Oregon State University Computer Science Senior Design 2016

Requirements Document

By Alex Hoffer, Jake Smith, and Chen Chongxian

Team Name: Stat Champs

Abstract

The Biochemistry and Biophysics department at Oregon State University will have a server that allows for students to select statistical categories from college basketball games and generate March Madness brackets from these selections.

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this software requirements specification is to outline what is necessary for this software product to include. That is, this document will describe what inputs the software will expect and utilize and what ways it will transform these inputs into meaningful outputs. The intended audience for this document is Dr. Victor Hsu of the Oregon State Biochemistry and Biophysics department.

1.2 Scope

The Biochemistry and Biophysics department at Oregon State University needs to train their students on basic machine learning concepts. This is because machine learning is a highly useful tool in Biochemistry and Biophysics. For example, finding and understanding biologically functional DNA sequences is made possible by machine learning. However, machine learning is not a cornerstone of these students' education. There needs to be an instructional tool that provides these students with the opportunity to understand basic machine learning principles. Grasping machine learning concepts from its applications to biochemistry/biophysics is difficult because these models are generally hard to interpret. Therefore, this tool should utilize results that are simple to interpret in order for students to understand how training data on certain statistics can either damage or improve the accuracy of their generated model. The use of March Madness brackets makes this possible because the output of the students' effort will be straightforward: a team can either win or lose in each round, and whichever basketball statistics the user chose to train their data on will be reflected in how far each team goes in their model.

1.3 Definitions, acronyms, and abbreviations

NCAA: National Collegiate Athletic Association, the organization that holds the March Madness tournament each year.
Scikit: A machine learning API supported by Python that provides us with the algorithms we will allow users to select from to use to generate brackets. The user guide for this API can be found in the References section at (1).
Webix: A graphical user interface API supported by JavaScript that provides us with the ability to present instructions and the generated brackets in a way that is appealing to the user. The user guide for this API can be found in the References section at (2).

1.4 References

[1] "User guide: Contents - scikit-learn 0.18 documentation," in *Scikit-Learn*, 2010. [Online].
Available: `http://scikit-learn.org/stable/user_guide.html`. Accessed: Nov. 8, 2016.
[2] X.S. Ltd, "Guides Webix Docs" [Online].
Available: `http://docs.webix.com/desktop_basic_task.html`. Accessed: Nov. 8, 2016.

1.5 Overview

The remaining contents of this document will consist of more specific descriptions of the software itself such as the product perspective, the product functions, user characteristics, constraints, assumptions/dependencies, as well as technical challenges and functional requirements. There will also be a Gantt chart to outline our project's lifespan.

2 Overall Description

2.1 Product Perspective

This product is independent and completely self-contained. As of right now, there are no plans to implement a larger product which this product must interact with. There are various APIs the product must interact with in order to function, and these are outlined in other sections of this document.

2.2 Product Functions

Without logging into a specially created account, the user will be able to access a web page where they can select a) basketball statistics and b) a machine learning algorithm. They will then be presented with a March Madness bracket that reflects the machine learning algorithm they chose that has been trained on the basketball statistics they chose. The user can run this program as many times as they like to generate more brackets.

2.3 User Characteristics

We do not place any restrictions on who can access this web page. Anyone with the link can utilize our service. However, this product is designed specifically for undergraduate or graduate students of Biochemistry and Biophysics at Oregon State University. It will be designed with this in mind by allowing for them to select machine learning algorithms which may be useful when applied to Biochemistry and Biophysics.

2.4 Constraints

This product will be using several libraries and languages. Thus, we must be aware of the limitations of these technologies. We believe that webix will be sufficient for developing a user-friendly experience, and scikit will be sufficient for implementing the machine learning algorithms we allow the user to select from. However, there are considerations, such as how scikit will communicate its results to JavaScript, that we must develop solutions for before the module can be fully functional. A more in-depth exploration of these technologies and the constraints they may impose can be found in our technology review.

2.5 Assumptions and dependencies

The browser that the user is viewing the web page with must have JavaScript enabled.

2.6 Technical Challenges/Issues

We would like this instructional tool to be hosted online. This means that we must consider the different browsers students may use. We want our tool to be compatible with Internet Explorer, Google Chrome, and Mozilla Firefox. We must be careful in developing our GUI so that it appears the same in each of these browsers. To ensure the appearance of the module will be identical between these three options, our GUI will be written in the JavaScript library webix, which provides support for these three browsers. Another technical challenge we must solve is allowing users to select from a myriad of statistical categories and transferring this input to our machine learning submodule. The statistical categories we will be including are individual player statistics such as points, rebounds, and assists per game, as well as team statistics, such as record against their opponent and points, rebounds, and assists per game. These statistics will be held in a .csv file and will be accessed using Python. We also must be sure that our machine learning submodule will function properly in each of these three browsers. That is, users should be able to select from which machine learning algorithm they want to use after selecting their statistical categories and the output of such a decision should accurately reflect both of these factors. We will include a number of machine learning algorithms for the user to choose from, such as generalized linear models, linear and quadratic discriminant analyses, and support vector machines. The description for how these algorithms are implemented can be found in the scikit documentation [1]. In order to achieve this, we will use the Python scikit library, which is stable for our purposes.

3   Specific requirements

3.1   Functional Requirements

1. Users start out by seeing instructions on how to use the tool presented to them using the JavaScript library webix.
2. Users then see a compilation of basketball statistics consisting of the categories such as points, assists, and rebounds per game (for each player and team) and select which statistics they want to train their model on.
3. The user is asked which machine learning algorithm out of a set including supervised and unsupervised learning algorithms we provide that they want to use.
4. The statistics are passed to the corresponding .machine learning algorithm which is computed using scikit. 5. The resulting bracket is generated and presented to the user.
5. The user is given an option to create a new bracket.

3.2   Technical Requirements

1. Cross-browser support (IE, Firefox, Chrome) for both GUI (using webix) and machine learning submodule (using scikit).
2. API with acceptable level of documentation

3.3   Usability Requirements

1. The system will look and act the same in all major browsers (outlined above).