

# Review Document – Requirements Document

David Klopp

Christian Stricker

Markus Vieth

November 25, 2015

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>User-Requirements</b>	<b>3</b>
2.1	Mit Algorithmen "arbeiten" (UR002) . . . . .	3
2.2	Daten (UR004) . . . . .	3
2.3	Ordnerstruktur (UR005) . . . . .	3
2.4	Superadmin (UR006) . . . . .	4
2.5	Trainingsdaten (UR015) . . . . .	4
<b>3</b>	<b>Non-Functional Requirements</b>	<b>5</b>
3.1	Nutzerdaten sollen "sicher" sein (NFR010) . . . . .	5
3.2	Speicherort der Passwörter (NFR013) . . . . .	5
<b>4</b>	<b>Functional Requirements</b>	<b>5</b>
4.1	Protokoll (FR002) . . . . .	5
4.2	Warteschlange (FR007) . . . . .	6
4.3	Ordner (FR016) . . . . .	6
4.4	"verschiedene Server" (FR017) . . . . .	6
4.5	Sicherheitslücke (FR020) . . . . .	7
4.6	Sicherheitslücke (FR021) . . . . .	7
4.7	Erweitertes GUI und erweiterte Funktionen (FR019) . . . . .	7
4.8	Default-Werte (FR003,FR004,) . . . . .	8

## 1 Introduction

Der Gruppe 16 sind einige Fehler beim Formulieren des Requirement Documents unterlaufen, welche im folgenden behandelt werde.

## 2 User-Requirements

### 2.1 Mit Algorithmen "arbeiten" (UR002)

**Statement:**

"Das System soll mit den Klassifikationsalgorithmen J48, Random Forest und SMO arbeiten können."

**Problem:**

Die Formulierung "mit Klassifikationsalgorithmen [...] arbeiten" ist missverständlich, da der Eindruck entsteht, als solle das Programm in der Lage sein Algorithmen zu editieren.

**Lösung:**

Verwende anstelle des Begriffs "arbeiten" die Formulierung "zur Verfügung stellen", um deutlich zu machen, dass diese Algorithmen bereits im System integriert sind, um Modelle zu generieren.

### 2.2 Daten (UR004)

**Statement:**

"Das System soll alle Daten in einer Datenbank verwalten.."

**Problem:**

Die Datenbank soll z.B. Pakete, Modelle und Parameter enthalten. Das Wort "alle" legt die Vermutung nahe, dass auch Algorithmen und Nutzerdaten in dieser Datenbank gespeichert werden sollen.

**Lösung:**

Zähle explizit jene Daten auf die in der Datenbank gespeichert werden sollen.

### 2.3 Ordnerstruktur (UR005)

**Statement:**

"Die Datenbank soll eine Ordnerstruktur verwalten."

**Problem:**

Der Begriff Ordnerstruktur legt nahe, dass die Datenbank einen Ordner abspeichern soll. Dies ist allerdings nicht möglich.

**Lösung:**

Da mit dem Begriff "Ordnerstruktur" im obigen Statement der Aufbau eines durch den Nutzer erstellten Paketes gemeint ist, sollte man an dieser Stelle eher Paketen sprechen.

**2.4 Superadmin (UR006)****Statement:**

"Das System soll durch einen Superadmin verwaltet und gewartet werden."

**Problem:**

Laut Definition des Superadmin unterscheidet sich dieser in keiner Weise von einem normalen Systemadministrator.

**Lösung:**

Oftmals wird von Admin gesprochen wenn eigentlich ein Nutzer gemeint ist. Wenn man diesen Fehler beseitigt, erübrigt sich die Notwendigkeit des Begriffes "Superadmin", da sich dieser nun durch "Admin" ersetzen lässt.

**2.5 Trainingsdaten (UR015)****Statement:**

"Der Benutzer soll auf bestehende Modelle Trainingsdaten anwenden können, sofern er die Berechtigung dazu hat."

**Problem:**

Trainingsdaten können nicht auf ein bestehendes Modell angewendet werden, sondern lediglich zur Generierung eines solchen verwendet werden.

**Lösung:**

Gemeint ist das Anwenden von Daten auf ein Modell zur Verifizierung der Funktionsfähigkeit des Selbigen. Man sollte im obigen Statement daher von Verifizierungs- oder Überprüfungsdaten anstelle von Trainingsdaten sprechen.

## System Requirements

### 3 Non-Functional Requirements

#### 3.1 Nutzerdaten sollen "sicher" sein (NFR010)

**Statement:**

"Alle Nutzerdaten sollen sicher sein und werden nicht außerhalb des Systems verwendet."

**Problem:**

Der Begriff "sicher" ist nicht präzise, da z.B sowohl gemeint sein könnte, dass die Daten frei von Schadcode sein sollen oder aber das kein Datenverlust auftreten soll.

**Lösung:**

"Die Nutzerdaten sollen verschlüsselt auf dem Server abgespeichert werden." trifft die gemeinte Aussage des Statements besser.

#### 3.2 Speicherort der Passwörter (NFR013)

**Statement:**

"Passwörter der Benutzer sollen mit SHA-256 und einem Salt sicher verschlüsselt sein."

**Problem:**

Es ist nicht definiert wo und in welcher Form die angegebenen Daten abgespeichert werden sollen.

**Lösung:**

Ergänze das Statement so, dass klar hervorgeht das die Daten in der Datenbank auf dem Server mit oben angegebener Verschlüsselung abgespeichert werden sollen.

### 4 Functional Requirements

#### 4.1 Protokoll (FR002)

**Statement:**

"Das System soll die übergebenen Datensätze auf dem Server verarbeiten."

**Problem:**

Das Functional Requirement ist ungenau und nicht messbar. Es wird weder genannt, in wie fern Datensätze verarbeitet werden, noch was mit ihnen passieren soll.

**Lösung:**

Als User Requirement notieren oder streichen

**4.2 Warteschlange (FR007)****Statement:**

"Es sollen standardmäßig zwei Modelle gleichzeitig berechnet werden können. Alle weiteren Berechnungen sind in einer Warteschlange. Der Superadmin soll einstellen können, wie viele gleichzeitig berechnet werden können."

**Problem:**

Die Art der Warteschlange ist nicht definiert.

**Lösung:**

Die Warteschlange könnte als first in, first out festgelegt werden.

**4.3 Ordner (FR016)****Statement:**

"Ein Ordner soll Paket heißen und Trainingsdatensatz, Testdatensatz, Modell und Ergebnisdatensätze enthalten. Pakete können geschachtelt werden."

**Problem:**

Der Begriff Ordner an dieser Stelle ist falsch/ungenau.

**Lösung:**

Das Wort Ordner streichen.

**4.4 "verschiedene Server" (FR017)****Statement:**

"Das System soll auf verschiedene Server zugreifen können."

**Problem:**

Die Formulierung ist mehrdeutig. Ein System soll auf einem Server laufen. Der Zugriff auf andere Datenbanken erfolgt über das REST-Interface.

**Lösung:**

Genauer ausformulieren.

**4.5 Sicherheitslücke (FR020)****Statement:**

”Algorithmen sollen von angemeldeten Benutzern der System-Bibliothek hinzugefügt werden können.”

**Problem:**

Große Sicherheitslücke, da so Schadcode in das System gespielt werden kann.

**Lösung:**

Der Administrator überprüft die Algorithmen vor der Integration in das System auf ihre Funktionalität und eventuelle Schadsoftware.

**4.6 Sicherheitslücke (FR021)****Statement:**

”Modelle sollen als serialisiertes Java Objekt portierbar sein.”

**Problem:**

Große Sicherheitslücke, da so Schadcode in das System gespielt werden kann.

**Lösung:**

Modelle sind in einem anderem Format zu importieren oder durch einen Administrator zu prüfen.

**4.7 Erweitertes GUI und erweiterte Funktionen (FR019)****Statement:**

”Der Superadmin soll eine erweiterte GUI erhalten, in der seine erweiterten Funktionen ausführbar sind.”

**Problem:**

"'Erweitertes GUI'" und "'erweiterte Funktionen'" sind nicht weiter definiert.

**Lösung:**

Als User Requirement aufnehmen oder genauer definieren.

**4.8 Default-Werte (FR003,FR004,)****Statement:**

"Anonyme Benutzer sollen über eine Laufzeitbeschränkung reguliert sein, welche der Superadmin einstellen kann."

"Anonyme Benutzer sollen ein Upload-Limit haben, über welches der Superadmin verfügen kann."

**Problem:**

Es werden keine Default-Werte genannt.

**Lösung:**

Default-Werte angeben.