**Software requirements specification**

Automatically supplement missing data



|  |
| --- |
| **Date : 17-03-2016** |
| **Version : 0.5** |
| **Status : Concept** |
| **Document name : Software requirements specification v0\_5** |
| **Author : Tim Hermens** |

**Version**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **Status** |
| 0.1 | 25-02-2016 | Tim Hermens | First concept. | Concept |
| 0.4 | 16-03-2016 | Tim Hermens | First concept. | Concept |
| 0.5 | 17-03-2016 | Tim Hermens | Processed remarks of Paul Keuren. | Concept |

**Distribution**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Recipient** |
| 0.1 | 10-03-2016 | Paul Keuren |
| 0.4 | 17-03-2016 | Paul Keuren |

**Approval**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date of approval** | **Name** | **Signature** |
|  |  |  |  |

**Glossary**

**KSC**

Knowledge, Competence, Skills. This is the data in a profile providing information about the person's competence and skills.

**H**

This can refer to the high indication of the level of urgency or importance of a requirement.

**M**

This can refer to the medium indication of the level of urgency or importance of a requirement.

**L**

This can refer to the low indication of the level of urgency or importance of a requirement.

**MoSCoW**

The MoSCoW method is a prioritization technique to reach a common understanding with stakeholders on the importance on delivery of each requirement. The capitalized letters stand for *Must have*, *Should have*, *Could have*, and *Would have*. Requirements labelled as *Must have* are critical for the product. Requirements labelled as *Should have* are important, but not as time-critical as *Must have* requirements. Requirements labelled as *Could have* are desirable, but not necessary for the product. Requirements labelled as *Would have* are not planned to be fully implemented for the product, but the product should support the possibility for them to be implemented in the future.

**Product B1**

Product B1 is one of the two products of the system. This product is the server-side algorithm functionality in the form of a Python library, that can predict missing skills in a profile.

**Product B2**

Product B2 is one of the two products of the system. This product is the stand-alone algorithm analysis tool which will test the quality (of sub-solutions) of product B1.

**Big Data**

Big Data is a term for datasets that are very large and/or complex. This data is difficult to process using traditional database and software techniques. **Table of contents**

1. Introduction 5

1.1 Purpose 5

1.2 Goal of this document 5

1.3 Intended audience and reading suggestions 6

1.4 Project scope 6

1.5 References 6

2. Overall description 7

2.1 Product perspective 7

2.2 Product functions 8

2.3 User classes and characteristics 8

2.4 Operating environment 8

2.5 Design and implementation constraints 8

2.6 User documentation 9

2.7 Assumptions and dependencies 9

3. Requirements regarding the server-side algorithm functionality 10

3.1 Overall product description 10

3.2 Features and rules 10

4. Requirements regarding the stand-alone application 16

4.1 Overall product description 16

4.2 Features and rules 16

5. Non-functional requirements 35

5.1 Performance requirements 35

5.2 Safety requirements 35

5.3 Security requirements 35

5.4 Software quality attributes 35

5.5 Business rules 35

6. User interface wireframes 36

# Introduction

## Purpose

The company 8vance Matching Technologies BV uses scrape techniques to collect a large amount of data in a short time span. The problem is that this data often misses interesting information - a complete list of skills of persons in particular (this missing data is called [KSC](#KSC)). The [KSC](#KSC) data gives an overview what a person is capable of.

The company wants to have a product that can solve this problem (this product will be called [product B1](#ProductB1) from now on). [Product B1](#ProductB1) should be able to automatically add the missing [KSC](#KSC) data for all the profiles and must be integrated in the company's server. [Product B1](#ProductB1) is basically a piece of algorithm functionality for the server that will add data to the profiles, meaning it won't have any interaction with users. [Product B1](#ProductB1) will be important for the following stakeholders:

* Data scientists. They'll want to use the product to supplement the missing skills in profiles to create a more complete overview of the capabilities of a person. This will very likely improve 8vance's matching algorithm that's used to match persons with jobs.
* Natural language processing experts. They can use the certainty scores along with the predicted skills to potentially create a new taxation or improve an existing one.

The company also wants to have a product that's able to test the quality (of sub-solutions) of [product B1](#ProductB1). The product is a stand-alone tool that can be used to analyze algorithms on correct input and output, and performance (this product will be called [product B2](#ProductB2) from now on). There're two users for [product B2](#ProductB2):

* Developers of [product B1](#ProductB1). They'll want to use the product to test the (sub-solution) algorithm(s) on correct input and output, and performance. If the analysis overall negative, it's a sign the algorithm needs to be improved.
* Other users. The previous user is the main user of the product. There're also other users who don't want to use the product with a problem-solution mindset, but who're just interested what predictions the algorithm comes up with and how well it performs.

To summarize, the company wants to have the following two products:

* Product B1: Server-side algorithm functionality. An algorithm that supplements the missing [KSC](#KSC) data for profiles that can be integrated into the server as a compiled Python library.
* Product B2: Algorithm analysis tool. A stand-alone application to test the quality and performance (of sub-solutions) of [product B1](#ProductB1).

## Goal of this document

All of the requirements of these two products are discussed within this document. Every requirement, function and rule in this document have a certain urgency and importance factor. The requirements describe the functionality the application must offer, the functions describe the functionality a requirement must contain, and the rules describe the limitations or decisions that must be made for a requirement.

The urgency factor for a requirement indicates how early the requirement needs to be implemented. Requirements with a high ([H](#H)) urgency need to be implemented first before everything else. Requirements with a medium ([M](#M)) urgency should be implemented as soon as no high urgency requirements are left. Requirements with a low ([L](#L)) urgency can be implemented as soon as no medium urgency requirements are left.

The urgency factor for a function and rule have the same meaning as for a requirement. The only difference is that the urgency factor indicates how early the function and rule need to be implemented so that the requirement can be realized.

The importance factor for a requirement indicates which requirement have higher priority than other requirements. Requirements with a high ([H](#H)) priority indication need to be implemented no matter what, as soon as possible (this doesn't mean they have to be rushed!). Requirements with a medium ([M](#M)) priority also need to be implemented, but not necessarily straight away. Requirements with a low ([L](#L)) priority are things that would be nice to have (like an enhancement of the user experience/interface, additional features, etc.). This roughly translates to the [MoSCoW](#MoSCoW) notation as follows: high priority = must have, medium priority = should have, low priority = could have.

The importance factor for a function and rule also have the same meaning as for a requirement. The only difference is that the importance factor indicates the importance/necessity of implementation of functions and rules for a requirement.

## Intended audience and reading suggestions

This document is intended to be read by developers and product owners. This document should be read in the following way:

* Overall description;
* Product descriptions;
* Product requirements;
* Non-functional requirements;
* User Interface wireframes.

## Project scope

The main goal of this system is to supplement the missing skills for every profile. The scope of the system will therefore be focussed around the supplementation of the missing skills.

In [product B1](#ProductB1), only the profiles of LinkedIn are supported to predict the missing [KSC](#KSC) data for. Whenever *profiles* are mentioned within this document, only the *LinkedIn profiles* are meant (read reference [R.1](#R1) why this decision was made).

## References

|  |  |
| --- | --- |
| **Reference code** | **Reference** |
| R.1 | Research document |

# Overall description

## Product perspective

Product B1, the server-side algorithm functionality, serves as an extension for the current profile matching algorithm, which aims to improve the overall profile matching results by adding missing [KSC](#KSC) data to the profiles. This product will be integrated in the server's Analysis framework.

Product B2, the algorithm analysis tool, is a new self-contained product. This product should work without a internet connection and doesn't have any interconnection with other existing products or external databases.

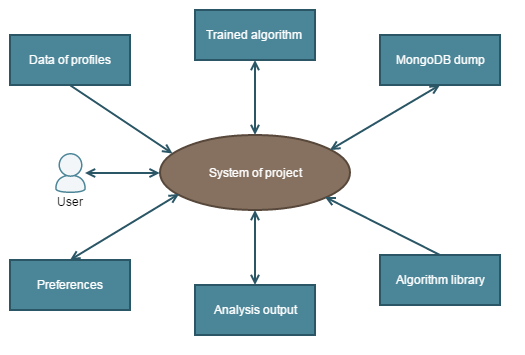


Figure 1 - Context view of the system

In Figure 1, you can see a context view of the system. The purpose of this context view is to provide an overall overview of things the system communicates with (be it files, databases, other systems, etc.). The things the system communicates with are represented by a block or symbol. There're seven "blocks" the system communicates with:

|  |  |  |
| --- | --- | --- |
| **Block** | **Communication direction** | **Description** |
| Trained algorithm | Ingoing and outgoing | The system can import and use a trained algorithm to make predictions for the missing skills in a profile. The system can also export a trained algorithm to an external target. |
| MongoDB dump | Ingoing and outgoing | The system can read profile data from and write it to a Mongo database. The profile data is used as the data source for the algorithm the user wants to analyze. |
| Algorithm library | Ingoing | The system can import an algorithm library so it can call its functions (for example the algorithm's prediction functions). |
| Analysis output | Ingoing and outgoing | The system can read the resulting data of an analysis of an algorithm from and write it to a file. |
| Preferences | Ingoing and outgoing | The system can read the preferences from and write it to a file. |
| User | Ingoing and outgoing | The user is able to provide input data for the system that it responds to. The system provides output data that's meant for the user. |
| Data of profiles | Ingoing | The system can read the data of profiles from a JSON file that will be used as the data source for the algorithm that predicts the missing skills for profiles. This data is [Big Data](#BigData). |

## Product functions

In this section, the major functions the product must perform or must let the user perform are briefly summarized. Details will be provided in chapter 3 and 4 for the different products separately.

### Product B1 - Server-side algorithm functionality

* The algorithm is able to predict missing skills for a selection of profiles.
* The algorithm can calculate a certainty score for every predicted skill per profile.

### Product B2 - Algorithm analysis tool

* The user can import an algorithm he wants to analyze.
* The user can provide a data source for the algorithm to use for the analysis.
* The user can save and load analysis results.
* The application can write/read the algorithm analysis result to/from a external target.
* The application can show the analysis results of the algorithm to the user. The results include the predictions, certainty scores of predictions and execution time.

## User classes and characteristics

Product B1 is a piece of algorithm functionality (library) for the server that will add data to the profiles. It won't have any interaction with users. However, this product can be used by the data scientists if they want to supplement the missing skills for the profiles.

Product B2 is a very specific product with a very specific goal: to analyze and test the algorithms that attempt to supplement the missing skills for the profiles. This product will mainly be used by the following users:

* The algorithm developer that develops the skill supplementation algorithm. He'll want to use the product to test algorithms on correct input and output, and performance. If the analysis is overall negative, it's a sign the algorithm needs to be improved. This user will use all of the product's functionalities frequently.
* Other users. The previous user is the main user of the product. There're also other users who don't want to use the product with a problem-solution mindset, but who're just interested what predictions the algorithm comes up with and how well it performs. It's anticipated these users only want to use the product's functionalities that allows them to see this information, occasionally.

## Operating environment

[Product B1](#ProductB1) will eventually be integrated in the Analysis framework, which runs on a UNIX-like server.

[Product B2](#ProductB2) is a stand-alone tool which has to be able to run on any operation system, assuming the user has installed the required libraries used by the product. This product must be able to run on systems without any network connection.

## Design and implementation constraints

* The documentation of both products have to be written in English.

### Product B1 - Server-side algorithm functionality

* The algorithm has to work with huge amounts of data which can't all be read into memory. There has to be a solution to split and read the data in smaller parts.
* The algorithm has to be able to be integrated in the server's Analysis framework (Python 2.7).
* The algorithm has to be able to predict missing skills of one profile in 1 second or less.
* Due to a limited amount of time and experience in the data science area, the algorithm will be written in Python 2.7 instead of possible better-performing alternatives. Python is also an ideal choice because the Analysis framework is written in Python.
* Only LinkedIn profiles are used to predict missing skills for (read reference [R.1](#R1) why this decision is made).

### Product B2 - Algorithm analysis tool

* The tool has to function without an internet connection.
* The tool must be able to run on any operating system.

## User documentation

There'll be additional documentation provided for [product B2](#ProductB2). This documentation will contain information how the product can be used. So basically, the documentation will be a user manual.

## Assumptions and dependencies

### Product B1 - Server-side algorithm functionality

* The algorithm won't communicate with the server's Apache Cassandra database. All save and read operations are done outside of the algorithm.
* The input data type for the algorithm must have a Python array format with dictionaries.
* The input data for the algorithm must contain all the necessary data fields for a typical LinkedIn profile.
* The product can use a pre-trained algorithm to make predictions.
* The product can train an algorithm to make predictions.

### Product B2 - Algorithm analysis tool

* Only a JSON file can be imported to be used as the data source for analyzing the algorithms for now.
* Only one algorithm analysis process can run at the same time.
* The user can't perform any actions during an algorithm analysis process.

# Requirements regarding the server-side algorithm functionality

## Overall product description

[Product B1](#ProductB1) can be seen as a algorithm package, which contains various different functions that return different values. The package will have functions that perform some pre-processing of the input data and the resulting data is returned. This data can then be used to call a function that will return predictions for the missing skills per profile. This function has two additional parameters which indicate if you:

1. want it to also return certainty scores per prediction (for every *predicted* skill);
2. want it to also return certainty scores of every skill in the system.

It's important to know that this product will not communicate with a external target. So reading any data from a external data source or writing any data to a external target has to be done outside this product.

## Features and rules

The following table contains all the features of this product (sorted by urgency and importance). More information about the features can be found in the upcoming chapters.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
| **a1** | The algorithm can pre-process the profile data of a selection of profiles to the wanted syntax. |  |  |  |  |  |  |
| **a2** | The algorithm can predict the missing skills for a selection of profiles. |  |  |  |  |  |  |
| **a5** | The algorithm can calculate a certainty score for every predicted skill in a profile. |  |  |  |  |  |  |
| **a3** | The algorithm can be trained to predict the missing skills. |  |  |  |  |  |  |
| **a4** | The trained algorithm can be exported to an external file. |  |  |  |  |  |  |
| **a6** | The algorithm can determine whether or not user-specified skills are credible. |  |  |  |  |  |  |
| **a7** | The algorithm can flag and exclude the incredible user-specified skills per profile from the data source. |  |  |  |  |  |  |

The following table contains all the rules of this product which are relevant for a feature (sorted by feature, urgency and importance). More information about the rules can be found in the upcoming chapters.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
| **a2r1** | The algorithm can only predict the missing skills if the input data has been pre-processed if pre-processing is required. |  |  |  |  |  |  |
| **a2r2** | The algorithm can only predict the missing skills if it's been trained |  |  |  |  |  |  |
| **a4r1** | A user-specified skill is considered to be credible if its credibility score is high enough. |  |  |  |  |  |  |
| **a4r2** | A user-specified skill is considered incredible if its credibility score is low enough. |  |  |  |  |  |  |

### Pre-process data

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm can pre-process the profile data of a selection of profiles to the wanted syntax. |  |  |  |  |  |  |

The algorithm needs to pre-process some of the profile data to be able to use it for further calculations. For instance, some data fields contain unwanted information and other fields need to be converted to a numeric representation of the data.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm reads the input data. |  |  |  |  |  |  |
|  | The algorithm removes unwanted information in the data per data field if necessary. |  |  |  |  |  |  |
|  | The algorithm converts the data per data field to another representation of the data if necessary. |  |  |  |  |  |  |
|  | The algorithm returns the pre-processed data. |  |  |  |  |  |  |

### Missing skill prediction

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm can predict the missing skills for a selection of profiles. |  |  |  |  |  |  |

The algorithm should be able to predict which skills are missing in every profile.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm reads the input data. |  |  |  |  |  |  |
|  | The algorithm performs operations to predict the missing skills per profile. |  |  |  |  |  |  |
|  | The algorithm returns the missing skills per profile. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm can only predict the missing skills if the input data has been pre-processed if pre-processing is required. |  |  |  |  |  |  |
|  | The algorithm can only predict the missing skills if it's been trained |  |  |  |  |  |  |

Pre-processing of the data isn't always required, unless conversions of the data to another format need to be done. For instance, if a string needs to be converted to a integer because the algorithm uses the integer values.

### Train algorithm

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm can be trained to predict the missing skills. |  |  |  |  |  |  |

The algorithm should be trainable with new data to find out how it performs when it's trained with this new data. This training could improve the algorithm's prediction performance.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm reads the pre-processed input data. |  |  |  |  |  |  |
|  | The algorithm trains itself on the input data. |  |  |  |  |  |  |
|  | The algorithm returns itself. |  |  |  |  |  |  |

### Export trained algorithm

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The trained algorithm can be exported to an external file. |  |  |  |  |  |  |

The trained algorithm should be able to export itself to an external file. The trained algorithm could then be used by [product B2](#ProductB2) for analysis purposes.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The trained algorithm exports itself to an external file. |  |  |  |  |  |  |

### Calculate certainty score of predictions

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm can calculate a certainty score for every predicted skill in a profile. |  |  |  |  |  |  |

It's important for the users to know how certain the algorithm is about the predicted skills. For instance, if the certainty score of a predicted skill is high even though that skill shouldn't be applicable for the profile in question, something could be wrong with the algorithm.

#### Functions

The following functions can start when function [a2f1](#a2f1) has been executed.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm calculates the certainty scores of a skill belonging to a profile. |  |  |  |  |  |  |
|  | The algorithm returns the certainty scores of skills that are predicted to belong to a profile. |  |  |  |  |  |  |
|  | The algorithm returns the certainty scores of skills that are predicted to not belong to a profile. |  |  |  |  |  |  |

### Control credibility of user-specified skills

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm can determine whether or not user-specified skills are credible. |  |  |  |  |  |  |

The profiles already have skills defined by their respective owners. However, it's possible that they don't own a skill even though they say they do. The algorithm should check the credibility of the person owning a particular skill.

#### Functions

The following functions can start when function [a2f1](#a2f1) has been executed.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm calculates the credibility score per user-specified skill per profile. |  |  |  |  |  |  |
|  | The algorithm returns the credibility score per user-specified skill per profile. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | A user-specified skill is considered to be credible if its credibility score is high enough. |  |  |  |  |  |  |
|  | A user-specified skill is considered incredible if its credibility score is low enough. |  |  |  |  |  |  |

### Exclude incredible user-specified skills

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm can flag and exclude the incredible user-specified skills per profile from the data source. |  |  |  |  |  |  |

Excluding incredible skills from profiles can improve 8vance's matching algorithm to create more accurate matches. Skills from one profile can be deemed incredible if there's no overlap in these skills with the skills of other similar profiles.

#### Functions

The following functions can start when function [a4f1](#a4f1) has been executed.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The algorithm adds a flag to the user-specified skills per profile if the credibility score is low. |  |  |  |  |  |  |

# Requirements regarding the stand-alone application

## Overall product description

Product B2 is a stand-alone algorithm analysis tool. With this tool, users can import algorithms they want to analyze and select a data source they want the algorithm to work with. Once these things have been specified, an analysis can be run for the algorithm. The analysis contains information about what predictions are made, a certainty score for every prediction and the execution time of the algorithm. This information can be used to identify flaws and strengths in the algorithm.

After running an analysis, the user is able to scroll through the list of profiles and view the predictions per profile by selecting a profile from the list. The user can also search for a profile by providing a profile id and/or prediction value (a skill) the profile must contain.

And finally, the user can import and export analysis results of an algorithm so he can look back to and compare previous results.

## Features and rules

The following table contains all the features of this product (sorted by urgency and importance). More information about the features can be found in the upcoming chapters.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
| **a6** | The user can select an algorithm he wants to analyze. |  |  |  |  |  |  |
| **a7** | The user can select a data source for the algorithm. |  |  |  |  |  |  |
| **a8** | The user can specify a dump target which is used to dump the pre-processing data for the data source of an algorithm. |  |  |  |  |  |  |
| **a10** | The user can run an analysis of the algorithm. |  |  |  |  |  |  |
| **a13** | The user can view the analysis results of a particular profile. |  |  |  |  |  |  |
| **a19** | The user can select a trained algorithm that can be passed to the algorithm the user wants to analyze. The latter algorithm uses the trained algorithm to create predictions. |  |  |  |  |  |  |
| **a18** | The application can clear the current analysis result data. |  |  |  |  |  |  |
| **a11** | The user can kill the analysis process when an analysis is running. |  |  |  |  |  |  |
| **a14** | The user can search for the analysis results of profiles. |  |  |  |  |  |  |
| **a9** | The user can test a specified database connection for validation. |  |  |  |  |  |  |
| **a12** | The user is able to view the analysis' progress. |  |  |  |  |  |  |
| **a15** | The user can search for predictions (skills) when viewing the analysis result of a profile. |  |  |  |  |  |  |
| **a16** | The user can export the algorithm analysis result. |  |  |  |  |  |  |
| **a17** | The user can import an algorithm analysis result. |  |  |  |  |  |  |

The following table contains all the rules of this product which are relevant for a feature (sorted by feature, urgency and importance). More information about the rules can be found in the upcoming chapters.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
| **a6r1** | Only one algorithm can be selected at a time. |  |  |  |  |  |  |
| **a6r2** | An algorithm needs to have the required function that returns predictions and certainty scores for all predictions to be valid. |  |  |  |  |  |  |
| **a7r1** | A newly selected data source will overwrite the previously selected data source. |  |  |  |  |  |  |
| **a7r2** | A data source can only be selected if an algorithm has been selected. |  |  |  |  |  |  |
| **a7r3** | A data source can only be selected if a dump target has been specified (see requirement [a8](#a8)). |  |  |  |  |  |  |
| **a7r4** | If a data source has already been selected and an error occurs when the user tries to select a new data source, the old data source will continue being used as the current data source. |  |  |  |  |  |  |
| **a9r1** | The database connection can only be tested if all fields for the specification of the database connection are specified. |  |  |  |  |  |  |
| **a9r2** | A database connection is deemed to be valid if a connection to the database could be made in 10 seconds. |  |  |  |  |  |  |
| **a11r1** | An analysis process has to be running to be able to kill an analysis process. |  |  |  |  |  |  |
| **a12r1** | The analysis' progress is only shown when an analysis is still running. |  |  |  |  |  |  |
| **a13r1** | The analysis result of one particular profile can only be viewed after a successful algorithm analysis. |  |  |  |  |  |  |
| **a13r2** | The analysis results of multiple profiles can be viewed at a time. |  |  |  |  |  |  |
| **a14r1** | The user can only search for analysis results of profiles after a successful analysis process. |  |  |  |  |  |  |
| **a15r1** | The user can only search for the analysis result of predictions in a profile if the analysis results of profiles is currently being viewed (see requirement [a13](#a13)). |  |  |  |  |  |  |
| **a16r1** | An analysis result can only be exported after a successful analysis (see requirement [a10](#a10)). |  |  |  |  |  |  |
| **a18r1** | The application will clear the current analysis result when a new algorithm analysis starts (see requirement [a10](#a10)). |  |  |  |  |  |  |
| **a18r2** | The application will clear the current analysis result when an algorithm analysis result is imported (see requirement [a17](#a17)). |  |  |  |  |  |  |
| **a19r1** | A newly selected trained algorithm will overwrite the previously selected trained algorithm. |  |  |  |  |  |  |
| **a19r2** | A trained algorithm can only be selected if an algorithm has been selected that the user wants to analyze. |  |  |  |  |  |  |

### Select algorithm

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can select an algorithm he wants to analyze. |  |  |  |  |  |  |

The user should be able to select all kinds of algorithms he wants to analyze. However, the algorithm must have the necessary functions for it to be able to be analyzed. One of the functions must return predictions and certainty scores of the predictions. The algorithm can also have multiple pre-processing functions that the application need to be able to identify.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application has a UI element that allows the user to select an algorithm when it's triggered. |  |  |  |  |  |  |
|  | The application offers a means for the user to select an algorithm. |  |  |  |  |  |  |
|  | The application offers a means for the user to the cancel the selection of an algorithm. |  |  |  |  |  |  |
|  | The application shows the selected algorithm on the UI. |  |  |  |  |  |  |
|  | The application warns the user if he wants to select a new algorithm if an algorithm has already been selected. |  |  |  |  |  |  |
|  | The application checks if the algorithm has the required function that returns predictions and certainty scores for all predictions. |  |  |  |  |  |  |
|  | The application checks if the algorithm has any pre-processing functions, and if so it saves their names. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | Only one algorithm can be selected at a time. |  |  |  |  |  |  |
|  | An algorithm needs to have the required function that returns predictions and certainty scores for all predictions for it to be valid. |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Selecting an algorithm to analyze. |
| **Description** | 1. The user triggers an UI element to indicate he wants to select an algorithm to analyze. 2. The application shows a means for the user to select an algorithm. [1] 3. The user selects an algorithm. [2] 4. The application checks if the algorithm has the required function that returns predictions and certainty scores for all predictions. [3] 5. The application checks if the algorithm has any pre-processing functions and saves their names. 6. The application closes the means for the user to select an algorithm. 7. The application saves the selected algorithm as the selected algorithm. 8. The application shows the selected algorithm on the UI. |
| **Exceptions** | [1] If an algorithm has already been selected, the application shows a warning message saying that an algorithm has already been selected. The user can indicate if he wants to proceed or cancel the selection. If the user chooses to cancel the selection, the use case ends right then.  [2] The user is also able to cancel the selection of an algorithm. If the user cancels the selection, the use case ends right then.  [3] The application doesn't have the required function that returns predictions and certainty scores. The application informs the user about this. The use case ends right then. |
| **Result** | An algorithm has been selected to be analyzed. |

### Select data source for the algorithm

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can select a data source for the algorithm. |  |  |  |  |  |  |

The algorithm will need some data so it can be executed and analyzed. The user can select a data source (e.g. a file). It's important that the structure of the data source is the same as the one used to create the algorithm. The application needs to check if the selected data source is valid.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application has an UI element that allows the user to select a data source when it's triggered. |  |  |  |  |  |  |
|  | The application offers a means for the user to select a data source. |  |  |  |  |  |  |
|  | The application offers a means for the user to cancel the selection of a data source. |  |  |  |  |  |  |
|  | The application shows the selected data source on the UI. |  |  |  |  |  |  |
|  | The application writes the data from the data source to the dump target if no data from that data source is found in the dump target (see requirement [a8](#a8)). |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | A newly selected data source will overwrite the previously selected data source. |  |  |  |  |  |  |
|  | A data source can only be selected if an algorithm has been selected. |  |  |  |  |  |  |
|  | A data source can only be selected if a dump target has been specified (see requirement [a8](#a8)). |  |  |  |  |  |  |
|  | If a data source has already been selected and an error occurs when the user tries to select a new data source, the old data source will continue being used as the current data source. |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Selecting a data source for the algorithm. |
| **Description** | 1. The user triggers an UI element to indicate he wants to select a data source for the algorithm. 2. The application shows a means for the user to select a data source. [1] 3. The user selects a data source. [2] 4. The application closes the means for the user to select a data source. 5. The application checks and notifies the user if any data from the selected data source has ever been written to the dump target. If it has, continue to step 6. If it hasn't, continue to step 7. 6. The application queries the user if he wants to use the data saved in the dump target, or if he wants to overwrite this data with the new data source's data. The application will overwrite the data if the latter option is chosen. Continue to step 8. 7. The application writes the data from the data source to the dump target. 8. The application saves the selected data source as the current data source. 9. The application shows the selected data source on the UI. |
| **Exceptions** | [1] If a data source has already been selected or specified, the application shows a warning message saying that an algorithm has already been selected or specified. The user can indicate if he wants to proceed or cancel the selection. If the user chooses to cancel the selection, the use case ends right then.  [2] The user is also able to cancel the selection of a data source. If the user cancels the selection, the use case ends right then. |
| **Result** | A data source for the algorithm has been selected. |

### Specify dump target

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can specify a dump target which is used to dump the pre-processing data for the data source of an algorithm. |  |  |  |  |  |  |

The dump target will be a locally installed MongoDB which is highly effective when dealing with large amounts of data. All the pre-processed data of the algorithm will be written to this dump target for future reference.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application has an UI element that offers the user to specify a dump target when it's triggered. |  |  |  |  |  |  |
|  | The application offers a means for the user to specify a dump target. |  |  |  |  |  |  |
|  | The application offers a means for the user to cancel the specification of a dump target. |  |  |  |  |  |  |
|  | If a dump target has been specified before, the information of this dump target will appear which can be modified by the user. |  |  |  |  |  |  |
|  | The application validates the dump target connection (see requirement [a9](#a9)). |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Specifying a dump target. |
| **Description** | 1. The user triggers an UI element to indicate he wants to specify the dump target. 2. The application shows a means for the user to specify the dump target. [1] 3. The user specifies the dump target. [2] 4. The user starts a connection test to validate if a connection can be made with the specified dump target (see requirement [a9](#a9)). [3] 5. The application closes the means for the user to specify the dump target. 6. The application saves the specified dump target. |
| **Exceptions** | [1] If a dump target has been specified before, the information for this dump target will appear which can be modified by the user.  [2] The user is also able to cancel the specification of the dump target. If the user cancels the specification, the use case ends right then.  [3] The connection is invalid. Go back to step 3. |
| **Result** | A dump target has been specified. |

### Validate specified database connection

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can test a specified database connection for validation. |  |  |  |  |  |  |

The user can specify a data source connection (see requirement [a7](#a7)). The user must test if a connection can be made to his specified database to determine if the specified connection is valid.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application has an UI element that offers the user to test the specified database connection when it's triggered. |  |  |  |  |  |  |
|  | The application tests if a connection can be made to the specified database. |  |  |  |  |  |  |
|  | The application returns whether or not the specified database connection is valid or invalid. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The database connection can only be tested if all fields for the specification of the database connection are specified. |  |  |  |  |  |  |
|  | A database connection is deemed to be valid if a connection to the database could be made in 10 seconds. |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Validating a specified database connection. |
| **Description** | 1. The user triggers an UI element to indicate he wants to test if a connection can be made to the specified database. 2. The system creates a connection to the specified database. [1] 3. The system deems the connection to be valid and notifies the user about this. |
| **Exceptions** | [1] The system can't create a connection to the specified data source. The system deems the connection to be invalid and notifies the user about this. The use case ends right then. |
| **Result** | The data from the data source has been written to the dump target. |

### Run analysis of algorithm

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can run an analysis of the algorithm. |  |  |  |  |  |  |

The user is able to run an analysis process to analyze the algorithm on performance and accuracy. The user can go over this analysis to check whether or not the made predictions are correct and expected. If they're not, something is likely wrong with the algorithm. The user can also view the execution times to check if the algorithm is fast enough in making its predictions.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application has an UI element that offers the user to start an analysis of the selected algorithm when it's triggered. |  |  |  |  |  |  |
|  | The application clears the current analysis result data (see requirement [a18](#a18)). |  |  |  |  |  |  |
|  | The application queries the user if he's sure to start a new analysis which will overwrite the previous analysis data. |  |  |  |  |  |  |
|  | The application reads the data source's data from the dump target. |  |  |  |  |  |  |
|  | The application calls the algorithm's pre-processing functions in the right order if they're present to pre-process the data source's data and write it in the dump target. |  |  |  |  |  |  |
|  | The application calls the required algorithm function to train the algorithm if no trained algorithm was selected by the user. |  |  |  |  |  |  |
|  | The application calls the required algorithm function to get the predictions and certainty scores per prediction. |  |  |  |  |  |  |
|  | The application shows the result of the analysis on the UI. |  |  |  |  |  |  |
|  | The application checks and notifies the user if the data for the data source of this algorithm has ever been pre-processed and written to the dump target. |  |  |  |  |  |  |
|  | The application queries the user if he wants to pre-process the data again if the data has already been pre-processed before for the algorithm and data source. |  |  |  |  |  |  |
|  | The application measures the execution times of the algorithm functions. |  |  |  |  |  |  |
|  | The application offers a means for the user to kill the analysis process. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | An analysis of an algorithm can only be run if a data source is specified. |  |  |  |  |  |  |
|  | An analysis of an algorithm can only be run if no other analysis is running. |  |  |  |  |  |  |
|  | The user can't do any action apart from killing the analysis process when an analysis is running. |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Running an analysis on an algorithm |
| **Description** | 1. The user triggers an UI element to indicate he wants to analyze the selected algorithm. 2. The application queries the user if he's sure to start a new analysis which will overwrite the previous analysis data. If so, continue to step 3. Otherwise, the use case ends right then. 3. The application clears the current analysis result data (see requirement [a18](#a18)). 4. The application reads the data source's data from the dump target. [1] 5. The application checks if any pre-processing functions are present in the algorithm. If so, continue to step 4. Otherwise continue to step 8. [1] 6. The application checks and notifies the user if the data for the data source of this algorithm has ever been pre-processed and written to the dump target. [1] 7. The application queries the user if he wants to pre-process the data again if the data has already been pre-processed before for the algorithm and data source. If so, continue to step 6. Otherwise continue to step 8. [1] 8. The application calls the algorithm's pre-processing functions to pre-process the data. [1] 9. The application writes the pre-processed data for the algorithm and data source in the dump target. [1] 10. The application calls the required algorithm function to train the algorithm if no trained algorithm was selected by the user. [1] 11. The application calls the required algorithm function to get the predictions and certainty scores per prediction for this data. [1] 12. The application measures the execution times of the functions. [1] 13. The application shows the result of the analysis on the UI. [1] |
| **Exceptions** | [1] The user has killed the analysis process. The use case ends right then. |
| **Result** | An analysis on an algorithm has been run. |

### Kill analysis process

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can kill the analysis process when an analysis is running. |  |  |  |  |  |  |

The user is able to kill the analysis process. Killing the analysis process could be useful if the algorithm takes too long or the user made a mistake and started the analysis process prematurely.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application has an UI element that allows the user to kill the analysis process when it's triggered. |  |  |  |  |  |  |
|  | The application kills the analysis process and informs the user about it. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | An analysis process has to be running to be able to kill an analysis process. |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Killing an analysis process |
| **Description** | 1. The user triggers an UI element to indicate he wants to kill the analysis process. 2. The application queries the user if he's sure to kill the analysis process. If so, continue to step 3. Otherwise, the use case ends. 3. The application kills the analysis process. 4. The application informs the user the analysis process has been killed. |
| **Exceptions** |  |
| **Result** | An analysis process has been killed. |

### View analysis' progress

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user is able to view the analysis' progress. |  |  |  |  |  |  |

Providing continuous feedback on the analysis' progress to the user ensures the user the analysis process is still running and hasn't frozen.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application shows the analysis' progress on the UI. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The analysis' progress is only shown when an analysis is still running. |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Viewing the analysis' progress |
| **Description** | 1. The user has started an analysis process. 2. The application continuously updates the analysis' progress on the UI. [1] |
| **Exceptions** | [1] The user has killed the analysis process. The use case ends right then.  [2] The analysis has finished. The use case ends right then. |
| **Result** | The analysis' progress can be viewed. |

### View analysis results of a profile

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can view the analysis results of a particular profile. |  |  |  |  |  |  |

The user can view the analysis results of a particular profile by selecting that profile. The analysis results of a profile includes information about the predictions that were made and the certainty scores of those predictions for this profile.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application has an UI element that allows the user to view the analysis results of a particular profile when it's triggered. |  |  |  |  |  |  |
|  | The application can show the analysis results of a particular profile. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The analysis result of one particular profile can only be viewed after a successful algorithm analysis. |  |  |  |  |  |  |
|  | The analysis results of multiple profiles can be viewed at a time. |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Viewing the analysis result of one particular profile |
| **Description** | 1. The user triggers an UI element to indicate he wants the view the analysis result of one particular profile. 2. The application shows the analysis result of this particular profile. |
| **Exceptions** |  |
| **Result** | The analysis result of one particular profile is visible. |

### Search for analysis results of profiles

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can search for the analysis results of profiles. |  |  |  |  |  |  |

The analysis result can include information about hundreds of profiles. With a search option, the user can quickly find the analysis results of profiles he's interested in.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application has an UI element that allows the user to search for a particular profile when it's triggered. |  |  |  |  |  |  |
|  | The search criteria can contain profile ids to search for. |  |  |  |  |  |  |
|  | The application shows the profiles that match the search criteria. |  |  |  |  |  |  |
|  | The search criteria can contain predictions (skills) to search for. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can only search for analysis results of profiles after a successful analysis process. |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Searching for the analysis results of profiles |
| **Description** | 1. The user specifies the search criteria. 2. The user triggers an UI element to indicate he wants to search for profiles matching the search criteria. 3. The application shows the profiles that match the search criteria. [1] |
| **Exceptions** | [1] If the user hasn't specified a search criteria, the application shows all the profiles. |
| **Result** | The profiles that match the search criteria are visible. |

### Search for predictions in profile's analysis result

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can search for predictions (skills) when viewing the analysis result of a profile. |  |  |  |  |  |  |

A lot of predictions can be made for a certain profile. If the user is only interested to know if certain predictions are made in a profile, he can search for these predictions to check if they are present in the profile.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application has a UI element that allows the user to search for the analysis result of predictions (skills) in a profile when it's triggered. |  |  |  |  |  |  |
|  | The search criteria can contain predictions (skills) to search for. |  |  |  |  |  |  |
|  | The application shows the analysis result of the predictions that match the search criteria. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can only search for the analysis result of predictions in a profile if the analysis results of profiles is currently being viewed (see requirement [a13](#a13)). |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Searching for the analysis result of predictions in a profile |
| **Description** | 1. The user specifies the search criteria. 2. The user triggers an UI element to indicate he wants to search for the analysis results of predictions matching the search criteria. 3. The application shows the analysis result of predictions that match the search criteria. [1] |
| **Exceptions** | [1] If the user hasn't specified a search criteria, the application shows the analysis result of all predictions. |
| **Result** | The analysis result of predictions that match the search criteria is visible. |

### Export analysis result

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can export the algorithm analysis result. |  |  |  |  |  |  |

The user is able to export the algorithm analysis result to a external file. This way the user can review past results of algorithm analyses.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application has a UI element that allows the user to export the algorithm's analysis result when it's triggered. |  |  |  |  |  |  |
|  | The application exports the analysis result to a external file. |  |  |  |  |  |  |
|  | The application notifies the user when analysis results has been exported successfully. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | An analysis result can only be exported after a successful analysis (see requirement [a10](#a10)). |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Exporting the algorithm analysis result |
| **Description** | 1. The user triggers a UI element to indicate he wants to export the algorithm analysis result. 2. The application queries the user where he wants to export it to. 3. The user specifies the location to export it to. 4. The application exports it to the specified location and notifies the user. |
| **Exceptions** |  |
| **Result** | The algorithm analysis result is exported. |

### Import analysis result

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can import an algorithm analysis result. |  |  |  |  |  |  |

The user is able to import an algorithm analysis result from a external file. This way the user can review past results of algorithm analyses.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application has a UI element that allows the user to import an algorithm's analysis result when it's triggered. |  |  |  |  |  |  |
|  | The application offers a means for the user to select the analysis result he wants to import. |  |  |  |  |  |  |
|  | The application imports the selected analysis result. |  |  |  |  |  |  |
|  | The application shows the analysis result when the analysis result has been imported successfully. |  |  |  |  |  |  |
|  | The application clears the current analysis result data (see requirement [a18](#a18)). |  |  |  |  |  |  |
|  | The application offers a means for the user to cancel the selection of the analysis result. |  |  |  |  |  |  |
|  | The application queries the user if he's sure to import the analysis result data which will overwrite the previous analysis data. |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Importing the algorithm analysis result |
| **Description** | 1. The user triggers a UI element to indicate he wants to import an algorithm analysis result. 2. The application queries the user if he's sure to import the analysis result data which will overwrite the previous analysis data. 3. The application queries the user which analysis result he wants to import. 4. The user selects the analysis result to import. [1] 5. The application clears the current analysis result data (see requirement [a18](#a18)). 6. The application imports the analysis result and shows it on the UI. |
| **Exceptions** | [1] The user is also able to cancel the selection of an analysis result. The use case ends right then. |
| **Result** | The algorithm analysis result is exported. |

### Clear analysis data

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application can clear the current analysis result data. |  |  |  |  |  |  |

The application can clear the current analysis result data when a new algorithm analysis starts (see requirement [a10](#a10)) or when an analysis result is imported (see requirement [a17](#a17)). This means that all the data of the current analysis result will be unloaded from the UI components. Any additionally opened window containing data of this analysis will be closed. All of this is done to prepare the UI for the new analysis results.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application clears all the data of the current algorithm analysis result. |  |  |  |  |  |  |
|  | The application unloads all the data of the current algorithm analysis result from the UI components. |  |  |  |  |  |  |
|  | The application closes all additionally opened window which contains data of the current algorithm analysis result. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application will clear the current analysis result when a new algorithm analysis starts (see requirement [a10](#a10)). |  |  |  |  |  |  |
|  | The application will clear the current analysis result when an algorithm analysis result is imported (see requirement [a17](#a17)). |  |  |  |  |  |  |

### Select trained algorithm

#### Description and priority

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The user can select a trained algorithm that can be passed to the algorithm the user wants to analyze. The latter algorithm uses the trained algorithm to create predictions. |  |  |  |  |  |  |

The user can select a trained algorithm that can be passed to the algorithm the user wants to analyze. The latter algorithm uses the trained algorithm to create predictions. If the user doesn't select a trained algorithm, the algorithm the user wants to analyze will have to train itself, because only a trained algorithm can make predictions.

#### Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Function description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | The application has an UI element that allows the user to select a trained algorithm when it's triggered. |  |  |  |  |  |  |
|  | The application offers a means for the user to select a trained algorithm. |  |  |  |  |  |  |
|  | The application offers a means for the user to cancel the selection of a trained algorithm. |  |  |  |  |  |  |
|  | The application shows the selected trained algorithm on the UI. |  |  |  |  |  |  |

#### Rules

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Rule description** | **Urgency** | | | **Importance** | | |
| [**H**](#H) | [**M**](#M) | [**L**](#L) | [**H**](#H) | [**M**](#M) | [**L**](#L) |
|  | A newly selected trained algorithm will overwrite the previously selected trained algorithm. |  |  |  |  |  |  |
|  | A trained algorithm can only be selected if an algorithm has been selected that the user wants to analyze. |  |  |  |  |  |  |

#### User actions

|  |  |
| --- | --- |
| **ID** |  |
| **Name** | Selecting a trained algorithm for the algorithm the user wants to analyze. |
| **Description** | 1. The user triggers an UI element to indicate he wants to select a trained algorithm for the algorithm that the user wants to analyze. 2. The application shows a means for the user to select a trained algorithm. [1] 3. The user selects a trained algorithm. [2] 4. The application closes the means for the user to select a trained algorithm. 5. The application saves the selected trained algorithm as the current trained algorithm. 6. The application shows the selected trained algorithm on the UI. |
| **Exceptions** | [1] If a trained algorithm has already been selected or specified, the application shows a warning message saying that a trained algorithm has already been selected or specified. The user can indicate if he wants to proceed or cancel the selection. If the user chooses to cancel the selection, the use case ends right then.  [2] The user is also able to cancel the selection of a trained algorithm. If the user cancels the selection, the use case ends right then. |
| **Result** | A trained algorithm for the algorithm that the user wants to analyze has been selected. |

# Non-functional requirements

## Performance requirements

This section contains the requirements that describe the responsiveness of the system.

### Product B1 - Server-side algorithm functionality

* The algorithm has to be able to predict missing skills of one profile in 1 second or less. Eventually, the algorithm has to make predictions for millions of profiles and the scraping of profile data will be a continuous process, meaning the algorithm needs to be able to keep up. Additionally, users can provide profile data themselves and the algorithm should be able to predict the missing skills in an instant.

## Safety requirements

This section contains the requirements that are concerned with safeguards or actions that prevent possible loss, damage, or harm that could result from the use of the system.

### Product B2 - Algorithm analysis tool

* Whenever the user wants to select/specify a new algorithm (requirement [a6](#a6)), data source (requirement [a7](#a7)), dump target (requirement [a8](#a8)), trained algorithm (requirement [a19](#a19)), or start an analysis process (requirement [a10](#a10)), kill an analysis process (requirement [a11](#a11)) or import an analysis result (requirement [a17](#a17)), the application must query the user if he's certain he wants to continue to prevent the loss of previous data.
* The user is able to kill an analysis process (requirement [a11](#a11)). When the user starts an analysis process, the user can't perform any actions in the application except from killing the analysis process. If there's a problem with the analysis resulting in the process to never end, that means the user could never perform any actions. By providing the user with the option to kill an analysis process, there's a safeguard to prevent this problem.

## Security requirements

TBD

## Software quality attributes

* *Adaptability*. [Product B1](#ProductB1) must be adaptable in different or evolving software products that support Python 2.7.
* *Availability.* Both products must be available for use 24/7.
* *Correctness.* [Product B1](#ProductB1) must be able to calculate certainty scores for every prediction with great precision.
* *Interoperability*. [Product B1](#ProductB1) and 8vance's Analysis environment as well as [product B2](#ProductB2) and [product B1](#ProductB1) must be able to exchange information with each other.
* *Modifiability*. Both products must be easily modifiable to improve the product quality.
* *Installability*. [Product B2](#ProductB2) must be installable and thus usable on any operating system.
* *Reusability*. [Product B1](#ProductB1) must be usable in both 8vance's Analysis environment as well as in [product B2](#ProductB2).
* *Testability*. [Product B2](#ProductB2) will mainly be used to test the quality and performance of an algorithm. Multiple test criteria can be established that should measure the quality and performance of an algorithm. The input and output data itself of an algorithm can also be tested on correctness.
* *Learnability*. [Product B2](#ProductB2) must be easy to use. Apart from algorithm developers, users without any understanding of the inner workings of the algorithm and [product B2](#ProductB2) should be able to use and understand the intended use of the product.

## Business rules

* Anyone with access to either product can use all functionality it offers.

# User interface wireframes

(GUI sketches of product B2)

TBD