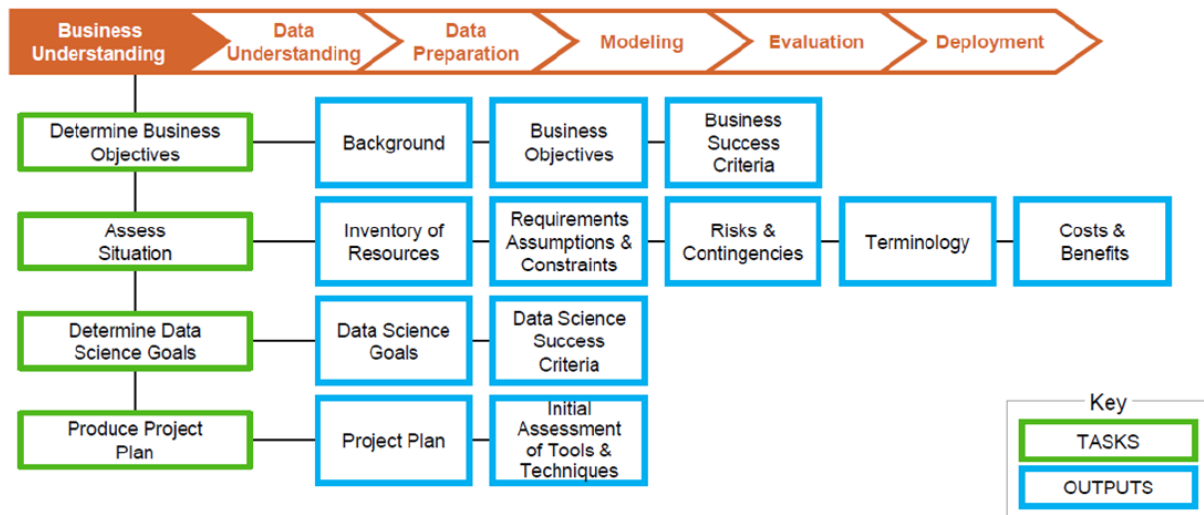


Figure 2: Statoo Consulting, Switzerland

As a starting point, to be able to determine the business objectives, it is imperative to collect background information by identifying the reasons of the problem, the people/units concerned and the current method used. This information will make it easier for the team members to determine the business objectives of the project and the key business success criteria that will take the project through the various phases of the data innovation process. If these are well defined, it might lead to a successful project with a business impact in the organisation.

In a second step, it is important to assess the situation by doing an inventory of the available resources such as the people who will be allocated to this project, from where the data will be collected and on which system/software the whole will be processed/developed. Afterwards, the team should set the project requirements, such as the different timeframes for each phase and how the results could be presented at that time. Next, a clarification of the various assumptions and diverse constraints need to be completed. Here it is meant to check whether the project could potentially face legal, ethical, financial or privacy obstacles. The team will also need to carry out a risk assessment to cover possible contingencies and a cost-benefit analysis to determine the cost-effectiveness of the project. Finally, to guarantee that everyone in the project understands all the terms that are used, a common terminology should be agreed upon. This last point shows the importance of the interdisciplinary collaboration and the gap connection between the various professional competences.

As it is a data innovation project, it is crucial to determine quantitative measures that will represent the data science goals such as a level of performance of the models used (e.g. predictive accuracy, etc.). Such performance metrics will define whether the developed system/model has achieved the objectives that have been set beforehand and will allow it to move on to the next phase according to the business success criteria.

The last point that the team must complete for the business understanding part is to produce a project plan, which enumerates the different tasks that have to be succinctly completed to achieve all the objectives set. In addition, an initial discussion about a monitoring of the data, methods and the product that are closely linked to the business objectives must have taken place - a monitoring which will be then sequentially developed in the first two phases and implemented in the production phase.

Data understanding

After describing the problem and the environment in which the project will evolve, it is time to understand what data will be collected. Here it is important to know the source of the data and to determine the degree of trust in it. First, a selection of the needed variables must be made for the data collection part. Second, a description of the collected data must be completed, such as the dimension of the dataset, the type of variables and what they are representing for the observations. The third task is the data exploration, which is likely to be one of the most interesting one in the data understanding step. Indeed, it will allow the team to learn more about the dataset, to have a first insight in the data and to have a better overview of it. This can be achieved by using charts or tables and to describe the outcomes from it. This will give the collaborators a first impression of the results that might appear in the future and might raise some new hypothesis. Before heading to the data preparation step, it is fundamental to do a data quality control in doing some checks such as detecting missing values, errors or anomalies in the dataset. If it is the case, it would be capital to correct them in the data preparation step to avoid errors for the good continuation of the project. Finally, this data understanding step could also be followed by the need to do a further iteration back to the business understanding step (as e.g. the data quality for the questions at hand may not be good enough).

Data preparation

Once the data have been collected, described and checked for quality, a selection or filtering of variables should be made, giving reasons why a variable is excluded from or included in the project. Following this, the retained variables need to be cleaned up to meet the expected quality and to correct the errors that have been detected in the data understanding section. Moreover, by fine-tuning the selection of variables, new ones can be formed using text mining as well as feature engineering methods to gain additional information and to add new content to the dataset. Before building prediction models, it is imperative to check the data format so that the system is correctly set for the following steps (e.g. if the variable is a text, then the type should be "character").

Modeling

In the modeling step, the collaborators who are in charge of the modeling must first define which methods and quality measures would or could be applicable by explaining their methodologies according to the expected outcome and objectives that the team has defined earlier in the business understanding step.

Subsequently, the training of the models and the fine-tuning of the diverse parameters can be performed on the prepared structured data. Thanks to the quality measures that the prediction system provides, it will be possible to determine the "best" model with the "best"

hyperparameters that have to be applied on the final model. However, it is important to report all steps and results to the collaborators so that they understand the chronology of the modeling and the determination of the final model. For the modeling process to be completed, the final model with the optimal parameters must be run and prepared for the subsequent validation stage. Here the team should receive the results in a predefined way such as the generalisation error and the confusion matrix when talking about classification or, for example, the mean square error for regression.

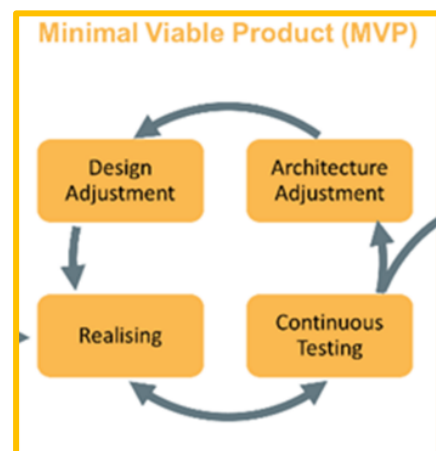
Validation

Finally, in the validation step, the business specialists and the coders of the team will evaluate the outputs and results that the prediction system computed. The final model of “Phase 1” (i.e. “Prototype (CRISP-DM)”) will be validated using the different measures of success as well as the objectives that have been determined in the business understanding stage at the very beginning of the phase and a detailed check will be performed whether they are fulfilled or not. They will also be able to give their feedback by filling out a questionnaire to give their remarks from a qualitative point of view. However, if the objectives are not met, the team will have to recheck the process and modify it to improve the prediction system until all the success criteria of “Phase 1” are met. Once the prototype meets the objectives/the expectations and moves in the “right” direction of solving the main business problem, one can proceed to the next phase: “Phase 2: Minimal Viable Product (MVP)”.

2.1.2. Minimal viable product (MVP)

After the prototype has been evaluated and validated by the team members, the project moves on to “Phase 2”, where the first model proposition will be turned into a Minimal Viable Product (MVP). There, the prototype will be improved and adjusted through a four-step cycle process so that it becomes robust and efficient enough to be deployed within the organisation within the “Phase 3: Production” (if needed).

Moreover, it will be particularly important to determine whether the hypothetical ideas of what the product might look like at the very beginning could become a reality and that the product can be proposed to the business team.



Realising

For the first step in the second phase, the model will be applied to a larger dataset to see if the model performs as well as it did in the prototype phase. However, it is not just a question of reproducing the same performance as before, but of increasing the quality measures by modifying the various parameters or adding new content to the model. As such, these operations will be carried out in the continuous testing, architecture adjustment and design adjustment steps.

Continuous testing

In this step, the aim will be to see if the results are reliable in a “continuous testing process” by changing some of the model elements. These may be new variables that were not taken into account in the first phase, as the model was already performing well for the objectives of the first phase. Especially, if the collaborators perceive that this additional information can be or are useful for the system. By introducing this new information to the dataset, the prediction system can improve and can become more efficient by giving even more accurate outputs.

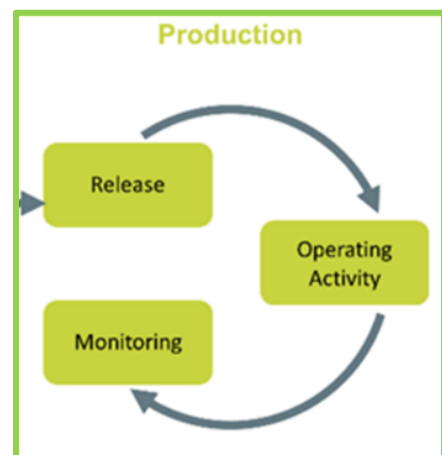
Architecture adjustment & design adjustment

Considering that new elements are added to the prediction system, it is imperative to adjust the architecture and the design of the model, since the previous final model may not be the most “optimal” one anymore. Therefore, it is important to retrain the entire model to ensure that the “best” parameters and hyperparameters are always obtained and applied to the final model after any modification of the model. Moreover, it might be possible that another machine learning method performs better than the one determined in the prototype phase due to additional content. This requires the people responsible for this part to provide explanations for all changes that happened during the previous steps so that other users can understand the changes. In addition, after every modification, the prediction system goes through the realising and continuous testing steps again to see if there is still any potential for improvement. The moment the prediction system reaches the quality and success measures judged sufficient by the team, the MVP can be ready to be transferred to the production phase.

Globally, the MVP life-cycle process enables also to create a first interface or application where the collaborators can have a first try and see what the final product should look like in the future. Nevertheless, it is also the moment to prepare the product to a possible shift to the production in terms of communication and cooperation between the different data science and development teams. Indeed, when the MVP tends to meet the success criteria and the other business objectives, collaboration about the product implementation can already take place so that the targeted customers can benefit as soon as possible from the product.

2.1.3. Production

Once the product has successfully met all the requirements for a deployment within the organisation and received the endorsement of all the collaborators that were involved in the two first phases, it can be sent to “Phase 2: Production”, i.e. the production phase. For the project to be well implemented, it must follow certain specifications, which will be detailed in the following steps.



Release

To ensure a good release of the product to an organisational application, it is essential that the whole process is well documented, that the algorithm and method scripts are well argued and that the first (user) interface is working properly. If these elements are properly completed, it will facilitate communication between the different units so that the principle of transparency in the project transition is respected. Moreover, this will allow on the one hand, a reproduction of the results as obtained in the MVP phase even though to improve them over time and, on the other hand a continuation in the development of a final application inspired from the first (user) interface created.

Operating activity

In the operating activity, both teams (data science and development teams) will have to decide how they want to deploy the project so that it works efficiently with the technologies available within the organisation. In addition, it is possible that the basic model will have to be modified to provide an optimal functionality.

In fact, the deployment of a machine learning project ("MLOps") is not as straightforward as that of a conventional development program ("DevOps") due to additional complexities due to data and methods changes. The data and methods complexities⁴ will require the platform that runs the machine learning model to consider the change in parameters depending on a data change for example. Indeed, the hyperparameters of the model can be different each time a fine-tuning of the hyperparameters occurs in a new model training. Such typical machine learning details are very important to be understood and to be considered for the team that is responsible for the machine learning model deployment, otherwise the project risks being delayed due to a lack of knowledge from the development team. Therefore, once again, the collaboration factor between the units should not be neglected but should be fostered.

Monitoring

The monitoring is the last but one of the most important stage in the data innovation process, as it will ensure the ongoing control of the final product. Indeed, this will allow the people in charge of the product to be alerted when the prediction system encounters a drop in quality. The monitoring can be seen as the stage where the product is sent back into the process to check whether it still complies with the project standards. Therefore, it is important to carefully define a monitoring framework for the three main elements of the project: the data, the methods and the product, which may compromise the effectiveness of the model if they are not regularly and properly reviewed. The number of times a monitoring is necessary in a certain timeframe should depend on the nature of the project.

Firstly for the data, it is essential to check that the input data have not been modified due to changes in the data source. Moreover, further checks should be made to ensure that the data are still of "good" quality ("veracity") and meet the project's requirements.

⁴ <https://www.kdnuggets.com/2020/12/mlops-why-required-what-is.html>

Secondly, it is important to update the model by retraining it with the new data delivered by the data source. This not only ensures an updated model but provides one that can handle new data types that did not exist in the previous dataset. With the new data introduced into the model, the monitoring can detect abnormal data that could represent outliers. These should be analysed by the team to find out why and to explain how the model fails to handle them. Furthermore, the method used should be also monitored to see if it is still performing well or if alternatives or new ones could be more efficient for a potential improvement of the model. Nevertheless, it is also the opportunity to check whether the performance metrics still represent the situation of the project or whether a change is needed in terms of data science goals indicators. If one of the elements is not checked properly, it may compromise the effectiveness of the model.

Thirdly, when talking about monitoring a product, it is about particularly checking whether the product still meets the business expectations and whether it is still of value for the organisation. In fact, if the product no longer satisfies customers or is overtaken by another product, the objectives must be reviewed and updated by the team in charge of the project. However, it is also a time to consider whether new or additional infrastructures as well as more efficient technologies can be allocated to the product to make it more efficient such as in response and computation time.

After having detailed and described in detail the data innovation process that a data science project should ideally follow, it is time to see how it was applied to the “NOGAuto” project.

2.2. Applied to the “NOGAuto” Project

Already in the business understanding step, the data scientists, the business specialists and coders determined the business objectives and success criteria that the “NOGAuto” project must satisfy to move on to the next phases. As already mentioned in the project description in Chapter 1, the main objective of this data innovation project is the creation of a prediction system, which can predict the NOGA codes of enterprises with the information already available within the office using machine learning techniques. As such, for the success criteria, it has been decided to divide these into two groups: one focused on the prediction system and the other one on the domain user. A summary is given in the Figure 3 below (larger version is given in Appendix 1) and further details are given within the next sub-sections.

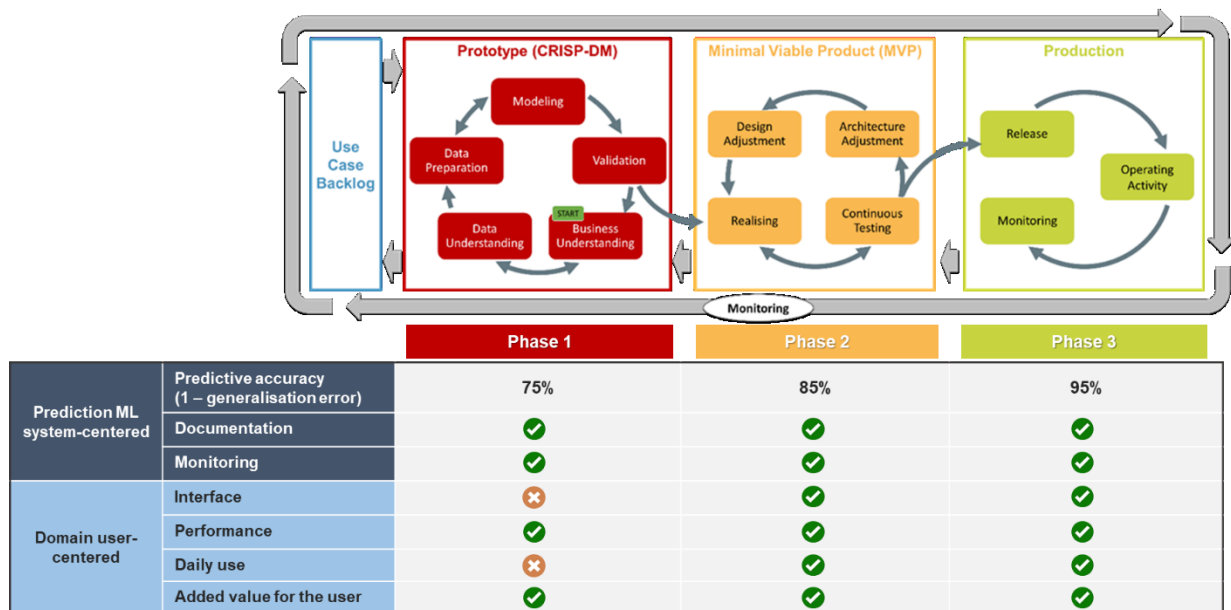


Figure 3: Detailed diagram of the success criteria applied to the “NOGAuto” project

2.2.1. Machine learning prediction system

On the one hand, three success criteria were set for the Machine Learning (ML) prediction system group, namely the performance metric predictive accuracy, the requirement of a documentation and the implementation of a monitoring. These are outlined in what follows.

Predictive accuracy

The first success criterion is about the performance indicator, which is here the predictive accuracy. This one is used to assess the performance of the prediction system and whether the predicted NOGA code matches the real NOGA code. This quality metric is represented as the percentages (%) of correct classification of NOGA codes once the model has been trained. The related metric generalisation error is then simply 100% minus predictive accuracy.

For the prototype phase, it is a question of getting quick feedback on the functioning of the prediction system. Therefore, the first criterion for the system is to reach the predictive accuracy threshold of 75%. Then, if the system shows convincing initial results and reaches 75%, it is time to see if its performance can be improved over time during the second phase. Therefore, the team decided that the system must reach a minimum of 85% of predictive accuracy to complete a first time the MVP phase. In the production phase, the prediction system must hold the minimum threshold of 95% of predictive accuracy and must not decrease over time, otherwise the system must be rectified. The implemented monitoring will enable to do this.

Documentation

The second criterion is the documentation of the three phases. Each phase should contain and be delivered with the following four elements:

- transparency, i.e. all the developments of the prediction system must be explained so that all collaborators can understand what has been done;

- reproducible results, i.e. anyone who wants to reproduce the prediction system with the original code should be able to obtain the same results;
- examples, i.e. to facilitate the understanding of the phases, the data and the methods that have been used, clear examples should be provided. With examples, collaborators will understand more quickly the subject, which will promote better interaction;
- code hygiene, i.e. quite often reading code is not obvious to everyone, so it is essential for the coder to document the code and keep it as clean as possible so that another person can find his or her way around.

Implementation of a monitoring

After having explained and detailed the function of the monitoring in a theoretical way, it will now be presented how this one has been implemented throughout the three phases of the “NOGAuto” project.

In the prototype phase, it will be a matter of imagining and collecting ideas on how the monitoring (of data, methods and product) can be deployed in the final phase. In addition, it will also be necessary to make an analysis of the resources and needs that the monitoring will generate to ensure, for example, a good continuation of the product once it is in use by the users. Then, at the end of the second phase, the ideas conceived in the first phase should be concretised by detailed descriptions and by a concrete implementation plan for the next phase. In the production phase, the monitoring process will be implemented thanks to the efforts made in the first two phases. This allows the data, the methods and the product to be immediately monitored.

2.2.2. Domain user

On the other hand, the team has defined four success criteria centred on the users that the prediction system has to satisfy, i.e. for the domain user group. Nevertheless, before detailing these four, it is important that the users can give feedback so that the system can evolve by considering the expectations of the coding experts and that it suits their needs. For this reason, a questionnaire was created and will be distributed in the second and third phase to the different users who have already used the prediction system several times. The content of the questionnaire can be found in the Appendix 2.

Interface

For the user to get the NOGA code, he or she should be able to use the prediction system within a suitable interface. However, the implementation of this success criterion only takes place in the MVP phase but the architecture of it should have already been designed in the prototype phase. During the MVP stage, the interface should give a good first impression in stability and design. Nevertheless, one element that should not be neglected is the comprehension of the interface, which should be easy to use for the final client. Furthermore, the readability and simplicity of the interface should also be considered, i.e. having all information in one place. During the production phase, the interface will be improved by user feedback. This will enable the deployment of a more robust and improved interface, which is more in line with users' expectations.

Performance

In the prototype phase, the user performance is not really an objective as a platform is not planned to be deployed at this stage of the project. However, the response time should be acceptable when performing different tests during the development of the prototype product. In the second MVP phase, when the user enters the information into the interface, it is imperative that the response time is faster than when they define manually the NOGA code. In the last phase, thanks to the feedback from users and the evolution of the platform, the speed of response should be improved.

Daily use

Since there is no product launch in the first phase, the prediction system must only be operable for users of the two last phases. For the MVP and the production phases, the prediction system must be able to support the frequency and duration of users' use. The interface must be flexible and upgradable, in terms of being easy to modify and to evolve over time. Besides that, during the MVP phase, a helpdesk and a user manual should be set up in order for them to be deployed in the last phase. Thanks to this "client support implementation" the users will be able to ask for help and will not be left alone.

Added value for the user

Finally, the last criterion is very important for the product's lifespan. Indeed, the prediction system must always be of added value for the user, in the sense that the automatic process must be quicker than the manual process and that it must be more accurate than the manual way. The "NOGAuto" system should keep the NOGA coding more homogeneous and should not encounter the problem of the different interpretation of each collaborator. In the first and second phases, it is important to demonstrate to the main users and convince them that the system is more efficient than previous used tools. Then, in the third phase, the application aimed to be adopted within the business group. In addition, this application should be also evolutionary as it might be a source of ideas for other processes that could benefit from this same methodology.

After having documented the problem in detail, having understood the reasons and the importance of implementing the "NOGAuto" prediction system, it was the moment to put all the gathered knowledge into practice. Indeed, all these identifications and descriptions of the problem allowed the team to have a common thread throughout the project and did not leave any room for misunderstanding. All operations are documented on the FSO's internal sharing network, namely Confluence⁵, which enables employees to see at any time the progress or even the changes that have been made to the "NOGAuto" project. Based on the business objectives and the success criteria, it is time to move on to the next exciting stages of the project and dive into the "NOGAuto" machine learning system.

⁵ <https://intranet.confluence.bfs.admin.ch/confluence/display/DI/MVP+NOGAuto>

3. “NOGAuto” Machine Learning System

The “NOGAuto” project can mainly be represented by the workflow in Figure 4 below (larger version is given in Appendix 3), also known as the “NOGAuto” machine learning system, that has been developed by the team at the beginning of the project. The great particularity is that these stages are interconnected, meaning that it is always possible to return to an earlier stage to modify and improve the process to get better outcomes.

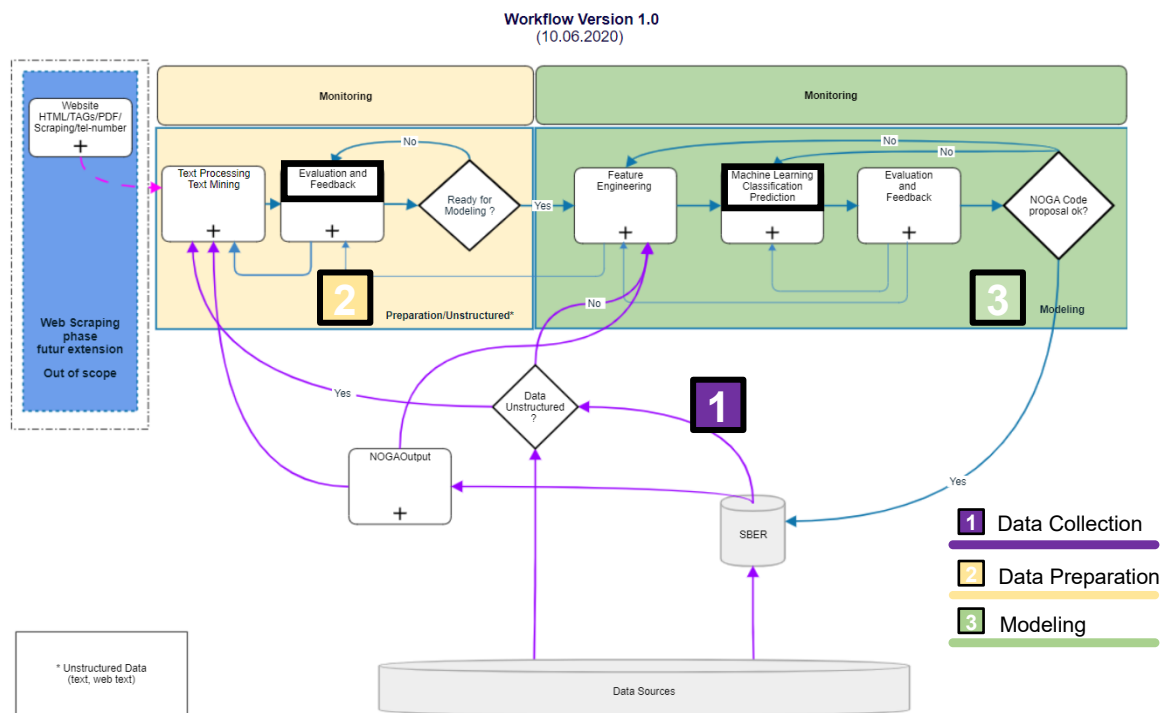


Figure 4: “NOGAuto” machine learning system workflow

The different principal steps are the collection of data (“Data Collection”) followed by the preparation of these data (“Data Preparation”) and their use in the modeling phase (“Modeling”). This workflow is applied to all three phases of the data innovation process, although in a slightly different way. For the prototype phase, all steps are carried out thoroughly, whereas in the MVP and production phases, it will be more a control if there is room for improvement such as adding new variables or gather new information from the data through feature engineering techniques.

The next chapters will detail everything that has been done to have a structured dataset applicable for the machine learning methods.

3.1. Data Collection

For the data collection stage, the data were already available internally in the FSO. In fact, the office deployed in 2018 the third generation of its internal database BER, which is today the Swiss Business and Enterprise Register (SBER) application, where all the company’s information are stored. This database collects the information from many different national register sources such as the commercial registers, value added tax register or customer

transactions register for each company in Switzerland. Therefore, this enabled the team to select various variables that were, according to their expertise, the most determinant in categorising a NOGA code and that could help the future “NOGAuto” prediction system to understand why it should be this NOGA code and not another one.

Below, is the list of the different variables that have been determined by the team. This list was transmitted to the employees in charge of extracting the data from the SBER database to receive a document with all active observations (=enterprises) since 1988.

Local from SBER									
Local_ID	Ent_ID	Name	Additional name	Registers	Universes (0,..., 7)	NOGA code	Admin statut	Stat statut	Type

Local from SBER									
Last nbr of FTJ	Nbr men	Nbr women	Street	Locality	Canton	Language	Location (est coord)	Location (nord coord)	EGID

Legal from SBER					General under the ENT_ID			Single or multi establishment	
Legal form	Registers (0,...,9)	Foundation (DD)	Foundation (MM)	Foundation (YYYY)	Turnover (attached to the ENT_ID)	Export (attached to the ENT_ID)	Import (attached to the ENT_ID)	Admin active local unit	Stat active local unit

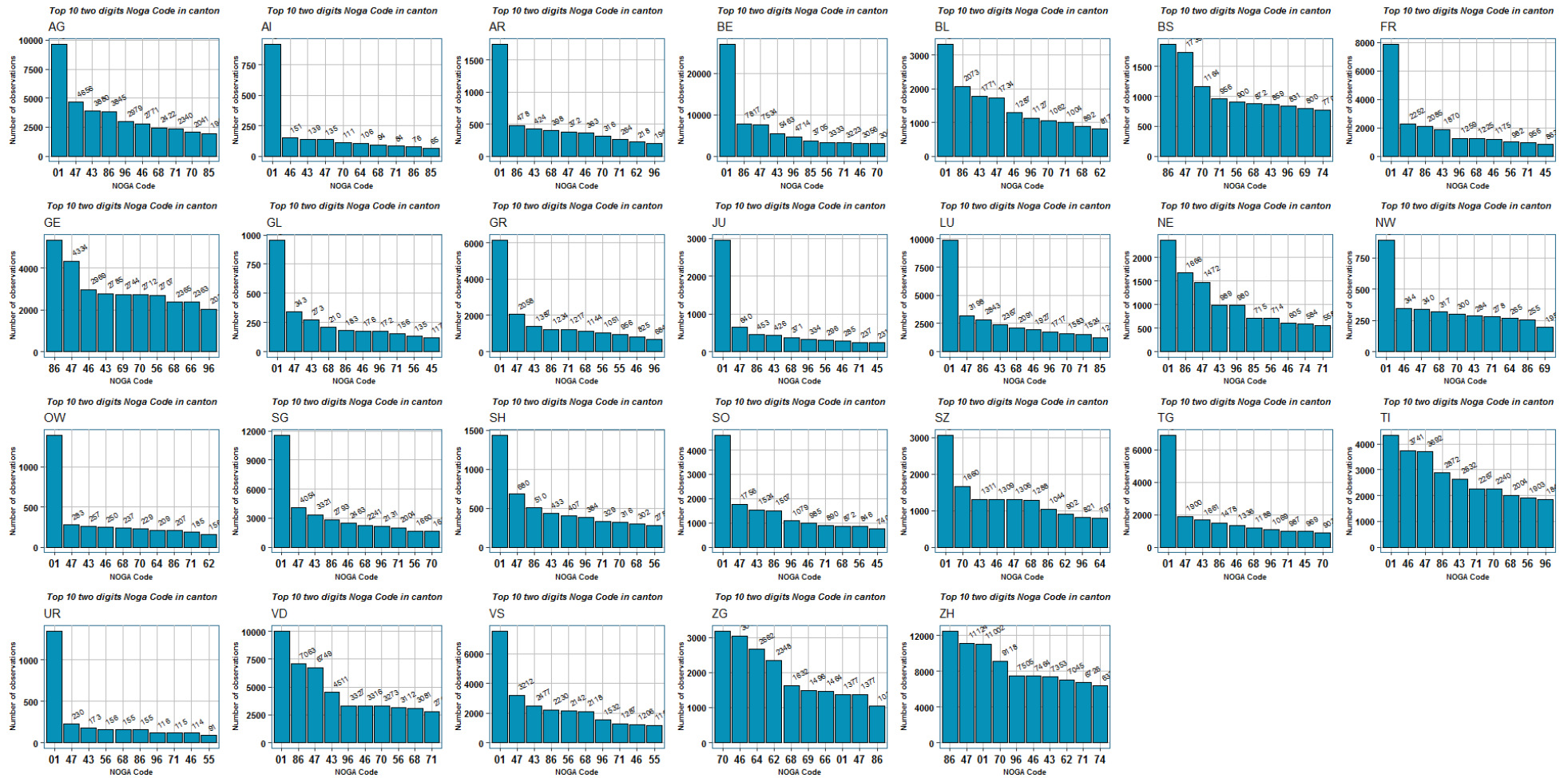
After receiving and uploading the data into the R software, the raw dataset counted in total 1'066'641 observations with 56 variables. The next step was to describe the variables according to their type (number, character or categorical) and what they represent so that the collaborators understand why these variables were selected.

Variable	Nom détaillé	Type	Description
NOGACODE	Code Noga	Character	Code Noga attribué
NOGACODE_TX	Code Noga texte	Character	Description du Code Noga
TYPE	Type	Integer	Catégorie du type d'entreprise
TYPE_TX	Type texte	Character	Description de la catégorie du type d'entreprise

More details can be found on the Confluence web page under the Chapter “MVP Data Understanding”⁶.

After having done this data description, a data exploration was done to get a first impression of the data with the help of graphics and tables. The visualisation, below, is one of many examples of graphs with its interpretation that were used and documented for this step.

⁶ <https://intranet.confluence.bfs.admin.ch/confluence/display/DI/MVP+Data+Understanding>

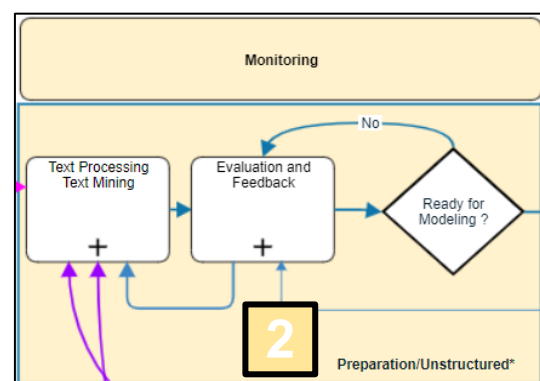


Above, we have a distribution of the two-digit NOGA codes conditioned on the cantons. We notice that for 4 cantons (BS, GE, ZG and ZH) out of the 26 cantons it is not the NOGA code “01” that is the most represented within the canton, but the NOGA code “86” or even “70” (=“activities of head offices; management consultancy”) in the canton of Zug (ZG). This can be simply explained by the fact that these are urbanised cantons or in the case of Zug with tax advantages. Other visualisations, such as the gender distribution by NOGA codes or the number of enterprises that are registered in the commercial register or other ones, were also performed and can be found on the Confluence web page⁷. With such information, it is possible to already have small hints about the results that the system might tend to predict.

Before heading to the data preparation stage, it was important to control the data quality by doing some controls to check if there were some irregularities. In fact, the dataset contained some small anomalies that were then corrected at the beginning of the data preparation section. However, globally, as the data comes from a trustful internal data source, we can guarantee the “veracity” of the data.

3.2. Data Preparation

After having explored the dataset, the data must go through the data preparation step to be consistent for the modeling phase. Here, the goal was to transform the unstructured collected data into structured data. However, not all variables were used in this section, as the NOGA codes are mostly categorised or determined using the activity descriptions of the enterprises. This implies that the data preparation was more focused on text processing, which involves the use of text mining techniques.



In Figure 5 below (larger version is given in Appendix 4), a more detailed view of the process in this section can be seen. The starting point is at the top left with the collection of unstructured data, which then moves on to the language detection followed by the text cleaning and the text mining to end up with structured data.

⁷ <https://intranet.confluence.bfs.admin.ch/confluence/display/DI/MVP+Explore+Data>

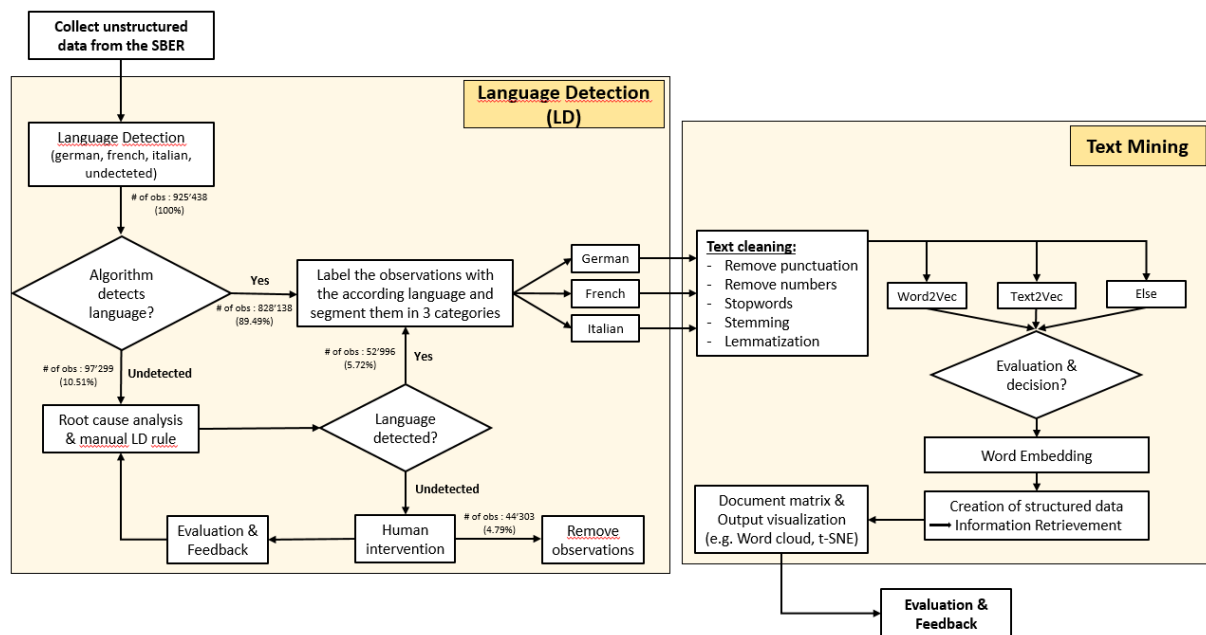


Figure 5: Data preparation workflow

As a first operation, it was necessary to separate the data according to the language of the activity descriptions since there is not a trustful “language” data field in the database, which would have allowed companies to be categorised directly according to their language after collecting the data. Therefore, an algorithm had to be created to detect the language of each observation. This will facilitate the processing of the data for the rest of the project.

After separating the observations according to their language, the texts were cleaned up and the words of the activity descriptions were transformed into categorical variables in a first time and into numerical variables in a second time. Nevertheless, to get a more detailed perception of each step, the following sub-sections will explain which methods were applied for each operation.

3.2.1. Language detection

The purpose of creating a language detection algorithm was to be able to subdivide the dataset into three datasets according to the language of the activity description. However, this algorithm can be considered as an additional product that was developed during the NOGAuto project. Indeed, this product could be deployed within the office to automate the language of other tasks. The language detection algorithm can categorise the observations into three national languages, namely German, French and Italian.

3.2.1.1 R packages

In a first step, to create the first results, linguistic R packages were used, each of them with different methods. These are detailed in the table that can be found in the Appendix 5. In total, there were four available R packages used to detect the languages. The “textcat” package can detect 74 languages, the “cld3” package from Google detects 47, the “cld2” package from Google detects 83 and the “franc” package manages to detect 310 languages.

Each package has an R function, which will detect one language. When one of the packages does not detect one of the three determined languages (German, French or Italian), it will be replaced with "else" in the allocated cell to specify that it is another language and, if it detects nothing (= "NA") it will be "none".

At the beginning, classification conditions were determined so that the activity descriptions of the observations were classified with the detected language. Here is an interpretation example of the first combination listed below: if the "textcat" package ("Textcat") detects the same language as "CLD3" and "Franc" then the observation takes the language found with the "franc" package ("Franc").

Here are the detected combinations only with the four packages:

- "Textcat=CLD3 & Textcat=Franc → Language= Franc";
- "Textcat=CLD2 & Textcat=Franc → Language= Franc";
- "Textcat=CLD3 & Textcat=CLD2 & Textcat ≠ «none» or «else» → Language=Textcat".

Globally, if at least three packages detect the same language, the activity description of the company in question will take the value of this language. Applied to all available observations resulted in the performance summarised in the table below.

	Language detected (in %)	Language undetected (in %)
Language detection algorithm	79.72	20.28

This already enabled almost 80% of the observations to be classified, but to improve the language classification, we used an additional variable, namely the SBER language variable. The additional conditions for classification are as follows:

- "SBER=Textcat & SBER=CLD3 → Language=SBER";
- "SBER=Textcat & SBER=CLD2 → Language=SBER";
- "SBER=Textcat & SBER=Franc → Language=SBER";
- "SBER=CLD3 & SBER=CLD2 → Language=SBER";
- "SBER=CLD3 & SBER=Franc → Language=SBER";
- "SBER=CLD2 & SBER=Franc → Language=SBER".

This has resulted in an increase of about 10% in classifications implying that 828'138 observations were correctly classified.

	Language detected (in %)	Language undetected (in %)
Language detection algorithm + SBER variable	89.49	10.51

To improve the detection of the language and to fine tune its classification, it was decided to use root cause analysis techniques.

3.2.1.2 Root cause analysis

The root cause analysis techniques allow finding relations between the four linguistic R packages and to set up some additional combinations for implementing them in the language detection algorithm. However, before detailing the techniques of association rules, mosaic

plots and correspondence analysis, it has been noted that the system could not recognise certain words.

Recurrent words

In fact, the language detection algorithm did not detect the language of some activity descriptions. In addition, it was noticed that for about 20% of the remaining observations, words that are recognisable to humans are not for the algorithm. Therefore, it was decided to directly allocate the corresponding language to these certain words. These were evaluated and validated by the team. In the table below, we can see the words with the corresponding language:

Word	Language	Word	Language
Médecin	French	Musiker	German
Beratung	German	Produktionsstätte	German
Advokatur	German	Praxis	German
Fotografie	German	Transporte	German
Kosmetik	German	Photographe	French
Medizinalperson	German	Avocate	French
Revier	German	Menuiserie	French
Architekturbüro	German	Grafik	German
Coiffeursalon	German		

It is interesting to see that the algorithm had more difficulties to detect German words. Thanks to this technique, the percentage of correct classification increased by 2.22%.

Association rules

For continuous improvement of the language detection algorithm, the technique of association rules was used to recognise associations in the dataset. Below, we see six rules that were detected by the R package "arules"⁸.

Rules	Textcat	CLD3	CLD2	Franc	SBER	Text	Result	Support	# Cases	Confidence	Lift
R1	-	-	Else	German	German	«Physiotherapie»	German	0.017	1376	1	4.18
R2	German	Else	None	Else	German	-	German	0.0185	1433	1	1.4
R3	Else	Else	None	Italian	Italian	-	Italian	0.0167	1268	0.9906	1.18
R4	Else	-	-	German	German	-	German	0.201	15040	0.961	1.15
R5	German	-	-	-	-	«Verkauf»	German	0.104	622	1	9.43
R6	Else	-	-	Else	German	«Hebamme»	German	0.0101	595	1	5.31

The table can be interpreted as follow:

- For rule 1 (R1): "If CLD2=Else & Franc+SBER=German & Text=Physiotherapie → Language= German";
- For rule 2 (R2): "If Franc+CLD3=Else & CLD2=None & Textcat+SBER=German → Language= German".

⁸ <https://cran.r-project.org/web/packages/arules/arules.pdf>

These rules were selected according to their confidence indicator. Knowing that the maximum value is one, only the rules with values close to or equal to this value were retained. Moreover, the rule 4 has the most impact, as the number of affected cases is 15'040.

	Language detected (in %)	Language undetected (in %)
After 1 st rule	91.71	8.29
After 2 nd rule	91.81	8.19
After 3 rd rule	91.94	8.06
After 4 th rule	93.57	6.43
After 5 th rule	93.64	6.35
After 6 th rule	93.70	6.30

With this technique, the percentage of correct classification rose by 1.99%.

Mosaic plots

The mosaic plots are another visual technique that detects the language of certain observations with combinations of variables. This allowed reducing the observations that had not been yet classified in one of the three language categories. Below, four combinations that were detected by the “vcd”⁹ package with the “mosaic” function are listed.

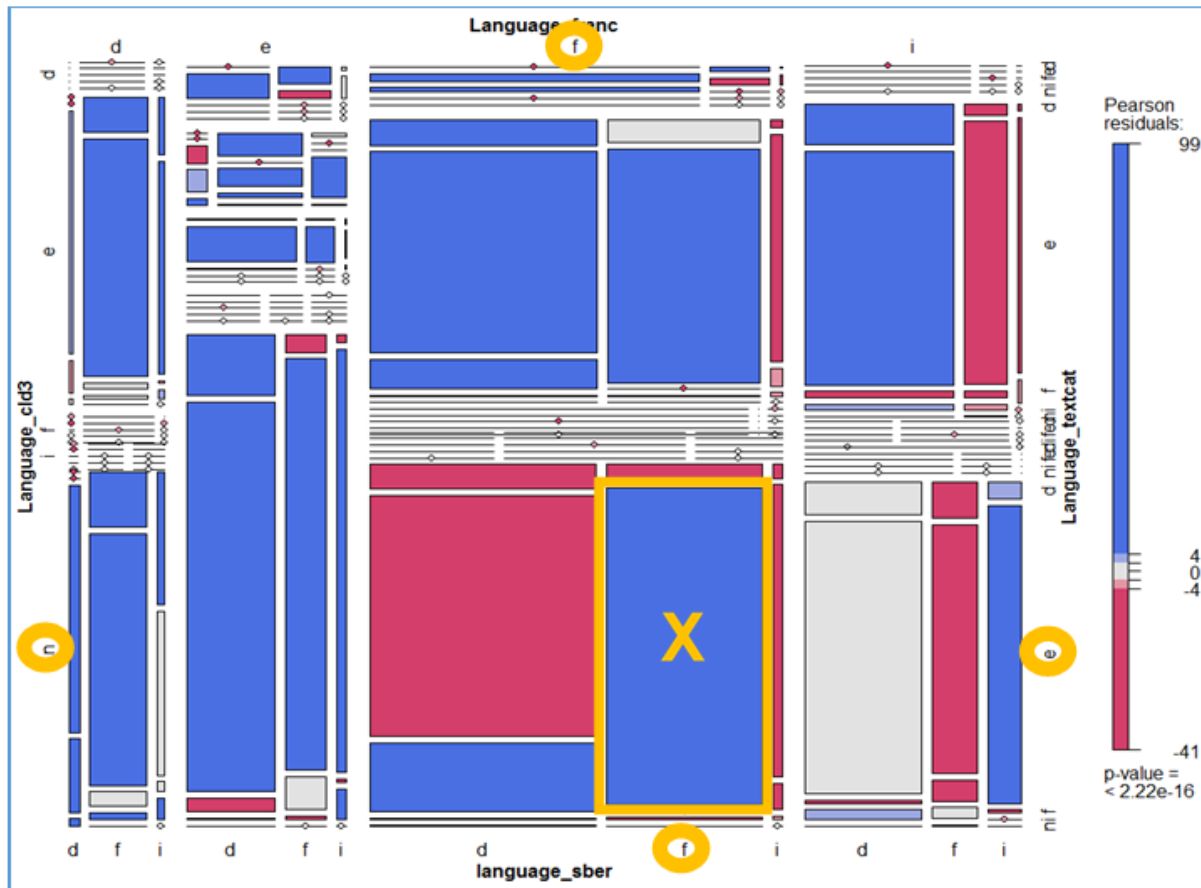
Mosaic	Textcat	CLD3	CLD2	Franc	SBER	Result	Pearson Res	# Cases
Mosaic1	Else	None		French	French	French	99	5'275
Mosaic2		Else	None	French	French	French	91	3'069
Mosaic3		Else	Else	French	French	French	92	1'220
Mosaic4		None	None	Italian	Italian	Italian	93	983

In order to detect the best combination, it is necessary to detect the largest rectangle with the highest Pearson residual (“Pearson Res”)¹⁰ value and, at least two linguistic packages must detect the same language. The combination of Mosaic 1 is an example:

- “Textcat=Else & CLD3=None & Franc+SBER= French → Language = Français”.

⁹ <https://cran.r-project.org/web/packages/vcd/vcd.pdf>

¹⁰ This value shows the strength of the relationship as well as the direction of the associations between the variables, the value represents the absolute value of the “residuals”. The higher this value is, the more strength will be the relation between the combinations.



The three other mosaic plots can be found on the Confluence website¹¹.

	Language detected (in %)	Language undetected (in %)
Mosaic 1	94.27	8.35
Mosaic 2	94.60	8.19
Mosaic 3	94.73	8.06
Mosaic 4	94.84	6.43

Thanks to this visual technique, the percentage of correct classifications has risen by 1.14%, which represents approximately 10'500 enterprises activity descriptions.

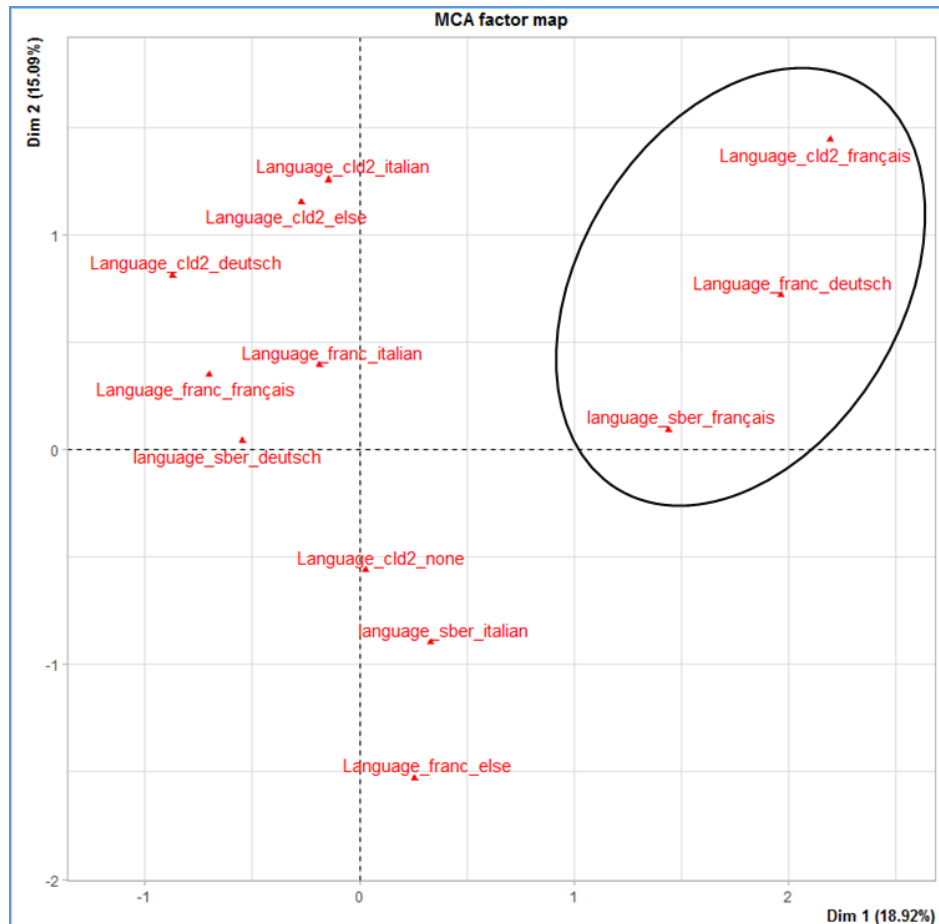
Correspondence analysis

As the last root cause analysis technique, the correspondence analysis will detect trends between variables. Below, we see four combinations that were detected by the "FactoMineR"¹² package with the "mca" function and the graphs. The G2 plot will be used as an example.

¹¹ <https://intranet.confluence.bfs.admin.ch/confluence/display/DI/MVP+Language+Detection>

¹² <http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/113-ca-correspondence-analysis-in-r-essentials>

Plot	Combinations	Result	Total variance	# of Cases
G1	SBER_it & CLD2_it	Italian	39.47%	125
G2	SBER_fr & CLD2_fr & franc_all	French	34.01%	829
G3	SBER_it & CLD3_it	Italian	35.64%	169
G4	Textcat_it & SBER_it	Italian	37.75%	310



The dimensions define the total variance. In the case of plot G2 above, the variation explained by the two first dimension is equal to 34.01% and is the sum of Dim 1 + Dim 2 (=18.92%+15.09%). The higher this variation indicator, the better it is.

For further examples, the other correspondence analysis plots can be found on the Confluence web page¹³.

	Language detected (in %)	Language undetected (in %)
Correspondence analysis 1 (G1)	95.06	4.94
Correspondence analysis 2 (G2)	95.16	4.84
Correspondence analysis 3 (G3)	95.18	4.82
Correspondence analysis 4 (G4)	95.21	4.79

¹³ <https://intranet.confluence.bfs.admin.ch/confluence/display/DI/MVP+Language+Detection>

Thanks to this technique, the percentage of correct classifications increased by 0.37%.

After performing these root cause analyses, it was able to improve and increase the quality of the language detection algorithm by reaching a detection of 95.21% or 881'134 enterprise activity descriptions in the three language categories, namely German, French and Italian.

	Correctly classified	Incorrectly classified
Number of enterprises	881'134	44'303

It should be noticed that each technique has detected one language better than another one. The association rules had better detected German, while the mosaic plots had better detected French and the correspondence analysis had better detected Italian.

Finally, after having separated the observations depending on their language, the three new datasets for each language count respectively:

- German: 625'909 observations;
- French: 208'999 observations;
- Italian: 46'226 observations.

There is still one limitation, the language of 44'303 observations has not been detected. It was decided to see if it is possible to use only the SBER language variable for the latter cases. This would be for:

- German: 31'690 observations;
- French: 10'610 observations;
- Italian: 2'003 observations

However, after performing a sample check on 200 observations for the three datasets, it could be confirmed that this is not a viable solution and that these observations will not be used for the rest of the project. In the table below, it is possible to see the misclassification rate per dataset (interpretation for German dataset: 21 enterprises misclassified and 14 of them were detected as being English activity descriptions):

Dataset	Misclassification (in %)	Misclassification without English (in %)
German	21/200 = 10.50%	7/200 = 3.5%
French	24/200 = 12%	8/200 = 4%
Italian	44/200 = 22%	20/200 = 10%

3.2.2. Text mining & natural language processing

Text mining is one of the most important phases in the “NOGAuto” project. It consists of extracting information from texts and allows the detection of some trends according to the NOGA codes. After having separated the observations into three different datasets, it was easier to process the contents of the activity descriptions. Indeed, all the activity descriptions of each observation served to build a word dictionary for each dataset.

As a starting point, the text of each activity descriptions has been “cleaned” and then the words were transformed into variables. First, they were represented by the number of times they appeared in the activity description by using the one-hot encoding technique and then upgraded to two types of document representation techniques, namely “Word2Vec” and “Text2Vec”. The results were represented with “word clouds” and “t-SNE plots”.

3.2.2.1 Data cleaning

Each activity description went through different stages of cleaning:

- lowering the activity descriptions;
- deleting the “stop words”. A list of neutral words for each language has been created by the team members and can be found on the Confluence web page¹⁴;
- removing the punctuation;
- removing the numbers;
- removing the whitespaces;
- stemming the words, meaning that the words with the same root were grouped together, for example words like “collaborators”, “collaboration” and “collaborate” will be shortened to “collab”;
- lemmatize the verbs, meaning that words from the same verb were transformed to their infinitive form, for instance “eating” and “ate” become “eat”;
- replace special characters, such as «é» with «e» / «ï» with «i», etc.

Below, there are some examples. The first picture shows the activity descriptions that are not cleaned up and in the second one, the cleaned activity descriptions.

```
[1] "Fabrication et commerce de savons et de tous produits moussants." $
[2] "principalement à l'étranger, offre de services et prestations (conseil, formation, étude, audit, expe$
[3] "Exploitation d'une boucherie-charcuterie, d'un magasin d'alimentation et d'abattoirs." $
[4] "Exploitation à l'année" $
[5] "la gestion d'établissements d'enseignement, ainsi que le commerce en gros et au détail de matériel di$
[6] "Vente de logiciels pour la menuiserie métallique" $
[7] "Exploitation d'une boucherie-charcuterie, d'un magasin d'alimentation et d'abattoirs." $
[8] "achat et vente de voitures neuves et d'occasion." $
[9] "tous travaux d'assainissement tels que vidange de fosses, débouchage de tuyaux, curage haute pression$
[10] "l'achat, la vente et l'élevage de chevaux de sport. Elle peut participer à d'autres entreprises, effe$
```



```
[1] "fabriqu commerc savon produit mouss" $
[2] "principal estrang offre servic conseil format etud audit expertis mati econom financi polit patrimonias$
[3] "boucher charcuter magasin aliment abattoir" $
[4] "anne" $
[5] "gestion etabl enseign commerc gros detail materiel didact educ informat ludiqu" $
[6] "vent logiciel menuiser metall" $
[7] "boucher charcuter magasin aliment abattoir" $
[8] "achat vent voitur neuw occas" $
[9] "travail assain vidang foss debouchag tuyal curag haut pression recyclag eau nettoyg rout surfac netto$
[10] "achat vent elevag cheval sport particip entrepris effectu oper immobili" $
```

3.2.2.2 Document representation

After having purified the activity descriptions of the enterprises, the words have been transformed into variables, also known as document representation methods. However, to not take all words into account, a selection of those has been made. Only the words that occurred

¹⁴ <https://intranet.confluence.bfs.admin.ch/confluence/display/DI/MVP+Statistical+Data+Preparation>

more or equal to five times in the dataset were kept. The number five was set to avoid words with spelling mistakes or proper names as variables and can therefore be considered as an additional data preparation step. This already reduced the number of potential variables in the dataset.

One-hot encoding

There are several forms for representing variables in a document such as the one-hot encoding method. This technique consists in representing a word in an integer form. For example, if the word occurs one time in the activity description it will be assigned the number of 1, if the word occurs more than once then the value will be the number of occurrences (ex: 2), otherwise it is 0. The table below shows an example:

[illegible]

Word2Vec

However, there are methods that are more accurate by calculating the similarity between the words and are more precise in categorising them. The more similar the words are, the closer the values will be. This will already allow to categorise the words together and will facilitate the machine learning model to understand the proximity between the words. The “Word2Vec”¹⁵ package, specially designed for Natural Language Processing (NLP), contains two techniques that use the neural network framework:

- **Continuous Bag of Words (CBOW)**¹⁶: a method that takes the context of the word and tries to predict the corresponding word;
- **Skip-Gram**: a method that takes the word in question and tries to predict its context.

Below is an example with the words "développement" and "chantier" from the French dataset using the first method CBOW:

¹⁵ <https://github.com/bnosac/word2vec>

¹⁶ <https://arxiv.org/pdf/1411.2738.pdf>

\$developp					\$chanti				
	term1	term2	similarity	rank		term1	term2	similarity	rank
1	developp	innov	0.8934842	1	1	chanti	traval	0.8824896	1
2	developp	recherch	0.8818625	2	2	chanti	demontag	0.8027724	2
3	developp	develop	0.8727502	3	3	chanti	demolit	0.7961435	3
4	developp	novateur	0.8611716	4	4	chanti	bat	0.7855909	4
5	developp	implement	0.8538113	5	5	chanti	desamiantag	0.7817187	5
6	developp	industrialis	0.8451074	6	6	chanti	adjud	0.7774373	6
7	developp	solut	0.8430938	7	7	chanti	echafaudag	0.7714188	7
8	developp	entrepreneuriat	0.8371705	8	8	chanti	etayag	0.7687325	8
9	developp	nouvel	0.8308249	9	9	chanti	balisag	0.7661577	9
10	developp	acceler	0.8248676	10	10	chanti	suiv	0.7618046	10

And using the method Skip-Gram yields:

\$developp					\$chanti				
	term1	term2	similarity	rank		term1	term2	similarity	rank
1	developp	innov	0.8946796	1	1	chanti	demolit	0.8981807	1
2	developp	develop	0.8824145	2	2	chanti	desamiantag	0.8936342	2
3	developp	nouvel	0.8751815	3	3	chanti	debarr	0.8822270	3
4	developp	technolog	0.8731808	4	4	chanti	bat	0.8775608	4
5	developp	solut	0.8701553	5	5	chanti	deconstruct	0.8726801	5
6	developp	concevoir	0.8560892	6	6	chanti	manutent	0.8613483	6
7	developp	agil	0.8539512	7	7	chanti	fouill	0.8567838	7
8	developp	implement	0.8509357	8	8	chanti	nettoyag	0.8545152	8
9	developp	logiciel	0.8454312	9	9	chanti	ben	0.8538262	9
10	developp	industrialis	0.8453522	10	10	chanti	traval	0.8474767	10

We can see that the similarity coefficient is the main factor that classifies the closest words. The closer the similarity is to the value one, the closer the words are.

Text2Vec

However, since one of the objectives of the project is to predict the NOGA code from the activity description and since they are not only composed of one word but of several words, another method had to be found. This alternative method comes from the "Text2Vec"¹⁷ package, which estimates the proximity of the words in the activity description in question. After running the algorithm, the table below clearly shows that the values of the new variables were not binary but continuous and could also be negative. This will enable the computer to understand the relationships between the words and probably increase the performance of the machine learning algorithm in the subsequent modeling step ("Modeling").

¹⁷ <http://text2vec.org/glove.html>

	acceuill	accroissement	accru	acid	activi	actu	adjuv	aerat	aerograph
1	0.0333002327	0.0120765251	-0.176130236	-0.0250507825	-0.0177054725	0.068750408	-0.048193215	-0.0006116576	0.03332669
2	-0.0858462533	-0.2537395582	-0.369437554	0.1294896680	-0.1692600944	-0.070491407	-0.003190667	-0.3589738886	0.01481694
3	-0.0924382687	0.0410125941	0.113246415	-0.0280771408	-0.3289400178	-0.002946138	-0.089174562	-0.0107090034	0.07060799
4	-0.1623503505	-0.1662164210	-0.068649169	0.0857311003	0.0143362360	0.126382077	-0.132575816	0.0864338864	-0.28878981
5	-0.0175020167	-0.2310284870	-0.250829753	0.1479708488	-0.0387301246	-0.210033472	0.171138465	-0.2903535727	-0.11695624
6	-0.1780628870	0.1813969207	0.090393753	0.1537020751	0.1525069566	0.144546001	0.059588506	-0.1458291823	0.11135098
7	-0.0990955192	0.0368563776	0.060118856	0.0964547510	0.0659683515	-0.033050601	-0.075812617	0.0267228793	-0.06642982
8	-0.0932830725	-0.0564619465	-0.154706335	0.0533642239	0.0256772100	-0.183455670	0.183852247	-0.4633879137	-0.07591548
9	-0.0029590800	0.0001425567	0.157595079	0.0051063578	0.2167194592	0.156321506	-0.233579498	0.1288207203	-0.07137411
10	-0.1110781240	-0.2506399548	-0.090130472	-0.0514540045	-0.0195015091	-0.159799816	-0.204201228	-0.0689856889	-0.14808976
11	-0.0182108910	-0.1503451097	-0.305213735	0.1352359433	-0.2034522953	0.052568645	-0.079318622	-0.1746861663	-0.04687389
12	-0.0137512132	0.0463051152	-0.029658019	0.0039305184	-0.0315909452	-0.149839155	0.118622630	-0.3416887278	-0.07099469
13	0.0452116994	-0.2258474544	-0.282211188	0.1084086555	-0.2232836631	-0.003866861	-0.108658982	-0.1964715447	0.03712575
14	-0.1110781240	-0.0045056953	-0.021298778	0.0709298271	0.2253838450	0.072883669	-0.244993624	0.0264422015	-0.15073701
15	0.1131345175	0.2539796919	-0.242314570	0.2099882417	0.2188035351	0.174974861	-0.022324428	-0.1134739810	-0.12369688
16	-0.0939382634	0.1763759079	0.015803073	-0.0416021495	0.0937655996	0.047951590	-0.012946354	0.1737429569	0.13902648
17	-0.0450343331	0.0133350556	-0.169741112	0.0842154816	-0.0521719644	-0.011185892	-0.138726881	-0.0376736627	-0.38239198
18	0.0829704659	-0.0021816200	-0.131563061	-0.0300061290	0.1340481403	-0.108062515	-0.061634127	0.0010201566	0.07082118
19	0.0240226387	0.0250375783	-0.005471615	0.0469975719	0.0490578003	0.048971188	-0.086115692	-0.0103603173	-0.15167346
20	-0.1765683645	-0.0388026735	-0.362169134	-0.1119114545	-0.1361763904	0.048964606	0.116298754	-0.2358830366	-0.21404580
21	-0.0374362383	-0.0337009629	-0.016470814	0.1545100496	-0.0017679503	0.233849241	-0.181718778	0.1418812545	-0.21770584
22	-0.1192581563	-0.1965929573	-0.056858078	0.0909616375	-0.0561164050	0.060240547	0.065589068	-0.1164835022	-0.04613366
23	-0.1180302845	-0.1267164990	-0.295972659	-0.1278907090	0.0864472592	-0.029849899	0.012448789	-0.2899268974	-0.14104165
24	0.0239409512	-0.0856476372	-0.233686437	0.0002584822	0.0029735163	-0.173528560	-0.048844305	-0.3028957455	0.25193954
25	-0.1084741620	-0.0388913633	-0.194881209	0.2875529726	0.0234326955	-0.188109989	-0.108325252	-0.1671232148	-0.21836944
26	-0.1118399384	-0.1287703918	-0.292592349	0.0171778179	-0.0254486681	-0.056830996	-0.107365295	-0.3174641988	-0.16846567

With the use of this so-called “word embedding” technique, it was possible to expand the set of explanatory variables to a total of 5'068 variables.

3.2.2.3 Evaluation and validation

To check the quality of the data preparation process, the outputs have been described and illustrated with appropriate visualisations on the Confluence web page¹⁸. This gives the different collaborators the opportunity to give their evaluation and approval about what has been done. Depending on the feedback, corrections may be required.

The visualisations are based only on the activity descriptions of the enterprises, since one of the main objectives is to predict the NOGA code from these. Hereafter, the words are visualised in the graphs below using the “t-SNE” package. This one has the specificity of classifying similar words in similar areas thanks to its dimension reduction feature. It is therefore possible to detect clusters of words and check if the word embedding process has worked correctly. In addition to this technique, simple word clouds were also used to see which words are the most represented in the activity descriptions of the dataset.

t-SNE

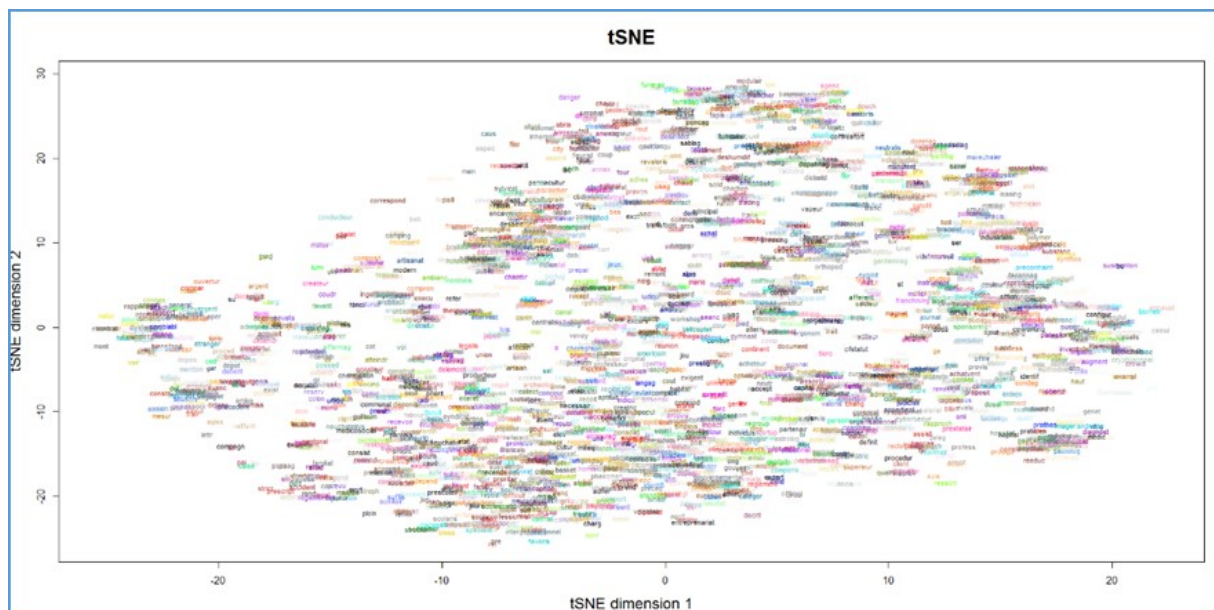
As being one of the most popular way today to show high-dimensional data in a simple graph, the t-distributed Stochastic Neighbour Embedding (t-SNE)¹⁹ aims to group similar data and enables to detect some local trends as well as global ones. These are possible to be visualised in 2D or even in 3D graphs²⁰ with the help of dimension reduction features.

¹⁸ <https://intranet.confluence.bfs.admin.ch/confluence/display/DI/MVP+Feature+Engineering>

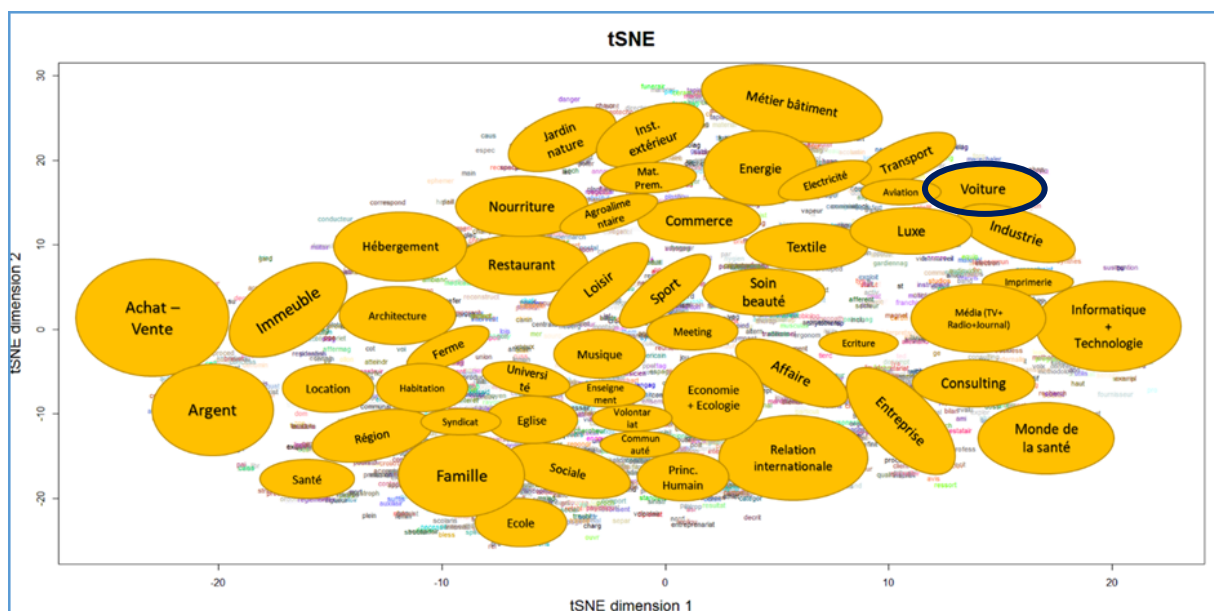
¹⁹ <https://cran.r-project.org/web/packages/tsne/tsne.pdf>

²⁰ <https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf>

To have a good representation of the words in the graphs as well as a correct visualisation of them, it is possible with this package to choose and determine parameters such as the number of reduction dimensions and the perplexity.



However, with the illustration above, it was not obvious to read the words in the graph, so it was useful to zoom in it. This allowed the team to detect similar words around same places in the plot. To get a better overview, it was useful to represent similar words by a representative word as shown below with the help of ellipses.



For example, under the circle “voiture” at the top right, there are words such as “autos”, “carrossier”, “moteur”, “pneumatique”, “garage”, etc. Moreover, not only local trends are visible but also global ones. In the upper part of the graph, it is more about outdoor activities, while in the lower part, it will be more about community life. In the middle, it will be more the entertainment theme, on the left side, housing and investment while on the right side, it will be more communication of information.

These explanations confirm that the use of the word embedding techniques has worked well and that this visualisation technique was useful for the evaluation of the algorithm.

Word cloud

Word cloud graphs can be used to see which words occur most often in the activity descriptions. Below, the graph is the output from the package “wordcloud”²¹ using the French dataset.

In this word cloud, the words that occur the most are in the center and in a larger size. The colours are here only to better distinguish the words. This graph comes from the first one-hot encoding technique. This indicates that the word “vent” is the most frequently occurring element in the activity descriptions before “entrepris”, “activite” and “conseil”.

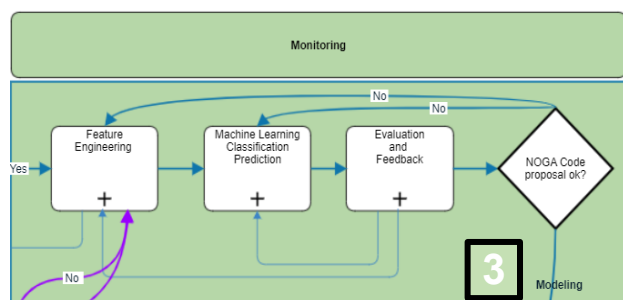


The reason why there is no word cloud for the word embedding techniques is because there is no word counting but rather a measure of word proximity which is not applicable to this visualisation method. Hence, this technique is only useful for finding out which words occur most often in the activity descriptions at the beginning of the data preparation process.

Finally, this chapter ends with the evaluation and validation of the collaborators. Indeed, they were asked to consult the Confluence website²² to see if everything that has been done was understandable and correct according to them. Moreover, it is important to note that a large part of the project's time was spent on understanding and preparing the data. Therefore, it is important to allocate enough time for these parts and not to underestimate them.

3.3. Modeling

The modeling chapter is about training the dataset with machine learning methods to get a predicted value or category for a new observation. The purpose of a machine learning process is to program a computer to learn from information (e.g. data) and to perform a task (e.g. prediction) in an optimal way in terms of a performance measure (e.g. loss function).



²¹ <https://cran.r-project.org/web/packages/wordcloud/wordcloud.pdf>

²² <https://intranet.confluence.bfs.admin.ch/confluence/display/DI/MVP+Data+Preprocessing>

In the case of the “NOGAuto” project, the main objective is to predict the NOGA code according to the activity description of the enterprise. A company is categorised with a 6-digit code at the Swiss national level, where 794 potential classes are possible. However, more global classifications exist, namely 2-digit and 4-digit. With the 4-digit classification level, the Swiss coding structure can be compared to the European one, namely the NACE (=“Nomenclature statistique des activités économiques dans la Communauté européenne”). As a first step, the team decided to predict the NOGA codes at the 4-digit level instead of the 6-digit level. The reason of this decision was to see if the model was already able to predict the 4-digit categories. The number of potential classes at this level is 615.

For this project, the method that has been chosen to train the models is the Gradient Boosting Machine (GBM), i.e. stochastic gradient tree boosting. This because of the lessons learnt from previous pilot projects that the FSO launched (in particular the project “Plausi++”²³ where the R package GBM²⁴ has been successfully used) and that this method is quite robust and efficient in multi-classification. This method has proven its competence in many areas and is one of the preferred methods that is used in Kaggle competitions²⁵. The error function that will be used with this method is the misclassification error meaning that it counts the number of times the computer misclassifies the data.

The GBM derives from the tree-based methods that, in general terms, segment the feature space (e.g. variables) recursively into simple regions. The main idea of the GBM method is to grow trees using information from the previous one to reduce the error term. This error term is reduced with the help of the gradient descent method that aims to find a local minimum where the loss function is optimized. The specification of the GBM method is that the first trained tree considers all the observations equally and measures the misclassification error. Then, before growing the next tree, the method gives more weight to those that were difficult to classify and afterwards train a new tree. This process will continue until the algorithm has reached the number of defined iterations.

In the remaining parts of this section details will be given regarding the training and prediction parts. It will be discussed how the data were treated, which R packages were used, how the first predictions were made and how they were presented to the collaborators.

3.3.1. Training

Unfortunately, before enumerating the modelling steps, it is important to mention that the IT infrastructure on which all the work was conducted did not have the computational capacity to support the computational demands. Therefore, there have been many program crashes, which have made the task of training the models more difficult. However, model training and prediction were possible for a small sample of the dataset, which is at least encouraging for the future of the project. Once the IT infrastructure would be good enough to train the models,

²³ <https://www.bfs.admin.ch/bfs/de/home.assetdetail.9847917.html>

²⁴ <https://github.com/gbm-developers/gbm>

²⁵ http://uc-r.github.io/gbm_regression

it will only be necessary to use the code that has already been prepared. Indeed, it has been possible to set up the scripts that will make the predictions.

caret

The “caret” R package²⁶ is one of the best-known and most widely used packages for training models in the data science world. It allows finding the “best” tuning parameters according to the method used for training machine learning models through external validation methods such as cross-validation, repetitions, etc. With these different operations, it is possible to find the “optimal” combination of hyperparameters that minimises the loss function.

In the case of the “NOGAuto” project and with the GBM ML method available in the caret package, the tuning parameters²⁷ that have been defined by the data scientist are:

- the “interaction depth”: the complexity of the tree, meaning the higher the value, the more complex the model will be;
- the “number of trees”: the number of time the algorithm grows trees;
- the “shrinkage” parameter: the learning rate of the algorithm, meaning the lower the value, the more complex the model will be;
- the number of “minobsinnode”: the minimum of observations in a node that enables the tree algorithm to be splitted.

An important parameter that can also be determined is the external validation method. In the case of NOGAuto, the methods that were used are the “repeatedcv” and the “cv”. This detail allows the algorithm to train the data together. However, the more repetitions and the more complex parameters the algorithm contains, the longer the system will take to calculate the model. Therefore, it is important to find the right balance.

Hyperparameters determination

After having trained several models with different numbers of observations and predictors (e.g. variables), the “best” model with the optimal hyperparameters gave a predictive accuracy of 52.93% with the “caret” package. This model was trained on 1’001 observations and 1’500 random variables from the activity descriptions only. In addition, the hyperparameters were: “interaction depth=5”, “shrinkage=0.01”, “number of trees=1’000” and the number of “minobsinnonde=10”. All this calculation took over 25 hours. Additional model runs can be found in Appendix 6.

GBM

Now that the algorithm has found the “optimal” parameters, these can be introduced in the GBM training function using the R package GBM²⁸. There, the algorithm trains a model that has been applied to the same small set of observations as well as the same 1’500 random variables.

²⁶ <https://topepo.github.io/caret/index.html>

²⁷ <https://topepo.github.io/caret/model-training-and-tuning.html#tune>

²⁸ <https://github.com/gbm-developers/gbm>

3.3.2. Prediction

Below are the first predictions that have been predicted with the NOGAuto model modeled with the GBM method. In addition, it was decided that the three most probable NOGA codes should be shown with their “percentage probability” and not only the most probable one. It should be remembered that the model was trained on a small part of the dataset and that the results would be different if the model would have been trained on the whole dataset. However, the results below seem to be very convincing. Indeed, thanks to the black boxes in Figure 6 below, it is possible to see that on this sample of 35 cases, 22 were directly predicted with the right NOGA codes, which represent a predictive accuracy of 62.85% in this sample. Moreover, if the second and third proposals are considered, the number of correctly classified cases is 27, representing a predictive accuracy of 77.14% in this sample.

	Real NOGA code	Prediction	Match (=1) vs No Match(=0)	1st proposition	2nd proposition	3rd proposition
47	N_0147	N_4778	0	N_4778 à 32.547	N_4669 à 5.18	N_5814 à 3.92
48	N_4120	N_4120	1	N_4120 à 100	N_1623 à 0	N_8121 à 0
49	N_6202	N_6202	1	N_6202 à 76.761	N_2651 à 9.678	N_4321 à 3.101
50	N_7111	N_7111	1	N_7111 à 99.881	N_1812 à 0.028	N_4778 à 0.015
51	N_7219	N_7219	1	N_7219 à 91.171	N_7490 à 2.784	N_4671 à 0.936
52	N_6832	N_6832	1	N_6832 à 95.679	N_6820 à 4.319	N_6831 à 0
53	N_4334	N_4334	1	N_4334 à 99.996	N_4939 à 0.001	N_9319 à 0.001
54	N_1413	N_4778	0	N_4778 à 42.109	N_4764 à 9.688	N_1413 à 5.492
55	N_4942	N_8121	0	N_8121 à 99.973	N_9900 à 0.006	N_4941 à 0.005
56	N_4674	N_4778	0	N_4778 à 42.938	N_4776 à 32.891	N_4634 à 6.467
57	N_4520	N_4520	1	N_4520 à 99.945	N_4941 à 0.018	N_4511 à 0.014
58	N_8559	N_8559	1	N_8559 à 56.843	N_7021 à 19.955	N_8552 à 7.548
59	N_6831	N_6831	1	N_6831 à 100	N_6810 à 0	N_4110 à 0
60	N_8899	N_8899	1	N_8899 à 70.424	N_8790 à 21.451	N_4939 à 1.252
61	N_5610	N_5610	1	N_5610 à 100	N_4932 à 0	N_4722 à 0
62	N_7022	N_7022	1	N_7022 à 100	N_6202 à 0	N_7021 à 0
63	N_4399	N_6820	0	N_6820 à 14.444	N_4399 à 11.798	N_4321 à 9.562
64	N_6499	N_6499	1	N_6499 à 99.977	N_9601 à 0.007	N_7010 à 0.006
65	N_8690	N_8690	1	N_8690 à 99.992	N_4776 à 0.008	N_4762 à 0
66	N_4771	N_4771	1	N_4771 à 18.945	N_4690 à 11.958	N_4759 à 7.829
67	N_6499	N_6499	1	N_6499 à 99.997	N_6832 à 0	N_4110 à 0
68	N_4334	N_4334	1	N_4334 à 100	N_1623 à 0	N_4329 à 0
69	N_5811	N_7410	0	N_7410 à 24.904	N_4762 à 24.28	N_5811 à 13.721
70	N_6810	N_6810	1	N_6810 à 99.998	N_6820 à 0.001	N_4614 à 0
71	N_4110	N_6820	0	N_6820 à 51.343	N_6832 à 14.57	N_6612 à 6.476
72	N_4643	N_5911	0	N_5911 à 42.549	N_1051 à 7.999	N_6209 à 5.259
73	N_6619	N_6622	0	N_6622 à 81.047	N_6630 à 17.775	N_6619 à 1.145
74	N_7740	N_6420	0	N_6420 à 35.799	N_6619 à 21.732	N_6622 à 3.812
75	N_6619	N_6630	0	N_6630 à 63.738	N_6619 à 14.737	N_7021 à 3.47
76	N_6810	N_6810	1	N_6810 à 99.998	N_4661 à 0.001	N_6820 à 0.001
77	N_4791	N_4776	0	N_4776 à 39.979	N_1105 à 28.053	N_1071 à 5.873
78	N_8710	N_8790	0	N_8790 à 45.087	N_8891 à 25.165	N_8610 à 11.678
79	N_6499	N_6499	1	N_6499 à 99.861	N_6420 à 0.139	N_6612 à 0
80	N_7010	N_4639	0	N_4639 à 52.041	N_5621 à 14.488	N_4690 à 6.172
81	N_4941	N_4941	1	N_4941 à 85.712	N_4939 à 7.016	N_8230 à 4.884
82	N_8690	N_8690	1	N_8690 à 100	N_6419 à 0	N_8610 à 0

Figure 6: First NOGA code predictions at the 4-digit level

These promising results show the potential of the “NOGAuto” project, in the sense that if the system had more computing capacity, the trained model would become very powerful and would make more precise predictions. However, with the transition of the project from the initial IT infrastructure (“Atlantica”) to the new collaborative data science platform “RENKU”²⁹, this

²⁹ <https://health.renkulab.datascience.ch/>

need for computational power may be assured and may allow the project to achieve its overarching objectives.

As the internship period has passed in a blink of an eye, it is already time to close this master thesis with the chapter lessons learnt as conclusion of this internship at the FSO.

4. Lessons Learnt

In this last chapter, I will discuss some points that could be improved, what I have learnt and experienced in the organisation, explain on how this project could be implemented at the FSO and how the collaborators would or could use it in the future.

First, there are some parts I would have done it differently to be faster in time and to meet more objectives for the FSO. By now, if the IT infrastructures had been efficient enough, an interface could have been set up for the collaborators. Second, as we have seen throughout this report, we have focused more on the activity descriptions of enterprises. This means that it would be possible to base the prediction of the NOGA code on these alone. Therefore, if the models, which will be trained in the future based only on the activity descriptions, show satisfactory results, it may not be necessary to collect all the variables of the enterprises but only those that would have an impact on the NOGA code. To find out which variables will be most interesting, techniques such as “univariate filtering” for feature selection could be used. Third, due to the limited access rights of administrator and the non-access to internet on the used IT infrastructure (“Atlantica”), some techniques could not be used. Indeed, this might have allowed the use of more resources such as additional linguistic packages, that require internet access, to improve the language detection algorithm. Because unfortunately 4.79% of the activity descriptions (=44’303 enterprises) were not detected and were lost for the rest of the project.

After having described the points that could be improved for this project, I will talk about what the project brought me in ten months. This internship allowed me to put into practice my theoretical knowledge that I acquired during my first year at the University of Geneva. However, the implementation of such a data science project can not be compared to the practical projects carried out during the diverse university lectures. Indeed, here it was a question of interaction between different professions to find a new solution or in this case, a new service suitable for the final customers, which would improve one of their manual tasks with new methods. This was done through an in-depth study of the situation, explaining in detail the problem, how it could be solved and with the help of which instruments. Therefore, it is not enough to collect data and apply data science methods to it, but to make a real analysis of the environment and create a cycle of continuous improvement. It was around all these analyses that I realised how important the initial business understanding step from the “Prototype (CRISP-DM)” phase was. This is where the whole project is conceived, planned, realised, implemented and developed in a rigorous way. Through a well-defined process, problems that the project may encounter can be better anticipated and managed by the project team.

Thanks to a meticulous description of the three-phase data innovation process of the “NOGAuto” project and its success criteria, the project was selected to be presented at the International Conference on New Techniques and Technologies for Statistics 2021 (NTTS

2021) by the organisers. This conference brings together all the countries that register to talk about new techniques and technologies that are used and implemented, either by public institutions, public or private companies. This conference was planned to take place physically from March 9 to 11, 2021, in Brussels. However, due to the COVID-19 pandemic situation and the health restrictions that the various governments have put in place, physical representations were not possible. Therefore, all presentations were broadcasted by video conferences, which nevertheless, allowed gathering more interested people. The “NOGAuto” presentation took then place on March 10, 2021 and was recorded³⁰. By opening the presentation to a wider audience, a few days later, the United Nations Economic Commission for Europe (UNECE) invited the “NOGAuto” project to a group seminar on standardising the development of a machine learning project within the HLG-MOS Machine Learning Project³¹. It was discussed how a MVP could be transferred to production and with which procedures.

It is important to underline that the effort invested in the realisation of the “NOGAuto” project is not only limited to the classification of the economic activities of enterprises. In fact, a slight adaptation of the system would allow a wider use of this technology to assign codes to, for example, professions, diseases, causes of death, products for more precise statistics within the FSO. With a small change in the algorithm script, this would only involve collecting the corresponding data from the new classification, adapting the text to be processed, indicating the new dependent variable and running the algorithm. Hence, these potential new applications of the prediction system will still have to be treated as a new project and face a transition into the data innovation process. However, thanks to the experience gained from the “NOGAuto” project, this will be more straightforward.

Another potential application that could be interesting is not only to apply the “NOGAuto” prediction system to the Swiss perimeter but also to extend it to the European level with the NACE codes. Indeed, since the “NOGAuto” project was programmed on the 4-digit NOGA codes and the European one follows the same structure, this could lead to interesting inertias. In fact, business statistics could be more accurate, as they would be less affected by the interpretation biases of the collaborators but especially the differences from country to country. This would increase the quality of comparison between countries, allowing a more consistent view of the situation between countries.

In addition to the potential that code classification may have in the future, the improvement of the language detection algorithm created upstream may also represent an opportunity for the FSO to develop its own language detection system. This will require further development and improvement of the product, which today still classifies more than 95% of observations in the correct language.

It was also a very interesting experience in the sense that I had the opportunity to work on site for the first 3 months and 7 months in home office due to the current situation and decisions that have been taken by the government to tackle the COVID-19 pandemic. However, this has

³⁰https://coms.events/NTTS2021/data/abstracts/en/abstract_0041.html

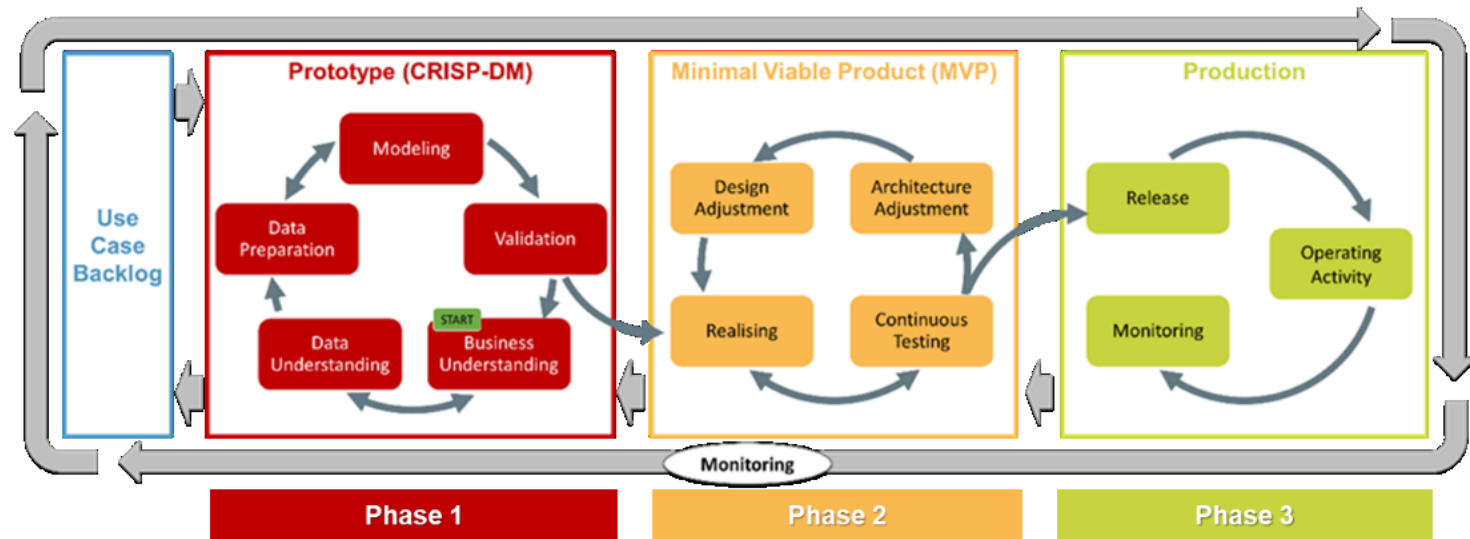
³¹<https://statswiki.unece.org/display/ML/HLG-MOS+Machine+Learning+Project>

allowed me to strengthen my work discipline and to develop my autonomy even more. Nevertheless, it is true that on-site meetings between collaborators are still irreplaceable.

In conclusion of this very interesting work, I would like to thank Professor Kuonen, Mrs. Duc Sfez and all the collaborators for all the advices and help they gave me during my ten month of internship at the FSO .

5. Appendix

Appendix 1: Detailed diagram of the success criteria applied to the “NOGAuto” project



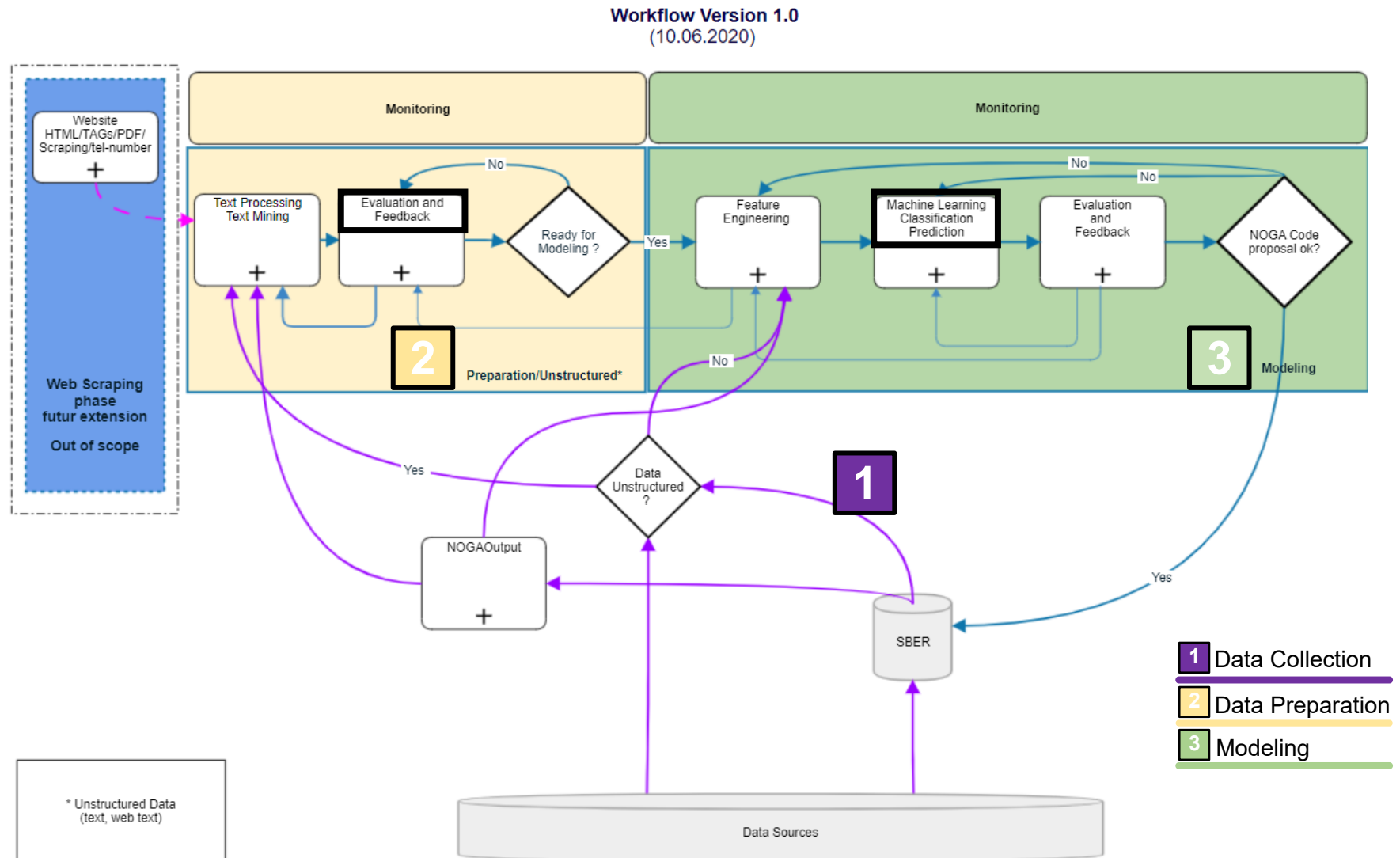
		Phase 1	Phase 2	Phase 3
Prediction ML system-centered	Predictive accuracy (1 – generalisation error)	75%	85%	95%
	Documentation	✓	✓	✓
	Monitoring	✓	✓	✓
Domain user-centered	Interface	✗	✓	✓
	Performance	✓	✓	✓
	Daily use	✗	✓	✓
	Added value for the user	✓	✓	✓

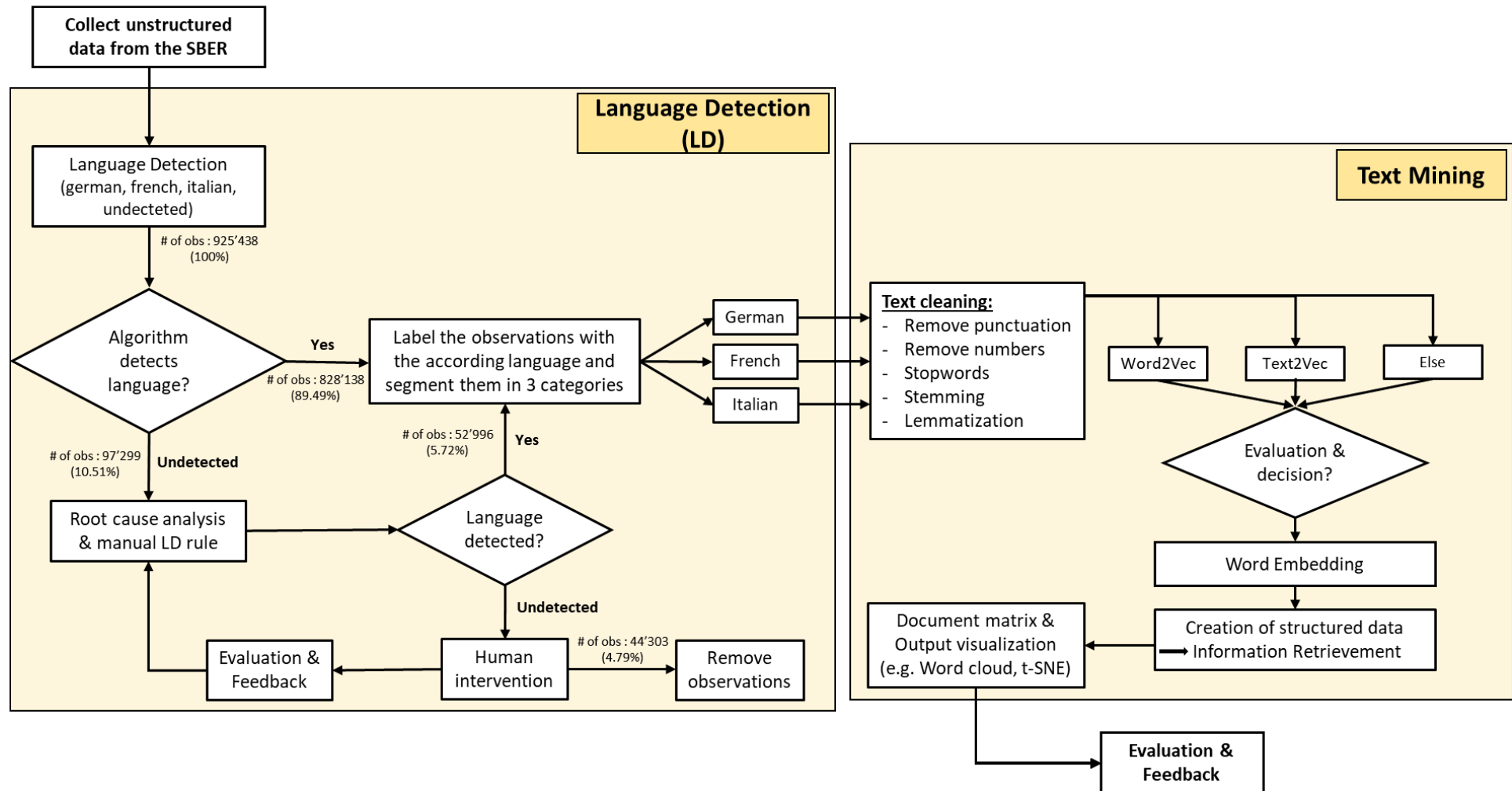
Appendix 2: «Fiche d'évaluation du prototype des collaborateurs»

The questionnaire below allows collaborators to give their evaluation and feedback.

Nom du logiciel	NOGAuto				
Date d'évaluation					
Nom évaluateur					

Environnement test					
Points généraux					
Nombre de cas traités					
Langue utilisée	☐ Allemand ☐ Français ☐ Italien ☐ Anglais				
Qualité de la prédiction					
Est-ce que la prédiction est satisfaisante ?	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Indiquez la raison de votre insatisfaction concernant la qualité de la prédiction <i>(uniquement si 1 ou 2) (facultatif)</i>					
Plus-value de la prédiction					
Est-ce que la prédiction fournie est une plus-value ?	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Expliquez votre note <i>(facultatif)</i>					
Layout					
Masque de saisie	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Performance					
Rapidité de l'affichage des résultats	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Estimez-vous qu'il y a des améliorations à apporter au prototype ? <i>(Veuillez indiquer le(-s) point(-s) concerné(-s))</i>					

Appendix 3: “NOGAuto” machine learning system workflow

Appendix 4: Data preparation workflow

Appendix 5: Details on the R packages used for language detection

Packages	Created by	Method used	Number of languages
Package «textcat» Version: 1.0-6 Year: 2018	Kurt Hornik, Johannes Rauch, Christian Buchta, Ingo Feinerer	<i>“This package uses two steps. In a first time the package collects, training corpus of texts all written in the same known language and from there builds language profiles. The algorithm classifies from the most to the least frequent n-grams and only keeps the most frequent ones. In a second time, from the unknown language of a given text, it computes again a document profile and classifies it according to the language profile performed in the first step. It minimises the “out-of-place”¹ distance between the n-gram profiles of both texts and categorizes as the most similar.</i> References, CRAN: https://cran.r-project.org/web/packages/textcat/textcat.pdf References, Researchgate: https://www.researchgate.net/publication/282255965_The_textcat_Package_for_n-Gram_Based_Text_Categorization_in_R ¹ https://github.com/halfvim/N-Gram/wiki/Out-of-place-Distance-Measure	74 languages
Package «cld3» Version: 1.3 Year: 2020	Jeroen Ooms, Google Inc	<u>Google's Compact Language Detector 3:</u> <i>“CLD3 is a neural network model for language identification. The model averages the embeddings corresponding to each ngram type according to the fractions, and the averaged embeddings are concatenated to produce the embedding layer. The remaining components of the network are a hidden (Rectified linear) layer and a softmax layer. To get a language prediction for the input text, the model performs a forward pass through the network.”</i> References, Github: https://github.com/google/cld3#readme	47 languages
Package «cld2» Version: 1.2 Year: 2018	Jeroen Ooms, Dirk Sites	<u>Google's Compact Language Detector 2:</u> <i>« For mixed-language input, CLD2 returns the top three languages found and their approximate percentages of the total text bytes using the Naïve Bayesian classifier.</i> References, Github: https://github.com/cld2owners/cld2#readme	83 languages
Package «franc» Version: 1.1.2 Year: 2019	Gabor Csardi, Titus Wormer, Maciej Ceglowski, Jacob R. Rideout, Kent S. Johnson	<i>Possibility to reduce the scope to 3 languages: German / French / Italian</i> References, Github: https://github.com/gaborcsardi/franc#readme	310 languages

Appendix 6: Additional trained models with corresponding hyperparameter settings

Machine learning prediction with the GBM method (“caret” package) and with <u>5 folds and 3 repeats for the external validation</u>							
# Observations	# Predictors	Interaction. depth	Shrinkage	Tree (<i>best accuracy</i>)	N.minobsinnode	Time	Best accuracy
101 (69 classes)	100	9	0.05	80	3	-	0.1644
101 (69 classes)	100	9	0.05	150	3	-	0.1693
101 (69 classes)	100	9	0.05	310	5	~5min	0.1605
101 (69 classes)	100	9	0.01	320	3	~6min	0.1707
101 (69 classes)	100	5	0.01	300	3	~3.5min	0.1704
101 (69 classes)	100	5	0.01	160	10	~2.33min	0.1695
101 (69 classes)	100	9	0.1	300	10	~6min	0.1433
<u>1'001</u> (216 classes)	100	5	0.01	400	10	~1.86 <u>h</u>	0.3297
1'001 (216 classes)	100	5	0.01	600	10	~3.68 h	0.3327
1'001(216 classes)	200	5	0.01	640	10	~7h	0.3519
1'001(216 classes)	400	5	0.01	500	10	~16.8h	0.3624
1'001(216 classes)	500	9	0.01	900	10	~28h	0.3552
1'001(216 classes)	500	5	0.01	500	10	~28h	0.3576
501 (161 classes)	1'000	5	0.01	900	10	~4.38h	0.3639
1'001(216 classes)	1'500	5	0.01	1000	10	~25.25h	0.5293

Machine learning prediction with the GBM method (“caret” package) <u>and with 2-fold CV for external validation</u>							
# Observations	# Predictors	Interaction. Depth	Shrinkage	Tree (<i>best accuracy</i>)	N.minobsinnode	Time	Best accuracy
1'000 (216 classes)	3'000	5	0.01	800	10	~22h	0.4026
5'000 (365 classes)	5'000	5	0.01	No result	10	Stopped for too long	No result