MACHINE LEARN YOUR WAY TO MARCH MADNESS GLORY!

Teaching Biochemists and Biophysicists Machine Learning

Introduction/Background

IMPORTANCE OF MACHINE LEARNING TO BIOCHEMISTRY AND BIOPHYSICS

Biochemistry and biophysics are two fields that are ripe with many exciting breakthroughs. Machine learning, a type of artificial intelligence where computer programs adapt to new data, is used by biochemists and biophysicists to do things like analyze genomic DNA sequences. Our client, a professor in the department of Biochemistry and Biophysics at OSU, recognized there was a need for his budding scientists to understand machine learning so they could be better prepared for their careers.

NEED FOR A MACHINE LEARNING INSTRUCTIONAL TOOL

Our client noticed that the Biochemistry and Biophysics curriculum at OSU did not encourage undergraduate students to learn machine learning. Even if machine learning classes were to become a cornerstone of their coursework, the content would be difficult for people without a Computer Science background. To make matters worse, teaching machine learning to these students through its application to biochemistry is particularly challenging, since biochemical results are non-definitive in that DNA sequences often do not need to be exact and can be unclear. Meanwhile, college basketball results are win-lose and therefore it is more straightforward to interpret the differences in results based on changing the inputs.

WHAT WE WERE COMMISSIONED TO DO

We were enlisted to produce an online instructional module where these students could grasp machine learning fundamentals in a clear manner. Our client wanted us to develop this module so that students could generate machine learned NCAA March Madness brackets. Since a fundamental aspect of learning machine learning is recognizing how the inclusion or exclusion of data influences resulting models, this module would satisfy the need by producing models (brackets) that were distinguishable from each other based on the college basketball statistics a user chose for training.

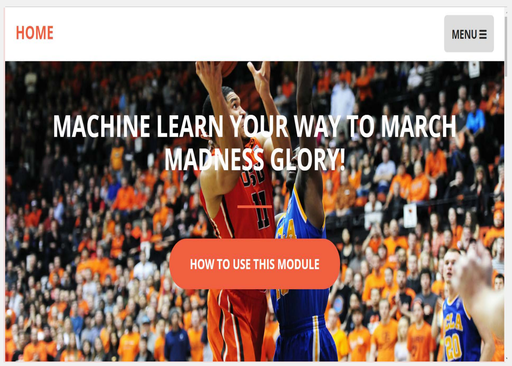

Fig. 1: Home page of website that includes project information and a link to our module.

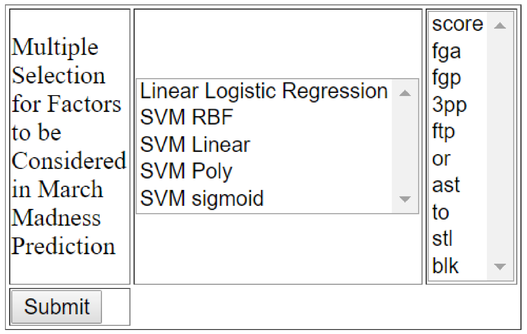

Fig. 2: Menu where the user chooses a machine learning algorithm and stats to generate a bracket with.

PROJECT INFORMATION

Class: CS Senior Capstone, 2016-2017

Developers:

- **Alex Hoffer** (hoffera@oregonstate.edu)
- **Jacob Smith** (smitjaco@oregonstate.edu)
- **Chongxian Chen** (chencho@oregonstate.edu)

Client: Dr. Victor Hsu, Oregon State University, Department of Biochemistry and Biophysics

PROJECT DESCRIPTION

To implement the module, we needed to complete the following five steps:

1. Develop a Graphical User Interface (GUI)
2. Aggregate/select college basketball statistics
3. Feed statistics to machine learner
4. Train a model using an algorithm of the user's choosing
5. Generate March Madness bracket that represents the model

The following headings are technical descriptions of the five steps:

1. GUI

Alex used HTML, CSS, and JavaScript to produce the GUI for our web page. HTML was used to split the page into logical sections such as Home (found in *Fig. 1*), Instructions, Module, Purpose, and About. We utilized CSS to make these sections look clean and usable. Finally, JavaScript was used to enhance the user experience by making the page interactive, such as turning certain buttons different colors upon clicking in order to notify the user of the action they had just performed.

2. AGGREGATE/SELECT STATISTICS

Jacob gathered college basketball statistics from 1985 to the current season from the website Kaggle.com in the CSV file format. Since the regular season didn't conclude until March, Jacob manually updated the database to reflect the current standings frequently until the final game was played. Then, he added stats from the tournament for future use in algorithms and analysis. We used a Python script to allow users to choose from a wide variety of stats including categories like field goals attempted per game to train a model on, as demonstrated by *Fig. 2*.

3. FEED STATISTICS TO MACHINE LEARNER

Using the Python SciKit-Learn library, Chongxian read the CSV files of the user selected statistics into Numpy arrays.

4. TRAIN A MODEL USING AN ALGORITHM

Along with their choice of statistics, users are also able to choose between different machine learning estimators such as Linear Regression and SVM Polynomial. By using a basketball ELO rating system, the supervised machine learning model is able to fit on the statistics and predict new matches. A CSV file of the match results between two teams with the probability is generated as a result. The bracket results effectively present how the users choice affects the machine learning prediction.

5. GENERATE BRACKET OF RESULTS

While a machine learned module is being generated, the user is presented with a screen that includes the command line arguments given to SciKit and informs the user on which steps are necessary to complete their request. The prediction CSV file generated from the machine learning model was then transferred into bracket form by Jake using a Python script.

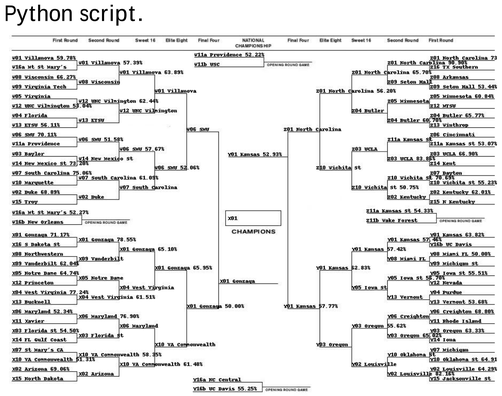

Fig. 3: A March Madness bracket predicted by the SVM RBF algorithm.

CONCLUSION

Alex Hoffer, Jacob Smith, Chongxian Chen

FEATURES PROVIDED BY THE MODULE:

- A Graphical User Interface (GUI) including a "Home" page, "Instructions" page, "About The Developers" page, and a "Purpose" page.
- The ability to select from a set of college basketball statistics.
- The ability to select from a set of popular machine learning algorithms.
- A machine learned bracket that corresponds to the specific statistics and algorithm the user requested the model to be trained on.

The project was completed in early April. All functional requirements as outlined by our client were completed. Future improvements to our module could include more machine learning algorithms, a wider variety of statistical categories, a more elegant looking outputted bracket, and finding a way to increase the speed at which the machine learning algorithms generate results. Additionally, the developers of such modules may wish to have prior machine learning experience, more proper modes of communication, and a more specific work schedule established before development in order to allow for time after completion to polish each component of the module separately.

Oregon State UNIVERSITY