# A DATA-DRIVEN ANALYTICAL SYSTEM TO OPTIMIZE SWIMMING TRAINING AND COMPETITION PERFORMANCE USING MACHINE LEARNING AND BIG DATA ANALYSIS

Tony Zheng[1] and Yu Sun[2]

[1]Troy High School, 2200 Dorothy Ln, Fullerton, CA 92831
[2]California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

***ABSTRACT***

*Many swimmers are constantly incorporating new and different training regimes that would let them improve quickly [2]. However, it is difficult for a swimmer to see their progress instantly. This paper develops a tool for swimmers, specifically swimmers, to predict their future results. We applied machine learning and conducted a qualitative evaluation of the approach [3]. The results show that it is possible to determine their future performance with decent accuracy. This application considers the swimmer's performance history, age, weight, and height to predict the most accurate results.*

***KEYWORDS***

*Machine Learning, Mobile APP, database.*

## 1. INTRODUCTION

Millions of young people dedicate themselves to the sport of competitive swimming [1]. They endure hours of training to push their athletic abilities forward. But the graph of effort vs progression is not linear. Sometimes swimmers experience a period of stagnant growth, causing them to lose faith in themselves and their efforts [4]. This application of machine learning will allow swimmers to see the light at the end of the tunnel. Since every swimmer will experience this problem of plateaued progress, this application will be utilized multiple times by millions of swimmers across the nation [5]. The application of machine learning is unlikely to leave, as each time the swimmers update their data, the algorithm will produce a different result [6]. By allowing the users to see a graph of progression, it will help swimmers get a clear sense of where they are in terms of progression. It also serves as a tracker for the swimmer's athletic performance. Using this application, swimmers can easily access data about their past performance. This will let the swimmer themselves compare and see the progression that they have achieved. It is very likely that the application of machine learning in performance data becomes an essential part of the swimmer's tool for checking the growth of their athletic abilities and part of the coach's strategy for examining the swimmer's potential.

David C. Wyld et al. (Eds): ARIA, SIPR, SOFEA, CSEN, DSML, NLP, EDTECH, NCWC - 2022
pp. 95-104, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121508

It is common knowledge to all time-based sports athletes that the graph of progression, performance time vs age, resembles the y=1/x graph [7]. In the beginning, there are huge improvements for athletes, with it not being uncommon to improve 5, 8, or even more than 10 seconds within weeks. But as they progress toward the limits of the human body, their progress slows dramatically or even comes to a halt. This is the ideal case, and as the world is not ideal. The first problem is that everyone has a different rate of progression that could be affected by numerous factors, causing impacts on accuracy. For example, some athletes experience a plateau in progression, which is stagnation in their athletic improvements. There are even those who experience a dip in performance even when they are training. The point is that everyone's progression is unique to their situation. The second problem is the inability to do so at scale. In order to determine the potential of the athlete, a person would have to know the athlete's performance and training. After obtaining this information, the person would have to deeply analyze the data for a long time. As a coach who has many athletes under their supervision, it is difficult to map out the rate of progression that their athletes are going through.

The solution proposed in this paper is the usage of machine learning algorithms [8]. Our goal is to accurately predict the performance times of swimming athletes. This method was inspired when I noticed a trend while viewing a graph of my swimming performance vs time. This graph shows a clear general curve that my past performance follows. So if it is possible to figure out the function that could generate that curve, then I will be able to accurately predict what kind of performance I will be able to have in a given year. With machine learning, it is possible to map out a progression graph for the athlete accurately and at scale.

In order to ensure that results were being generated, we ran tests to see the accuracy of the machine learning algorithm named AdaBoost [9]. In order to test the algorithm, we first create a model based on data that is available to us. Out of the 100% of our data, only around 80% of the data are actually used to generate these models. The rest of the 20% are used as tests todetermine the model's accuracy. When giving the machine learning model some data that it has never seen before effectively tests if the model is accurate. If the model-generated values are similar to the test data, the model would be considered accurate. The algorithm AdaBoost has tested a 99% accuracy after several trials. This means that the model is 99% accurate to match with the predicted results compared to actual data. With an accuracy this high, it is considered a valid model to use.

The paper will be organized into a total of 6 parts. Part 1 is the introduction so far. The next part, Part 2, will be discussing the challenges that were met on the way to the solution. Part 3 will be the solution to the problem described in the introduction, as well as the solution to challenges from Part 2. Part 4 will be detailing the experiments that were conducted. Part 5 is the related work that's similar to this paper. Finally part 6 is the conclusion that will summarize and give future works potential.

## 2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

### 2.1. Picking a specific Machine Learning Model for our predictions is difficult

Picking a machine learning model is a challenge, due to the different types of data that it could potentially have to process. There are many machine learning models that could solve a problem, but depending on the situation, one might yield higher accuracy than the other. For our problems, we have to work with an athlete's performance, which is measured in time. We need a model

that considers an individual's performance history to predict future times. Any regression model would work, but it would be better to have higher accuracy. This would be done by testing out the accuracy of each potential model. Using cross-validation to compare various regressive and classification models, we were able to pick out the best model for our specific use case.

### 2.2. Gathering related sports Swimming Data for our project database can prove challenging as there are not many resources

Gathering sufficient data is challenging because it is absolutely necessary for machine learning models. The more well-organized a model's data-set is, the more easily it can be trained and the more accurate its results will be. Therefore it is ideal to have both plenty of data and well-organized data. To predict the performance of a swimmer, we must have the swimmer's past performances. It is also crucial to have plenty of data so that it can be as accurate as possible. Their results also change as they improve over time. Overall, a database API would give all the data that is needed, if one is available. Many sites or databases allow one to simply import the data through their premade library. Then it is easy to format and make usable. Since there was no API available for any of the online databases, we obtained data through web scraping. Scraping through each individual's listed page of times, this method was able to gather the complete history of performance by any swimmer.

### 2.3. Predictions for each user cannot rely upon the data of other users

When running machine learning related to individuals, it is important to note that each person has a different condition or abilities. One person's data is not reflective of the future of another person. There are a huge range of athletes, from beginner to Olympic level. It would not make sense to impose a professional's requirements on a novice. A general solution would be to separate the data from person to person. This is usually done by creating profiles for each user. Similarly, we also separated each person's data by having users sign in to their own accounts. This way the user would have their own profile, unaffected by other people's data.

## 3. SOLUTION

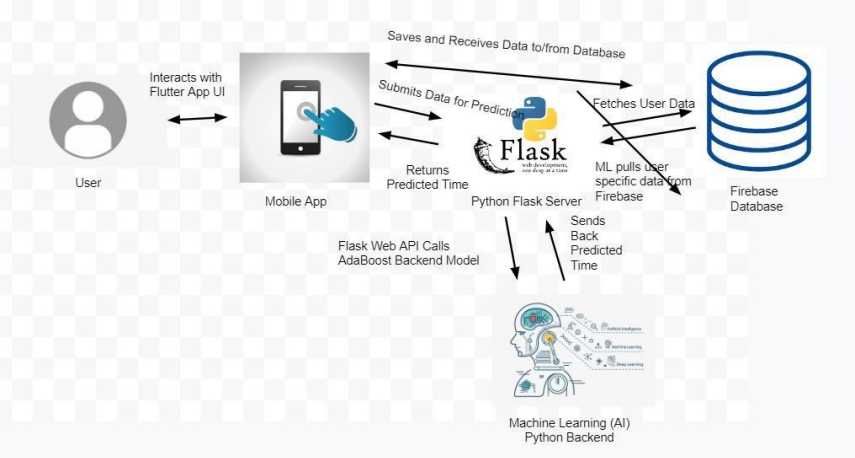


Figure 1. Overview of the solution

Figure 2. Mobile App

The interface for the user is a mobile app that allows the user to access and change information. This application, named Swim Wizard, is connected to a database as well as a server that responds to the user's prompts to run the machine learning algorithm. With the connection to the database, users can manually add, take away, and view the data that are under the user's access. The server, which utilizes flask, is connected to both the database as well as the mobile application to know when the user wishes to run the machine learning algorithm and can directly access the database for data to feed the machine learning algorithm. When the server needs to run the algorithm, it calls upon the AdaBoost backend model to run the code. When it is done, the results will be sent back to the server, then the mobile app, and ultimately the user. There are many components within this project. Each of them is closely connected to work as intended. All of the connections are two-way, as information needs to be both accessed and changed at all components. For ease of access as well as clarity, the user only has access to the mobile end, which will ultimately open access to every component within the project.

The mobile app was constructed using a developing platform called Android Studio [10]. Firstly built was the user interface. The creation of pages was followed by the population of buttons, text fields, drop down menus, graphs, user text fields, and much more. After the essential components were laid down, each component was given its own role and functions, so they can either act or be acted upon. For example, the functionality of what happens after the user presses a button was necessary for the user interface to work as intended.

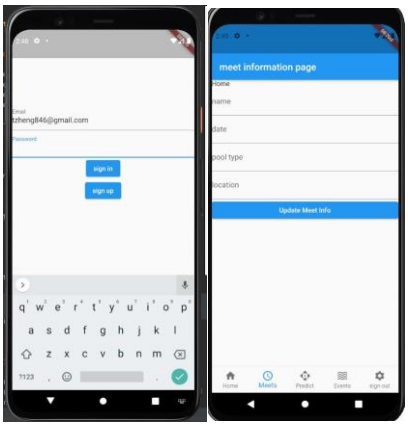


Figure 3. Sign in and information page

The database was built using Google's Firebase. It is an online database that can be easily connected to apps. Its most useful features, its free price, and the ease of integration were the reasons that this specific database was chosen. After creating a database, it is necessary to organize the data by categorizing them based on hierarchy. The hierarchy would be ordered as the user's id, meet, event, and performance time. This ensures ease of access when the user is trying to find a specific performance time.

```
84
85  #Return a Meet object (Dictionary)
86  def createMeet(name, events, _date, poolType, location):
87    meetDict = {
88      name : {
89      'events' : events,
90      "date": _date,
91      "pool type": poolType,
92      "location" : location
93      },
94    }
95    return meetDict
96
97  createUser("tempPerson@gmail.com", 26, 5.11, 160, "temperson", "team Team")
98  getUserInfo("tempPerson@gmail.com")
99
100 event_info = createEvents([["100 FR", "afternoon", 49], ["50 FR", "morning", 23]])
101 meet_info = createMeet("Cheezit Invitational", event_info, "Mon 4th", "LCM",
    "Columbia")
102 updateUser("tempPerson@gmail.com", meet_info)
103
104
```

Figure 4. Screenshot of code 1

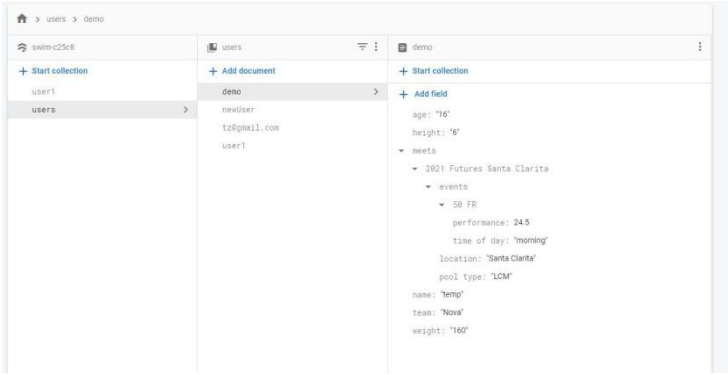


Figure 5. Screenshot of code 2

The server was made using Python Flask. The server's task is to listen for requests to run certain codes. When it receives a request from the mobile app, it runs a snippet of code. For example, when the mobile app requests to run the predict function, the server, upon receiving the request, obtains necessary data from the database and runs the Adaboost machine learning algorithm. Once the algorithm returns a result, the server sends the result back to the mobile app. AdaBoost was chosen as the prime choice of machine learning algorithm due to its superior accuracy during tests.

```
app = Flask(__name__)

@app.route("/")
def home():
  return "Welcome to our Swim App"

@app.route("/test")
def test():
  return "TEST SUCCESSFUL!"

def sec_to_mins(seconds):
    a=str(int((seconds%3600)//60))
    b=str(int((seconds%3600)%60))
    c=["{} mins and {} seconds".format(a, b)]
    return c


# https://Tony-Swim-Time-Web-Scraping.gunnellevan.repl.co/runPrediction/12_3_2020/home/Free/200/15/0/0/Yd
@app.route("/runPrediction/<Date>/<location>/<stroke>/<distance>/<age>/<gender>/<team>/<pool_type>/<rankId>")
def doPredictions(Date, location, stroke, distance, age, gender, team, pool_type,rankId):
  f = open("data.txt", "r")
  data = json.load(f)
  f.close()
```

Figure 6. Screenshot of code 3

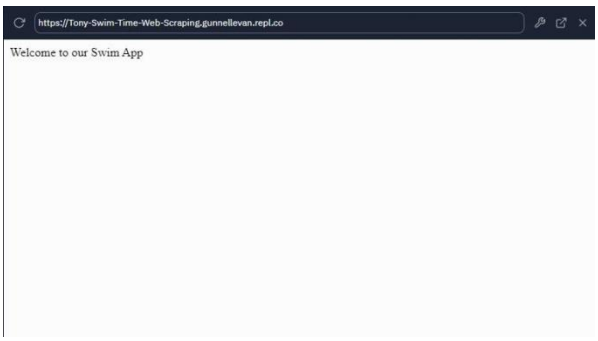


Figure 7. Screenshot of website

Then came the final part of integrating all the parts. Using the instructions from Google's Firestore, it was easily integrated with both the mobile app and the server. It establishes a two-way stratosphere of data for each part. Finally, the mobile app was easily linked up with Python Flask when the URL was provided to the app.

## 4. EXPERIMENT

Our goal is to determine if it is possible to utilize machine learning to predict the future performance of a swimmer. Some questions need to be addressed. What is the best machine learning algorithm? What CV is the best for machine learning? What parameter has the most effect on the results of the prediction? The experiment that we set up specifically addresses these questions. First, we gathered many different machine learning algorithms. Then we tested each algorithm using built-in scikit-learn functions. We used the "cross_val_score()" function to obtain the accuracy of each function. The function requires many parameters such as the model, input, output, and CV score. By setting all algorithm's parameters as constants, we were able to determine the accuracy of each algorithm. The second question is solved by setting the algorithm and all parameters except for the CV score as constants. By adjusting the CV score, we were able to determine the most optimal CV score. The third question is solved by using the selected machine learning algorithm and using a function from scikit-learn library to test out the effectiveness of each parameter.

We were able to collect the data that we intended to gather. The data that is shown in figure 4.1 and 4.2 shows the accuracy performance of each model. Using the two figures, it is clearly shown that AdaBoost has the highest accuracy with an outstanding 95.06% accuracy. Thus we determined that AdaBoost is the machine learning algorithm that is best suited for predicting future swimmer performance. The data for experiment two is recorded in figure 4.3 and graphed in figure 4.4. Using these data, we concluded that using a CV score of 7 produces the best results. We also speculated that the higher the CV score, the better the accuracy. But since we have not tested any higher CV count higher than 7, we can not conclude that the statement is correct. The data for experiment three is recorded in figure 4.5 and graphed in 4.6 for visual clarity. Using the two figures, we can see that there are only two major factors to the production of results. Age takes up a majority, with date filling up almost the rest. The effects of location are almost negligible.

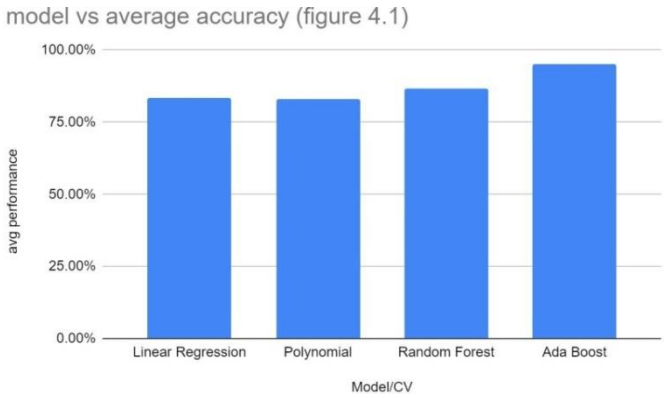


Figure 4.1 Model vs average accuracy

| Figure 4.2 | |
|---|---|
| Model | avg performance |
| Linear Regression | 83.52% |
| Polynomial | 83.09% |
| Random Forest | 86.74% |
| Ada Boost | 95.06% |

Figure 4.2 Model vs average performance two

| figure 4.3 | | | |
|---|---|---|---|
| Model/CV | 3 | 5 | 7 |
| Linear Regression | 86.45 | 77.65 | 86.45 |
| Polynomial | 83.05 | 83 | 83.22 |
| Random Forest | 86.84 | 86.45 | 86.92 |
| Ada Boost | 94.83 | 94.52 | 95.83 |

Figure 4.3 The data of experiment

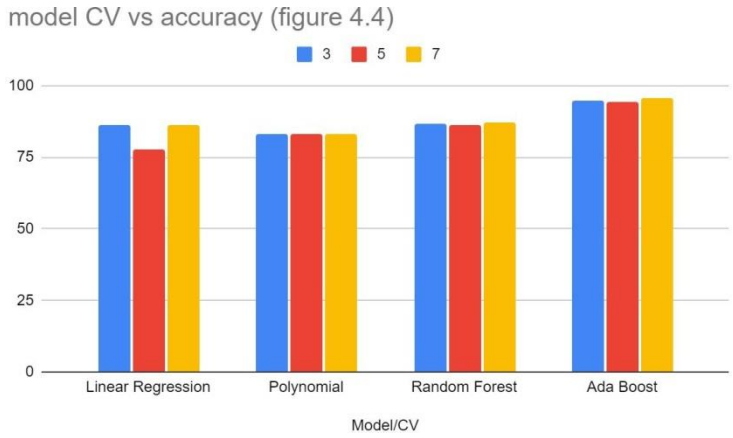


Figure 4.4 The graph of experiment two

| Figure 4.5 | |
|---|---|
| Date | **0.20556553** |
| location | 0.00486057 |
| stroke_type | 0 |
| distance | 0 |
| age | 0.7895739 |
| gender | 0 |
| team | 0 |

Figure 4.5 The data of experiment three

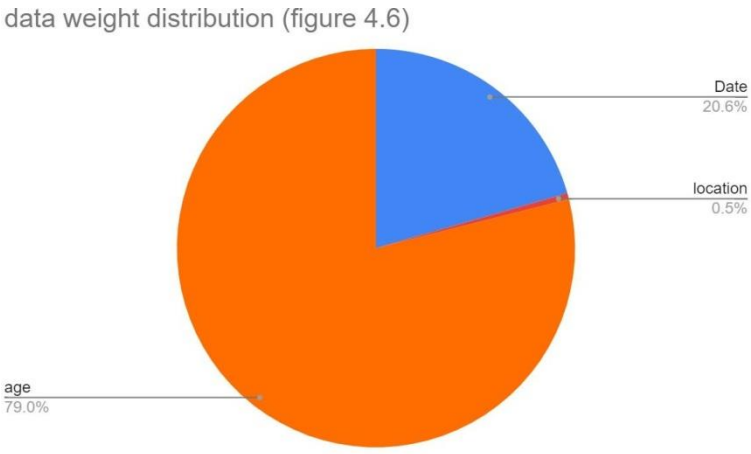


Figure 4.6 The graph of experiment three

The three experiments went as expected. We were able to gather the data that we expected. For experiment one, we initially thought polynomial regression would be the top pick. But the results of the experiment proved that AdaBoost is superior for our intended purposes. AdaBoost outperformed the second best choice, Random Forest, by 9% in accuracy. This is a significant performance difference. For the second experiment, we did not know what to expect. But the results showed that a CV of 7 outperformed CV score of 3 and 5. This was true for all algorithms

that we tested. Finally, experiment three is mostly what we have expected. We also predicted that the most prominent factor would be time related. Both date and age are very similar data that relates to time.

## 5. RELATED WORK

Using a prediction model of machine learning, Zhu aims to accurately predict the athlete's performance [11]. It improves the prediction model by incorporating specific changes of the athlete's performance, finding hidden rules using chaotic theory, and using vector machines and particle swarms. Zhu uses advanced techniques in order to obtain extremely accurate results. The application of this paper is strictly analytical. Compared to this paper, SwimWizard is tailored to the swimmers, allowing them to view their performance history as well as get a decent prediction of their future results.

This paper regression analysis in order to research the critical period of swimmer's athletic training [12]. In addition, it also reviews many methods of predicting swimming performance using correlation of swimmer age. "Machine learning of swimming data via wisdom of crowd and regression analysis" is a very in depth analysis of using quantitative data in order to find the answers to many important answers to swimming. It is significantly more advanced than this paper. The main difference in our work is that we explore not only age, but other factors that could affect performance. These factors that we explored include location, and team. Since some locations may provide better facilities, causing a difference in performance. In addition, each team offers a different training regiment and different coaches.

This paper focuses on the classification of breaststroke styles for each swimmer [13]. Using machine learning, the author hopes to find a way to identify the difference in technique for each swimmer. Although both this author and this paper focuses on swimmer performance, there is a huge difference. "A Machine Learning Approach to Breaststroke " uses categorization machine learning algorithm. They have highly complex algorithms that feed on visuals on the technique of breaststroke. It analyzes the technique qualitatively and produces results via clustering. We focus purely on quantitative analysis, using a set of values to produce another value.

## 6. CONCLUSIONS

Using data from the history of a swimmer's performance to predict the swimmer's future performance. This could be done by feeding data into a regression type machine learning algorithm. With Scikit learn library functions, we have found that AdaBoost is the best machine learning algorithm [14]. Using the AdaBoost machine learning algorithm, we have achieved 95% accuracy in prediction. It produces a reasonable result. Although 95% is very high in terms of general accuracy, unfortunately it is not enough as a satisfactory result for a swimmer, as even 1% could cause a huge variance. We tested many different parameters such as age, location, team, gender and date. We found out which variable can cause differences in results, as well as how much of an impact each variable makes.

One of the biggest limitations is the amount of data we had access to during the experiment. Using only one swimmer's data, we were very limited in our options. As such, the accuracy as well as the conclusions will be skewed. The practicability of producing a prediction using machine learning that is better than the prediction of a professional coach is very low. Although we found out that we can achieve 95% accuracy, it is not enough. A professional coach is able to create a prediction that comes close to 99% accuracy. This could optimize this by trying this experiment on a larger data set.

We want to get access to more data. In order to do this, we could apply to gain access to the official USA Swimming data set, which contains information about millions of swimmers [15]. With a larger data set, the machine learning model would be able to train much more than previously. This would likely result in a much higher accuracy in its predictions.

## REFERENCES

[1] Mujika, Inigo, et al. "Effects of training on performance in competitive swimming." Canadian journal of applied physiology 20.4 (1995): 395-406.
[2] Geiger, Mario, Leonardo Petrini, and Matthieu Wyart. "Landscape and training regimes in deep learning."Physics Reports 924 (2021): 1-18.
[3] Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." Science349.6245 (2015): 255-260.
[4] Petrakis, Panagiotis E., Dionysis G. Valsamis, and Kyriaki I. Kafka. "From optimal to stagnant growth: Therole of institutions and culture." Journal of Innovation & Knowledge 2.3 (2017): 97-105.
[5] Conner, Deondra. "The effects of career plateaued workers on in-group members' perceptions of PO fit." Employee Relations (2014).
[6] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9 (2020): 381-386.
[7] Pierson, William R., and Henry J. Montoye. "Movement time, reaction time and age." Journal of Gerontology 13.4 (1958): 418-421.
[8] Singh, Amanpreet, Narina Thakur, and Aakanksha Sharma. "A review of supervised machine learning algorithms." 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). Ieee, 2016.
[9] Vezhnevets, Alexander, and Vladimir Vezhnevets. "Modest AdaBoost-teaching AdaBoost to generalize better." Graphicon. Vol. 12. No. 5. 2005.
[10] Esmaeel, Hana R. "Apply android studio (SDK) tools." International Journal of Advanced Research in Computer Science and Software Engineering 5.5 (2015).
[11] Zhu, Pan, and Feng Sun. "Sports athletes' performance prediction model based on machine learning algorithm." International Conference on Applications and Techniques in Cyber Security and Intelligence. Springer, Cham, 2019.
[12] Xie, Jiang, et al. "Machine learning of swimming data via wisdom of crowd and regression analysis." Mathematical Biosciences & Engineering 14.2 (2017): 511.
[13] Zanchi, Marco. "A Machine Learning Approach to Breaststroke."
[14] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.
[15] McCubbrey, Donald J., Paul Bloom, and Brad Younge. "USA Swimming: the data integration project." Communications of the Association for Information Systems 16.1 (2005): 13.