# Technology Review for Machine Learning for March Madness

Alex Hoffer, Chongxian Chen, and Jacob Smith
Team Name: Stat Champs

**Abstract**

Students of Biochemistry and Biophysics at Oregon State University need to have the opportunity to learn machine learning algorithms that will be useful in their careers. However, grasping the nuances of these algorithms is difficult when they are taught through their application to Biochemistry and Biophysics. One way that may facilitate this learning process is introducing these concepts through their application to something fun and simple, like college basketball statistics. This project will develop an online module where these students can select a machine learning algorithm, choose which college basketball statistics they want to train this algorithm with, and generate a NCAA March Madness bracket.

TABLE OF CONTENTS

## I. Introduction

THERE are many important technological considerations when developing this instructional tool. The page must be reasonably fast, responsive, and usable, it must provide statistics for the user to choose from, and it must provide these statistics as input to machine learning algorithms. Finally, the bracket must be generated and presented to the user. Each one of these necessary components has several potential technologies that could be used satisfactorily. It is the purpose of this document to identify the possible technologies and make reasoned decisions as to what technologies we will be using. The first three possible technologies were written by Alex Hoffer. These first three technologies are chiefly concerned with the usability of the service, specifically the design of the graphical user interface, the instructions that will be presented to the user, and the presentation of the machine learning bracket that is generated. In technologies 4 and 5 Jake Smith will talk about how we will obtain and store the sports statistics for easy use by our machine learning algorithms. It is our belief that in order for the tool to be effective in educating students it must be well designed and easy to understand. Chongxian Chen will be responsible for designing and implementing machine learning algorithm. Technologies 6, 7 and 8 will be discussing technology involved in designing machine learning algorithm, namely algorithm library, statistics model and cloud server for hosting the project

Stat Champs
November 14, 2016

### A. Technology 1: Graphical user interface of web page

*1) Options 1, 2, and 3:* Kivy, PyQt, PyGUI. All three options are libraries to be used with Python.

*2) Goals for use in design:* The page must be visually pleasing, interactive, and usable. This tool will be used to demonstrate machine learning processes to programming neophytes so the logic and design of the module has to be easy to understand to ensure that the user will not be confused.

*3) Criteria being evaluated:* Since there are not major concerns with security for this module because of the lack of sensitive information being used, security is not an important criterion. Cost is also not a very crucial factor because there are plenty of free technologies available that can generate visually pleasing and usable web pages. Availability is an important factor because we want the module to work the same on all major browsers, specifically Chrome, Firefox, and Internet Explorer. Speed is also a valuable consideration because we want the machine learning component of this service to be quick. Therefore, the technology that we use to design the web page itself should be reasonably economical so the page loads quickly and doesnt absorb too many resources that should be allocated for the machine learning processes. Readability is also important- we would like the technology to be easy to understand. Of similar importance as speed is economy. We want this technology to be able to do a lot with only a few lines of code.

*4) Table comparing options:*

| Technology | Economy | Readability | Availability | Speed | Notes |
|---|---|---|---|---|---|
| Kivy | Extremely | Extremely | Cross-platform | Very fast, built on OpenGL ES 2 | Focus is on complex UIs |
| PyQt | Extremely | Somewhat | Cross-platform | Fast, written in C++ | Focus is on mobile development |
| PyGUI | Extremely | Extremely | Cross-platform | Not conclusive | Smallest and simplest |

*5) Discussion:* Each of these three technologies are viable options. Since they are all Python libraries, they should be easy to learn, which is important because the GUI is the first thing to be developed in our project and the other requirements build from it. Each of the three are powerful enough for our purposes, and each have different strengths. Kivy is sophisticated and elegant, but it might be too vast for our uses. Kivy is used in complex operations like touch screen animation [1], and our project doesn't call for an extremely sophisticated GUI. In fact, our GUI should be simple and unobtrusive so as not to draw attention away from the machine learning components. PyQt's strength appears to be in mobile development [2]. Since our module doesn't have to be usable from a mobile device, it may be best to go with a technology with a focus in computer based web development. This brings us to PyGUI, which is the best of the three options for our purposes. PyGUI is the oldest of the three and was developed to be simple to implement [2]. Its focus matches with ours: we need a simple GUI, and PyGUI is the least flashy and most reasonable of these technologies.

*6) Selection of best option:* PyGUI is the technology that seems to most closely capture the needs of our GUI without adding fancy features that may slow our module down and make maintenance and testing difficult.

### B. Technology 2: Instructions for the user to interact with the module before the module itself is presented

*1) Options 1, 2, and 3:* Webix, UKI, MochaUI. All three options are freely available JavaScript libraries. Use of JavaScript is important here because we want the instructions to seamlessly lead into the module itself without requiring the user to refresh the page.

*2) Goals for use in design:* The module must have instructions for the user to follow so they can effectively use the tool. These instructions should be easy to read and flow seamlessly into the tool. The machine learning component of this project should be the central focus, and so the instructions that we provide to the user have to be designed in such a way that they are not visually offensive or disorganized.

*3) Criteria being evaluated:* Security is again not particularly important due to the innocuousness of the data being transferred from user to server. Cost is not concerning because of the wide availability of open source technologies that can properly satisfy this requirement. Speed is important because we want our server's processing resources to be saved for the machine learning algorithms and we want the page to be quick. Availability is important because we want this tool to be usable on each of the major browsers. Readability and economy are also important.

*4) Table comparing options:*

| Technology | Economy | Readability | Availability | Speed | Notes |
|---|---|---|---|---|---|
| Webix | Extremely | Very | Cross-platform | Not conclusive, no metrics | Easy to learn, 128KB |
| UKI | Very | Somewhat | Cross-platform | Fast (progressive rendering) | 34 KB, meant for desktop web apps |
| Angular JS | Very | Below average | Cross-platform | Somewhat fast | Easy to start, hard to maintain |

*5) Discussion:* Webix is a popular and reliable option. What makes it so attractive is how feature rich it is, its ease of use, and its readability. The example program listed on [3] demonstrates how economical it is, too. With only several lines of code, Webix is capable of generating a clean and usable user interface. One drawback of Webix is that it's difficult to tell how fast it is. Its documentation merely makes note that is fast, but provides no usable metrics. It also isn't as lightweight as UKI (128 KB for Webix [3] and 34 KB for UKI [4]), but it is still quite small, so its size won't be slowing down the server. UKI is also a good option and its focus appears to directly parallel ours. It is meant for desktop applications [4], which is the medium of our project. It is also quite fast due to its progressive rendering and can produce 30,000 tables almost instantaneously [4]. Angular JS seems to be the worst of these three options, as it is hard to learn [5] and its programs grow rapidly in complexity. It may be too ambitious for the requirements of our project to learn Angular JS.

*6) Selection of best option:* Webix is the best option because it is easy to use, reliable, and offers the features that we need without slowing down the server. UKI is another good option but is harder to use. Angular JS is not necessary for our project.

### *C. Technology 3: Presentation of the machine learned bracket that is generated by scikit*

*1) Options 1, 2, and 3:* TkInter, VTK, Wax. All three options are freely available Python libraries.

*2) Goals for use in design:* The module is functionally useless without the presentation of the bracket that was generated by the machine learning algorithm. This technology must present the bracket in a way that is visually pleasing and easy to understand.

*3) Criteria being evaluated:* Cost and security are again not important for the reasons outlined earlier. Availability is crucial because we want the bracket to appear the same in all major browsers. Speed is important too because we want the user to see their bracket promptly. Readability and economy are also useful considerations.

*4) Table comparing options:*

| Technology | Economy | Readability | Availability | Speed | Notes |
|---|---|---|---|---|---|
| TkInter | Extremely | Somewhat | Cross-platform | Slow | Most popular Python GUI library |
| VTK | Extremely | Extremely | Cross-platform | Fast, written in C++ | Focus is on data display |
| Wax | Very | Somewhat | Cross-platform | Somewhat fast | Good for quick development of small project |

*5) Discussion:* TkInter is the standard for Python GUIs. However, it isn't very fast [6] and offers many features that aren't necessary for our needs. It also isn't particularly easy to read, which has implications for both testing and maintenance. We need a framework that can accept data generated by a machine learning algorithm and then present the data in bracket form. This is, in essence, data visualization. This makes VTK a compelling option. VTK is economical (you can do a lot with a small amount of code), easy to read, very fast, and was developed with data visualization in mind [7]. Wax is the least compelling option. It is difficult to read, only somewhat fast, and appears to be in many ways flawed. [8] suggests that Wax is a good option if you want to quickly hack something together, but the presentation of the bracket is perhaps the most important part of this project, and as such, it should not be hacked together. We require an elegant solution to this problem and VTK appears to be the tool that will be most poised to fit our needs.

*6) Selection of best option:* VTK is the option that is most likely to give us the results we want. It is designed with data visualization in mind, which is what we need it for. The other two frameworks are flawed.

### *D. Technology 4: Where the data will come from*

*1) Options 1, 2, and 3:* Ncaa.com, sports-reference.com, and kenpom.com

*2) Goals for use in design:* The Goal is to determine where to get the statistics from that can provide an easy way to take the data off the website as well as have all the necessary statistical categories were looking for.

*3) Criteria being evaluated:* Cost is a factor because there are websites such as kenpom that offer advanced statistics for mens ncaa basketball but you must pay to access the data. Websites such as ncaa.com and sports-reference are free to use and browse the statistics. The amount of stats available is important because the more data points you have to work with the better the machine learning algorithms will do. Kenpom has an advantage in this area, it offers more statistical categories than both ncaa and sports-reference.com. Some websites even offer statistics of where each player on the court was at all times during the game. The ease of use is another important factor both kenpom and sports-reference provide downloadable csv files that can easily be dealt with to put in a database while ncaa.com does not. Correctness of the data is important because the data must be accurate for the machine learning algorithms to do their best. Statistics for both ncaa and sports-reference are backed up by the ncaa or a third party like espn. Kenpom and cites you must pay for your data there is no certainty that the advanced data will be accurate.

*4) Table comparing options:*

| Website | Cost | Amount | Ease of use | Correctness | Sp |
|---|---|---|---|---|---|
| NCAA.com | Free | All normal statistics | Need to write script to get data | Confirmed | S |
| Sports-reference.com | Free | All normal statistics | Downloadable csv file | Confirmed | F |
| Kenpom.com | Costs money | All normal statistics plus more advanced | Downloadable csv file | Not validated | f |

*5) Discussion:* Sports-reference is a great option because it has most of the statistics we will need and it is easy to download and use in the csv files unlike the other free option ncaa.com which has roughly the same statistics but without the easy to use downloadable csv. For ncaa.com we would need to write a piece of code that scrapes statistics that we need from the website to gather the data. But if we did go that route we could scrape all the sites (ex. Espn, yahoo) and compare them against each other to see if they all are correct. Or we could pay for the data using a site like kenpom.com and possibly get data points we would not get otherwise using the two free options should as player movement and touches.

*6) Selection of best option:* Sports-reference.com is the best option because its free, easy to deal with and has enough data to go on. More data can always be scraped off other free websites. It also provides a deep history of the teams statistics so we could incorporate those data points as well.

### *E. Technology 5: Storage of the data*

*1) Options 1, 2, and 3:* Excel, Python SQLite, and Amazon Aurora

*2) Goals for use in design:* Goal is to figure out the best database option to use that will be the easiest to integrate with the website and machine learning algorithms.

*3) Criteria being evaluated:* Cost isnt a factor is this one because they are all free or offered through OSU. The availability and security of the system arent a factor because they are all available and the data in the database is not sensitive so it does not need to be encrypted or protected in any extra fashion. Ease of use is a factor because the database needs to be able to integrate with the website nicely as well as be able to add and delete data easily. Speed wont be a factor because they all offer around the same speeds.

*4) Table comparing options:*

| Technology | Cost | Ease of use | Security | Speed |
|---|---|---|---|---|
| Excel | Free | Easy | Protected | Fast |
| Python SQLite | Free | Easy | Protected | Fast |
| Amazon Aurora | Free | Easy | Protected | fast |

*5) Discussion:* They are all viable options to use for this project each do about the same thing except excel and Amazon Aurora would be external from the machine learning python code. Both excel and Amazon Aurora are both nice to use because you can search on your dataset easily. They are all very easy to interact with via python code but Amazon Aurora has the big bonus of having a nice user interface for the database as well as the ability to add data very easily via csv files which are the form most of the statistics would come in. The big downside of SQLite, a database included with Python, is it creates a single file for all data per database. Other databases such as MySQL, Amazon Aurora, and Oracle and Microsoft SQL Server have more complicated persistence schemes while offering additional advanced features that are useful for web application data storage.

*6) Selection of best option:* Given that Amazon Aurora has the user interface and it is very csv file friendly I am going to have to choose it for our database storage.

### *F. Technology 6: Machine Learning Libraries/Platforms to Train the Model*

*1) Options 1, 2, and 3:* SciKit, Amazon Machine Learning(AML), Pylearn2.

*2) Goals for use in design:* Our Algorithm should be as accurate as possible. It should be highly scalable considering we may have a lot of basketball match data. We also want full control of our algorithm, i.e. have access to the source code if we are using some library in case we may need to modify them for our project.

*3) Criteria being evaluated:* The cost is definitely one important aspect to be considered in our project. We want to make our model available to most people so they can learn that machine learning is a power tool to use in many areas. Providing access to as many people as possible at a low rate or free is our goal. The availability is also considered. Some service may only be available in some countries. We should try to avoid those that are only accessible in a small area. Speed is also an important aspect of our algorithm. Making our algorithm efficient really enhance the user experience a lot. Finally security is something to be considered in the process but not too important since we dont have private user information. But keep in mind that a major flow in our project could cause danger to the system.

*4) Table comparing options:*

| Technology | Economy | Readability | Availability | Speed | Notes |
|---|---|---|---|---|---|
| SciKit | Extremely | very | Open Source | Very | The most popular ML library |
| AML | very | very | Open Source | Normal | Easy to start with but hard to modify |
| Pylearn2 | Extremely | very | Open Source | Very | Popular Library but no longer actively developing |

*5) Discussion:* All three options are very reliable. Amazon Machine Learning(AML) is a fast developing platform that provides easy Machine Learning model to many customers with different background to start with[9]. The advantage of Amazon Machine Learning model is that it has great GUI that makes the process easier. And according to its introduction, AML is very closed to the purpose of our model. With data training the model, it will give a prediction based on the data. Users can also narrow down the searching areas. But because AML is not open sourced I will first dismiss this option. Our project is likely to modify a lot of algorithm in order to better meet the potentially changing needs. SciKit is currently the most popular open source library on Python and has active developers developing them and fixing bugs[10]. While Pylearn2 is also a famous and a widely used library on Python, there is no developers responsible for developing them. Although Pylearn2 claims that they will continue reviewing pull request on Github, there is no active developer for the project[11].

*6) Selection of best option:* SciLearn is the best option for the need of our project. It is open-sourced, reliable and has active developers developing it. It is also widely popular which means we will be more likely to get community support.

### *G. Technology 7:Server for computational power and hosting our database*

*1) Options 1, 2, and 3:* Amazon Web Service(AWS), Google Cloud Platform(GCP), Oregon State University Student Engineering Server(OSU Server).

*2) Goals for use in design:* Our project is likely to require strong computational power because our prediction model needs to evaluate a large amount of data. And if we want to access our model from different devices and different locations, it is important that we train our model on the cloud. A reliable database server on the cloud is also needed for the users to easily use our model worldwide and cross-platform.

*3) Criteria being evaluated:* Choosing the server for our project should consider the cost, availability, speed, reliability. All of these factors are extremely important. The reliability is the most important factor of them. Our data is precious and the machine-learning-algorithm-trained model is peculiar. Thus we can suffer any data loss. The speed is important as well. The AlphaGo from DeepMind trains itself by playing GO with itself millions of times every day. With fast speed, our model could be more accurate. Availability is also to be considered. For example, Google Cloud Server is not accessible while OSU server and Amazon Web Service are accessible in China.

*4) Table comparing options:*

| Technology | Economy | Readability | Availability | Speed | Notes |
|---|---|---|---|---|---|
| AWS | very | very | Worldwide | Very | TA very popular service by Amazon |
| GCP | very | very | Most Countries | Very | Not Accessible from China |
| OSU Server | Extremely | Normal | Worldwide | Varies | Speed slows down when off campus |

*5) Discussion:* When considering the server we use to host our model and data, it is extremely important to consider the factors mentioned above carefully. OSU Student Engineering Server is functionally complete. You can host Linux project there and OSU also provides database server. It is easy and familiar to use. If we have difficulty, we can directly access to tech support on campus. But it is not very fast outside campus. And OSU engineering server is not very flexible if Windows server is to be used. Also, student usually only have access to MySQL database[12]. Google Cloud Platform and Amazon Web Service are very similar. They both are hosted by giant companies in the US. The major difference between them when considering starting a new project is that Google is not accessible in China[13].

*6) Selection of best option:* Overall, after carefully comparing important aspects of them, I think Amazon Web Service is the most reliable and appropriate solution for our project.

### *H. Technology 8:Statistics Model to determine the importance of different categories of data*

*1) Options 1, 2, and 3:* py-statistics, scipy.stats, Pandas

*2) Goals for use in design:* An important aspect of our project is a statistical model to determine the importance of data. With good python library and combination of machine learning training, we will get a pretty good estimate of how these data factors affect our result. Choosing a functional and fast statistical library will make our data analyzing easier and make our prediction more accurate.

*3) Criteria being evaluated:* The cost are all free for the three libraries. The availability is important to be considered. The speed is important to consider when choosing the libraries. We are going to use the functions a couple times, even likely in a loop that cause higher complexity. So the basic running time of functions in these libraries are important. Security is not too important since we dont collect sensitive data from user but should be considered for the security and reliability of the system.

*4) Table comparing options:*

| Technology | Economy | Readability | Availability | Speed | Notes |
|---|---|---|---|---|---|
| py-statistics | Extremely | Extremely | Cross Platform | Not Sure | Official statistics library by Python Foundation |
| scipy.stats | Extremely | very | Cross Platform | Not Sure | Widely popular and sponsored Open Source Project |
| Pandas | Extremely | very | Worldwide | Cross Platform | Sponsored Open Source Project |

*5) Discussion:* Py-statistics is a powerful library developed by python foundation. It is reliable and widely used. Most importantly, Python Foundation will keep updating it to ensure its performance and security[13]. Scipy is a widely used library that may provide some functions that others don't. Scipy also highlights its probability functions which will suit our project for predicting basketball result[14]. Scipy is sponsored open source project by ENTHOUGHT. It is very reliable and popular. Pandas is a Python data analysis Library sponsored by NUMFOCUS. The Pandas library highlights its high-performance, easy-to-use data structures and data analysis[15]. It is also very popular, widely used and actively maintained by developers.

*6) Selection of best option:* All three libraries are very powerful. Our project will use py-statistics first. But the performance will need to be tested when in developing. The other two options may also be used in different context.

## II. Conclusion

There are many tools available which can be used to achieve the exact software product we desire. With careful consideration, we have located what we believe to be the correct tools for the job. For developing a graphical user interface which will allow students to easily and accurately interact with the module, we will be using PyGUI to develop our basic interface, Webix to provide the user with instructions on how to use the tool, and VTK to present the machine learned bracket in a way that is visually appealing. For getting the basketball data we will use sports-reference.com because of their downloadable csv files. We will store all the data using Amazon Aurora which offers a nice user interface for us to work with. We will be mainly using SciKit Learn library in Python for our machine learning algorithm. Amazon Web Service is the best choice to host our project on cloud.

## References

[1] Kivy: Cross-platform Python Framework for NUI, *Kivy*. [Online]. Available: https://kivy.org/. [Accessed: 14-Nov-2016].
[2] D. Bolton, 5 Top Python GUI Frameworks for 2015 - Dice Insights, *Dice*, Feb-2016. [Online]. Available: http://insights.dice.com/2014/11/26/5-top-python-guis-for-2015/. [Accessed: 14-Nov-2016].
[3] JavaScript Framework and HTML5 UI Library for Web App Development-Webix, *Webix*. [Online]. Available: https://webix.com/. [Accessed: 14-Nov-2016].
[4] UKI, *Best Web Frameworks*. [Online]. Available: http://www.bestwebframeworks.com/web-framework-review/javascript/117/uki/. [Accessed: 14-Nov-2016].
[5] J. Shore, An Unconventional Review of AngularJS, *Let's Code Javascript*, 14-Jan-2015. [Online]. Available: http://www.letscodejavascript.com/v3/blog/2015/01/angular_review. [Accessed: 14-Nov-2016].
[6] F. Lungh, Notes on Tkinter Performance, *Effbot*, 14-Jul-2002. [Online]. Available: http://effbot.org/zone/tkinter-performance.htm. [Accessed: 14-Nov-2016].
[7] VTK-Enabled Applications, *VTK*. [Online]. Available: http://www.vtk.org/. [Accessed: 14-Nov-2016].
[8] Wax GUI Toolkit, *Python*. [Online]. Available: https://wiki.python.org/moin/wax. [Accessed: 14-Nov-2016].
[9] Amazon Machine Learning, *Amazon*. [Online]. Available: https://aws.amazon.com/machine-learning/. [Accessed: 14-Nov-2016].
[10] "SciKit Learn,Machine Learning in Python, *scikit-learn*. [Online]. Available: http://scikit-learn.org/stable/. [Accessed: 14-Nov-2016].
[11] "Pylearn2 devdocumentation, *Pylearn2*. [Online]. Available: http://deeplearning.net/software/pylearn2/. [Accessed: 14-Nov-2016].
[12] "ONID - OSU Network ID, *Oregon State University*. [Online]. Available: http://onid.oregonstate.edu. [Accessed: 14-Nov-2016].
[13] "Python statistics Mathematical statistics functions, *Python*. [Online]. Available: https://docs.python.org/3/library/statistics.html. [Accessed: 14-Nov-2016].
[14] "Statistical functions (scipy.stats), *scipy.stats*. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/stats.html. [Accessed: 14-Nov-2016].
[15] "Python Data Analysis Library, *Pandas*. [Online]. Available: http://pandas.pydata.org. [Accessed: 14-Nov-2016].