ML Data Classification System: Software Requirements Specification Version 2.0

Data Plumbers

We Give Your Data the Pipe™

Computer Science Department
California Polytechnic State University
San Luis Obispo, CA USA

October 24, 2018

Contents

Credits

Name	Date	Role	Version
Zachary Richardson	October 8, 2018	Document Owner	1.0
Tony Chen	October 8, 2018	Document Owner	1.0, 2.0
Bennett Robinson	October 8, 2018	Document Owner	1.0
Kyler Ramsey	October 8, 2018	Document Owner	1.0
Samuel Nayerman	October 8, 2018	Document Owner	1.0

Revision History

Name	Date	Reason for Changes	Version
Data Plumbers	October 9, 2018	Initial document. Added Section 1, 3's Use Cases, 4's Functional Requirement, and 6's Non-Functional Requirements	1.0
Tony Chen	October 24, 2018	Added information for sections 2.1, 2.4, 2.5, 2.6, 2.7, 2.8, and Glossary. Added FR-6 and NFR-1B.	2.0
Sam Nayerman	October 24, 2018	Added business rules for sections 2.9, modified personas to comply with feedback	2.0
Kyler Ramsey	October 24, 2018	Added assumptions and dependencies for section 2.8	2.0

1 Introduction

1.1 Purpose

The Data Discovery Workbench, and - more specifically - its Data Classifier component, are presented and outlined herein. As designed, planned, and implemented by the Data Plumbers development team, the purpose of this system is to aid in the classification of diverse or otherwise non-standardized data-sets for the later use of other data cataloging and anonymizing software tools and components. The system's ability to intelligently interface with data-sets that do not provide an explicit schema or ontology will allow the system to be used as a tool to unify and interface between sources of data that would otherwise be difficult to use with one-another. The system's primary function will be to elicit a set of classes or categories from select data sets which are provided to it, while allowing users to add or alter the generated classifications to their liking. This document will serve as the definitive reference for the planning and implementation of this project, with specific regard to the various requirements and use cases which are expected of it.

1.2 Document Conventions

Term	Definition
FR	Functional requirement
NFR	Non-Functional requirement
MarkLogic	American software business that develops and provides an enterprise NoSQL database
GUI	A graphical user interface

1.3 Intended Audience and Reading Suggestions

This document is intended for the developers of this project (our CSC 402 team) as well as user's and tester's of our data classification application. It is designed to serve as a reference for future design, planning, and implementation of this product. This SRS document contains an overview of our product and system features, its intended audience, important use cases, and the primary functional and non-functional requirements of our software. Beginning at the overview section is advised, then a glance of the functional and non-functional requirements to establish context is recommended before proceeding to our use case section, which can be quite detailed.

1.4 Project Scope

The goal of this project is to provide a data classification component within the Data Discovery Workbench system in order to help reduce the cost and time of a industrial or academic data consumer in discovering and migrating complex data sets.

For further information, reference the team's Vision and Scope document:

https://github.com/DataPlumbers/Documentation/blob/master/Vision_Document.pdf

2 Overall Description

2.1 Product Perspective

MarkLogic provides software solutions that allows its users to be productive in dealing with large amount of data, such as having the ability to search through a pool of datasets from various data sources in a single application interface, but what MarkLogic does not have is a system that automatically classifies the users' pool of input data. The product, the ML Data Classification System, is one of the main components of the Data Discovery Workbench that will allow data scientists or industrial data consumers to efficiently categorize and classify large amount of datasets in a way that will make the task of data discovery and categorization to be less time consuming and error-prone.

2.2 Product Features

The system will have several key features as annotated in Section 3. The system will need to allow users to view datasets, at various points in the training cycle, as well as upload custom datasets. Sets will need to have the capability to be labeled as sensitive, so that only specific users may view the sensitive information. Building off of that, users will also need to be able to view both sensitive and non-sensitive data with the information fields anonymized - likewise, that data will be stored anonymously as well.

2.3 User Personas

2.3.1 Business Graduate

Author: Sam

Stan is a recent business graduate from San Diego State University. Stan's always had the idea to open a microbrewery in the heart of San Diego, and market the perfect IPA. Helping Stan is his long-time friend, and beer expert, Josh. Unlike Stan, Josh has next to no experience running a business, managing inventory or any other duties required to run a successful business. Together they have a chance at success, but alone they will fail.

Stan and Josh have invested their savings and bought a warehouse where they can begin the brewing process. However, the process of ordering beer-making materials in bulk is not as simple as they would have anticipated! Items come in labeled in strange codes, which are seemingly undecipherable. Stan then remembers his Software Engineer friend from school who was telling them about managing inventory through a database management system - and Mark Logic comes up as an ideal candidate during their search.

Mark Logic sold the young entrepreneurs with their Database Ordering Service, which promised to label and categorize the mens' inventory, and allow for it to be viewed easily. Even with Josh's paranoia over cyber-security, they men figure out a perfect solution - anonymize the data; a feature that Mark Logic offers with their service! Perfect!

2.3.2 Business Analyst

Author: Zach

David is a 22 year old graduate of SomeName State University, where he majored in Business. He hadn't found a job after graduating, so he reached out to his uncle (Elon Musk) who was able to get him a job at a mid-sized software company as an analyst. David is unsure exactly what his duties will be, or what relevant skills he has to add to the company, but he gladly accepted the job.

As it turns out, David's company uses MarkLogic as a database software. Part of David's job is analyzing and merging different datasets across the company. David has little technical knowledge concerning databases, but a MarkLogic representative recommended that he take a look at their "Data Classifier" tool, promising that it would simplify the process.

David finds that his company had multiple data sets represented as CSV files, and decides to try out the recommended MarkLogic tool. To his surprise, he is able to load up his datasets, categorize them, and make informed decisions based on the simple and easily understandable readouts from the tool!

2.3.3 Nurse practitioner

Author: Tony

Jennifer Hyde is a 24 years old nurse practitioner who is currently working at St. Mary's Hospital in the busy center of San Francisco. She recently graduated from UC Davis' Nursing program around 2 months ago. Her new job includes taking care of patients and occasionally working at the back of the office when needed by the other hospital staffs.

One day, one of the managers of the hospital has requested Jennifer to help in registering the new patients who are currently staying at the hospital. The information of the patients has already been sent electronically to the hospital by other medical clinics, usually in various file data formats such as CSV and Excel. Without having a confident level of expertise in handling data with advanced software technologies, she began manually looking through each patient data sets and manually typing each demographic information of every patient into the hospital's main registration portal. Jennifer spends most of her day doing data entry more than her other required duties.

Jennifer soon found a product online offered by a NoSQL database company called MarkLogic, who has recently released a Data Discovery Workbench with a data classifier, advertising to medical and research facilities about a service in data migration and catalog.

The web application was easy to use since no computer configurations were necessary on the hospital's part. Jennifer was able to upload the patient datasets to the data classifier that matches with a standard ontology of a patient record in the system. As a result, the hospital has adopted this software, reduced the staffs needed for the data entry work and the chance of human error in handling sensitive patient data.

2.3.4 Data Scientist

Author: Kyler

Leeroy Jenkins is male, age 34, and a graduate of Stanford University. He graduated Stanford to acquire a PHD in data science; prior to that he obtained his bachelors and masters in Computer Science from Cal Poly SLO. Leeroy is well versed in navigating computers and software interfaces and has strong preference for solutions that leverage the power of computing to save valuable time. He drives a Tesla and is very passionate about manipulating large sets of data, drinking craft beer alone at home, and playing World of Warcraft.

Leeroy is currently working as a data scientist for the company LarkMogic, the premier company for analyzing enterprise data. As part of his workflow, Leeroy works with a variety of large data sets concerning various topics. Before Leeroy can begin analyzing his data sets for statistics, he must manually classify sensitive fields within the data sets he works with, to protect customer integrity, before he can perform further analysis on the remainder of the set. With the recent news of data scandals such as Facebook and Google+, data security has been a paramount issue in the business world. Given the varying nature of data sets, classifying anonymous data takes a significant amount of time. In fact, Leeroy estimates that up to 40 percent of his working day revolves around parsing through data sets; time that would be much better spent analyzing actual data.

Leeroy's ideal solution would be something that leverages computational power to perform redundant and tedious work. Automating Leeroy's data anonymization process would save a significant amount of time in his working day and enable him to enhance the efficiency of his work flow. Thankfully, at Leeroys company, they use MarkLogic for their enterprise NoSQL database solution. Since MarkLogics NoSQL database products now include a Data Classification Workbench, Leeroy can leverage this feature to automatically classify sensitive fields of data sets in record time.

2.3.5 Data Analyst

Author: Bennett

Henrietta Anderson is a female, aged 46, and a graduate of Purdue University with a Master's Degree in Mathematics. After graduation, Henrietta acquired a job at a newly founded financial data analytics firm, which sought to develop early software tools for describing and predicting market trends. Though she did not have much direct experience

with computers or programming, Henrietta had a strong background in theoretical computing, and so picked up the necessary skills relatively quickly, contributing heavily to the designs and underlying algorithms that drove the company's in-house software. She has interest and experience in designing mathematics-backed software solutions, but has limited skill in direct software development. She is married and has two children, both of whom are high school-aged. Her interests include DIY-decoration and furniture-making, as well as gardening. As such, she has a keen eye for the aesthetics of the products and companies she chooses to patronize, but keeps a look-out for function-oriented solutions.

Today, she works as a data engineer in the sales department of a large oil and petroleum distribution company. Her responsibilities include analyzing market trends and incorporating internal sales and production figures into predictive sales plans which aim to capitalize on developing opportunities and stay ahead of industry competitors. Her corporation undertook an effort to digitize their operations roughly 15 years prior, but due to their size and geographical spread, the firm's various departments and locations bid out their software projects to different companies, resulting in a number of disparate solutions and data silos which the company is still struggling to reconcile.

Henrietta's ideal solution would combine sharp aesthetic form with intuitive functionality, allowing for the flexibility she needs to integrate sales data and projections from multiple departments and locations into one concise and easy-to-access source into which she can integrate the company's existing sales-trend and projection tools. She values the ability to interface with a GUI, rather than bare data, but also wants to have access to a more unadulterated API which will allow her to directly pipe results into the company's analytics pipeline.

2.4 User Classes and Characteristics

User Class	Description
Data Scientist	Uses their analytical and technical capabilities to extract meaningful insights with the data.

2.5 Operating Environment

The software will operate on a modern web browser, such as Chrome, FireFox, Microsoft Edge, Safari, with an active internet connection on a computer. Since the product is based on a web application, any computing devices with an operating system capable of installing and running a modern web browser should be able to access this software. There is no hardware limitation other than meeting the minimum system specifications for running a browser on the computing device.

2.6 Design and Implementation Constraints

The data classifier component requires files of datasets to be uploaded to the system in order to have the machine-learning aspect process its meta-data. The size of the input files can be unpredictably large and possibly require a lot from the system processing capability so therefore, the development team will have to apply an upload size limit constraint on dataset files in order to maintain an acceptable performance quality level of the web application.

2.7 User Documentation

The development team will be using JIRA to maintain and track issues of the product's development. In addition, documents that will be created during the process of the project, such as the SRS document, prototypes, and software architecture diagrams, will be used as a purpose for documentation as well.

2.8 Assumptions and Dependencies

Most of the assumptions and dependencies for the project will be from changes to the requirement by the stakeholders. However, for the development of this project we are assuming several things.

- There is no guarantee of "correct" data formats. Our software will need to check for format validity.
- There will be no modification of the upload file. A metadata wrapper will be generated containing classification information.
- The classifier will ultimately interface with a MarkLogic DB. However, for the prototype, simple upload functionality for files of a few MB is appropriate

2.9 Business Rules

Business Rule ID	Business Rules Affecting the SRS
BR-1:	Ensure secure connection is established between user client and server
BR-2:	Cap data-set uploads to 1MB in size to reduce storage overhead
BR-3:	Verify request is originating from authenticated user

3 Use Cases

3.1 Use Case 1: View dataset

Use Case ID:	1
Use Case Name:	View dataset
Created By:	Sam Nayerman
Last Updated By:	Sam Nayerman
Date Created:	October 9, 2018
Date Last Updated:	October 9, 2018
Actors:	User
Description:	A User uses the GUI to view a dataset in a consistent format.
Preconditions:	1. User has already loaded at least one data set into the system.
Postconditions:	1. Data is displayed in a table, properly formatted.
Exceptions:	1. Dataset has corrupt values. 2a. File format error. 2b. Dataset too large to load at once.
Includes:	None
Priority:	High
Frequency of Use:	Potentially thousands of users with a varied amount of usages at the same time per day
Business Rules:	BR-1, BR-2, BR-3
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

3.2 Use Case 2: Label category as sensitive

Use Case ID:	2
Use Case Name:	Label category as sensitive
Created By:	Zachary Richardson
Last Updated By:	Zachary Richardson
Date Created:	October 9, 2018
Date Last Updated:	October 9, 2018
Actors:	User
Description:	A User uses the GUI to label a preexisting category as sensitive data.

Preconditions:	1. User has already loaded at least one data set into the system. 2. User has at least one non-sensitive category of data.
Postconditions:	1. Data category is labeled as sensitive for future classification and anonymization.
Normal Flow:	Mark a category as sensitive 1. User has previously uploaded at least one data set and views available data categories. 2. System presents user with existing data categories. 3. User selects a preexisting data category that is not currently marked as sensitive. 4. System presents user with an "Edit Category" view. 5. User marks category as sensitive. 6. System marks category as sensitive in database. 7. User exits "Edit Category" view and returns to view with existing data categories. 8. System presents user with existing data categories, indicating that the selected category is now marked as sensitive.
Alternative Flows:	Mark sensitive category as non-sensitive 1. Follow steps 1-4. 2. At steps 5-6, mark as non-sensitive rather than sensitive. 3. Follow steps 7+.
Exceptions:	Category already marked as sensitive 1. At step 5 category will already be marked sensitive. 2. User can then skip steps 5-6 and follow steps 7+.
Includes:	None
Priority:	Low
Frequency of Use:	High in classified/sensitive environments, low otherwise
Business Rules:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

3.3 Use Case 3: Uploading dataset

Use Case ID:	3
Use Case Name:	Uploading dataset
Created By:	Tony Chen
Last Updated By:	Tony Chen
Date Created:	October 9, 2018
Date Last Updated:	October 9, 2018
Actors:	User
Description:	A User accesses the Data Discovery Workbench from the web browser. The User views the Data Classifier menu for an upload button and presses it to select a dataset for the system to classify.
Preconditions:	1. User is using a modern internet browser. 2. User's computing device has an internet connection. 3. User's dataset file is in JSON or CSV format.
Postconditions:	1. Dataset is uploaded and classified in the backend. 2. Display the newly classified dataset.
Normal Flow:	1.0 Upload a single Dataset

	1. User selects the dataset upload button. 2. System displays a file selector window of available dataset files to upload from the User's computing device. 3. User selects one or more dataset files from the file selector window. 4. User indicates the dataset upload file selection is complete. 5. System validates the selected dataset upload file for the appropriate file format (if fails validation, back to step 1). 6. System displays the selected dataset file name, a progress bar, and a cancel button to abort the file upload in progress. 7. Patron confirms dataset upload or requests to upload a different file (back to step 3). 8. System displays an estimation of the dataset's upload progress. 9. System saves the dataset and then sends a signal for the next step in the process to classifying the dataset.
Alternative Flows:	1.1 Upload additional datasets with existing datasets already selected for upload 1. User asks to upload another file of datasets on top of the datasets that were already selected for upload during the current session. 2. Return to step 2.
Exceptions:	1.0.E.1 Invalid dataset file format (at step 5) 1. System informs User that the input dataset file format was invalid. 2a. User confirms the file format error. 2b. System terminates the file upload. 3a. Patron requests to select another file with an invalid file format. 3b. System restarts use case.
Includes:	None
Priority:	High
Frequency of Use:	Potentially thousands of users with a varied amount of usages at the same time per day
Business Rules:	BR-1, BR-2, BR-3

Special Requirements:	1. User shall be able to cancel the selected dataset file upload at any time prior to confirming the upload. 2. User shall be able to view all the dataset file uploads in the current session.
Assumptions:	None
Notes and Issues:	None

3.4 Use Case 4: View categories classified as anonymous

Use Case ID:	4
Use Case Name:	View categories classified as anonymous
Created By:	Kyler Ramsey
Last Updated By:	Kyler Ramsey
Date Created:	October 9, 2018
Date Last Updated:	October 9, 2018
Actors:	User
Description:	A User has already uploaded a data set and wants to see what categories were classified as anonymous.
Preconditions:	1. User is authenticated with MarkLogic's interface. 2. User has uploaded the data set into the Data classification system.
Postconditions:	1. User is able to see categories marked as anonymous 2. User is able to select the option to mark new category as anonymous (separate use case) 3. A record of classified categories for the data set is saved for the user.
Normal Flow:	View anonymous categories (min 1)

	1. After completion of "upload data set" use case, the application will display an "Anonymous Data" tab 2. User selects which data set they want to view anonymous categories 3. User selects "Anonymous Data" tab for the given data set 4. Application displays a table of categories that have been classified as anonymous 5. User can download a record of these anonymous categories as a .txt file 6. User has the ability to deselect any categories mistakenly classified as anonymous be selecting a check box next to the category
Alternative Flows:	1.1 View "Anonymous Categories" with no classified data (after step 2) 1. Application displays a message stating that no categories have been identified as sensitive/anonymous 2. User can begin use case "Label category as anonymous" if they want to classify categories as anonymous 3. Return to end of step 3.
Exceptions:	1.0.E.1 No categories marked as anonymous 1. User is presented with use case option to mark categories as anonymous 2a. User proceeds to mark categories 2b. Or user decides to not mark any categories 3a. Return to step 3
Includes:	None
Priority:	High
Frequency of Use:	Potentially thousands of users with a varied amount of usages at the same time per day
Business Rules:	BR-1, BR-2, BR-3
Special Requirements:	1. User shall be able to view the identifies of anonymous categories if they decide to
Assumptions:	None
Notes and Issues:	None

3.5 Use Case 5: Add Category from Data Classifier GUI

Use Case ID:	5
Use Case Name:	Add Category from Data Classifier GUI
Created By:	Bennett Robinson
Last Updated By:	Bennett Robinson
Date Created:	October 10, 2018
Date Last Updated:	October 10, 2018
Actors:	User
Description:	A user has uploaded a data-set and run it through the classifier, but wants to add a category for classification.
Preconditions:	1. User is authenticated with MarkLogic's interface. 2. User has uploaded the data set into the Data classification system. 3. User has run the classifier on the data-set.
Postconditions:	1. User is able to see a GUI element inviting them to add a category for classification. 2. User is able to select the option and add a custom category. 3. The category is enumerated among the previously identified categories. 4. The user-entered category is identifiable as having been added by the user.
Normal Flow:	Add Category

	1. After completion of "classify data set" use case, the application will display the identified classifications, as well as an option to add or modify classifications 2. User selects 'add or modify classifications' 3. User inputs a label for an additional category/classification 4. Application displays a button to confirm the addition of the classification(s) 5. User selects 'confirm changes' to save their addition(s) 6. Application returns to the 'display identified classifications' view 7. Application displays the user-added category/categories alongside existing classes, and highlights newly added classes which have not been run through the classifier 8. Application displays an option to re-run the classifier with the user's additions included
Alternative Flows:	1.1 User selects 'Confirm Changes' with no pending changes (after step 5) 1. Application displays a message stating that no changes have been made to existing categories 2. User can input desired changes, or continue without any changes 3. Return to end of step 5.
Exceptions:	1.0.E.1 User continues without re-executing classifier upon addition of new categories 1. System displays warning that new classes will not be incorporated into report 2. User can either re-execute classification, or continue without displaying their additions 3. Return to end of Use Case 1.0.E.2 User adds category that already exists (after step 4) 1. Application displays a message that this categorization already exists 2. User exits message dialog 3. Return to Use Case at step 3.
Includes:	None
Priority:	High

Frequency of Use:	High degree of use by many different users, with declining use for return users as they settle on their various data-sets and sources.
Business Rules:	BR-1, BR-2, BR-3
Special Requirements:	1. User shall be able to add-to or modify existing classifications from an inputted data-set
Assumptions:	None
Notes and Issues:	None

3.6 Use Case 6: Generating JSON from Classification

Use Case ID:	6
Use Case Name:	Generate JSON from Classification
Created By:	Kyler Ramsey, Zach Richardson, Tony Chen, Bennett Robinson
Last Updated By:	Kyler Ramsey, Zach Richardson
Date Created:	November 1, 2018
Date Last Updated:	November 1, 2018
Actors:	User
Description:	A user has uploaded a data-set and run it through the classifier, and wants to download generated metadata as a JSON file.
Preconditions:	1. User has run data classifier and generated associated metadata. 2. User has available storage space on local machine.
Postconditions:	1. User is able to download a classification file in JSON format, containing the metadata necessary for classification.
Normal Flow:	Download JSON 1. After completion of "classify data set" use case, the application will display the identified classifications. 2. User selects 'Download JSON'. 3. The system generates the JSON file to download. 4. User downloads generated JSON file.
Exceptions:	1.0.E.1 System is unable to generate a JSON file with classification ontology for the dataset

Includes:	None
Priority:	High
Frequency of Use:	Varies depending on user profile.
Business Rules:	BR-1, BR-2, BR-3
Special Requirements:	1. User must have previously generated metadata using classifier.
Assumptions:	None
Notes and Issues:	None

4 System Features

4.1 Data Classifier

The Data Classifier will - among other things - allow users to add and modify categories elicited from the input datasets.

4.1.1 Description and Priority

The Data Classifier is the highest priority module of the system, and will offer a set of functionalities related to uploading and classifying data sets according to a learning-algorithm model. It will allow the user to upload datasets to the system and specify pre-existing structures or categories. After classification, the system will display the data in a more visually accessible format, allowing the user to add-to, modify, or remove categorizations elicited from the classification stage.

4.1.2 Functional Requirements

Requirement	Description
FR-1	Data Classifier shall allow users to upload dataset files.
FR-2	Data Classifier shall display the classified data set to the users.
FR-3	Data Classifier shall categorize data using a machine learning algorithm.
FR-4	Data Classifier shall allow categories to be explicitly classified as anonymous.
FR-5	Data Classifier shall allow users to modify and add to categories elicited by the Data Classifier.
FR-6	Data Classifier shall allow users to generate and export a machine readable description of the classified data.

5 External Interface Requirements

5.1 User Interfaces

UI Requirement	Description
UI-1	Data Classifier shall launch to a login page to allow for user authentication.
UI-2	Data Classifier shall display an indicator that the machine learning algorithm is in process to the user after uploading.
UI-3	Data Classifier shall display metadata learned from the uploaded dataset.
UI-4	Data Classifier shall allow the user to edit certain metadata elements.
UI-5	Data Classifier shall follow MarkLogic design guidelines and recommendations, if any.

5.2 Software Interfaces

SI Requirement	Description
CI-1	Data Classifier shall be runnable on MacOS, Windows, and Linux based operating systems.
CI-2	Data Classifier shall use React to display user-facing components.
CI-3	Data Classifier shall use Python with the PyTorch library for the machine learning components.
CI-4	Data Classifier shall be compatible with CSV and JSON datasets.

5.3 Communications Interfaces

CI Requirement	Description
CI-1	Data Classifier shall allow uploads of up to 1MB in size.
CI-2	Data Classifier shall verify a user's authentication before allowing user to upload, view, or edit datasets.

6 Other Nonfunctional Requirements

6.1 Performance

NFR-1A	The interface will not slow down when processing data.
NFR-1B	The system will process the uploaded dataset files within a reasonable timing.

6.2 Safety

NFR-2A	There can be sensitive information in a dataset input that will be processed and stored in the system. The system will tag and indicate to the user of what information is considered sensitive.

6.3 Security

NFR-3A	The Data Classifier will process and validate file inputs going into the system.

6.4 Software Quality Attributes

NFR-4A	The Data Classifier application will run on OSX and Linux based operating systems.
NFR-4B	The Data Classifier shall have a unified 'look-and-feel' aesthetic quality across its various GUI components.

7 Other Requirements

None.

A Glossary

BR	Business Rule
FR	Functional Requirement
GUI	Graphical User Interface
NFR	Non-Functional Requirement

B Analysis Models

TBD

C Issues List

There are no known issues at the moment.