Data Classifier:
Software Architecture
Version 1.0

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

November 7, 2018

Contents

Credits

Name	Date	Role	Version
Matt Yamolich Spencer Schurk	November 7, 2018 November 7, 2018	Contributing Member Lead Author of Introduction	1.0 1.0
Sally Jones	October 10, 2009	Lead Author of Business Requirements	1.0

Revision History

Name	Date	Reason for Changes	Version
Spencer Schurk	November 7, 2018	Completed Introduction	1.0

1 Introduction

The purpose of this document is to describe the architecture of our Data Classifier. This document will be used as a reference when development begins. This document may change over time as we learn more about specific technologies being used for this project. For more information regarding this project, please see the Vision and Scope document, as well as the SRS document.

2 Problem Description

[Tell the audience what problem is being solved, or requirements are being met, by this design. Include requirements or constraints that drive the direction of the design.

Any quality attribute requirements or constraints should also be explained in this section.]

3 Solution

This project is meant as an interface between a database and the data being inputted to it by the user. The main goal of this project is to provide analytics for the user in order to better identify what type of data is being inputted. With this project, the user will be able to tell the categories of data being inputted, and how they potentially relate to each other. The classifier will allow the user to modify these relations and to interact with a visual model based on these interactions. As defined by the customer, the user will interact with a web based GUI, most likely based on React. This GUI will interface with the Python backend, which is another design restraint, and utilize Scikit, a python library, as our chosen machine classifier. This classifier will act as the main interface/buffer between the user and actually logging the data to the database.

3.1 Overview

[This section should be a summary of the solution in a few paragraphs, preferably less than a page.]

3.2 Components

[This section should list and briefly describe the components of the design. The word "components" can refer to deployable units or to areas of the design, as appropriate. This section should start with the deployment diagram.]

3.3 Design

[This is where the meat of the design lives. For each component, there should be a subsection describing the design of that component in as much detail as you want to provide in a high-level design. There should also be a subsection that describes how the components work together. This section should include class, sequence, and collaboration diagrams as appropriate.]

3.3.1 Classification Structure

Figure 1: svg image

4 Test

[This section should describe how the solution will be tested, especially if the test scenario or test infrastructure are complex.]

5 Issues

[Any issues or open questions should be described here.]

A Glossary

Define all the terms necessary to properly interpret the software architecture, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each softwar architecture.

B Issues List

This is a dynamic list of the open architecture issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.