

AN INTELLIGENT AND DATA-DRIVEN MOBILE PLATFORM FOR YOUTH VOLUNTEER MANAGEMENT USING MACHINE LEARNING AND BIG DATA ANALYSIS

Alyssa Huang¹ and Yu Sun²

¹Arnold O. Beckman High School, Irvine, CA 92602

alyssahuang10@gmail.com

²California State Polytechnic University, Pomona, CA, 91768

yusun@cpp.edu

ABSTRACT

Volunteers are often desperate to find any volunteering opportunities available to them but do not have the information to judge which events will suit their personal needs. In this paper, we use machine learning algorithms to study the correlation between 1) volunteer information (e.g. interests, previous volunteering events, supervisor ratings, etc.) and volunteer performance in specific events and 2) the correlation between volunteer information (e.g. interests, previously volunteered events, etc.) and event recommendations. Two highest-scoring machine learning algorithms are proposed to make predictions on volunteer performance and event recommendations. We demonstrate that the two highest-scoring algorithms are able to make the best prediction for each query. Alongside the algorithms, an application, which allows efficient communication and management between supervisors and volunteers is created. This application will facilitate volunteer and supervisor interactions, and allow volunteers and supervisors to utilize the machine learning predictions for an overall more positive and beneficial experience for both sides.

KEYWORDS

Machine learning, Flutter, Volunteer Management, Scikit-learn

1. INTRODUCTION

Community service and volunteering events are prevalent, not only helping serve the underserved but also providing people, particularly younger students, with the opportunity to better their community and influence those around them [3]. Since many universities seek students who demonstrate the passion and ability to impact those around them, community service and volunteering often play a significant role in college applications. Aside from general community service events such as packing food boxes for the poor and donating to toy drives, there are also specialized events that focus on a particular topic or subject. Participating in volunteering projects relating to a student's interest is both gratifying and beneficial to the student's resume [1][2].

Students are often desperate to find any volunteering opportunities available to them. However, many students don't have the information to make a judgment about what type of event will suit their pursuit, passion, and ability. Similarly, volunteer events may have trouble deciding which volunteers to accept in order to maximize the execution of events. Instead, they may follow a first-come, first-serve basis, or have simple minimum requirements. Doing so may lead to an unsatisfactory performance from volunteers who are disinterested or unsuitable for the event. This paper serves to solve this problem by being the bridge between volunteers and supervisors,

and by finding connections between volunteers and performance. By doing so, not only will volunteers generally have a more positive experience during volunteering and find extra benefits in participating in events, but event organizers will also boast better results during events.

Open Questions. In this paper, we aim to solve the aforementioned problems regarding volunteers and to answer the following inquiries:

1. *Given the user's previous performance on events, interests, and age, among other details, and the event's type, time commitment requirement, and additional information, how well will the user perform in volunteering for this event?*
2. *From the user's interests and previously participated events, what event should be recommended to the user?*

Many studies have found other methods in predicting volunteer activity and outcomes. For example, one study found that grouping volunteers into different motivational groups will predict volunteer outcomes better than knowing only the motives of the volunteers. Furthermore, another study proves that the motives of volunteers have a weaker than expected effect on the length of volunteerism and service. Rather, a volunteer's identity and expectations are what best predicts the period of volunteerism. A different study links subjective experiences before and during volunteering to the ultimate satisfaction and length of volunteer service. This study, however, aims to predict the effects of volunteerism on college admissions and the effects of volunteer interest on volunteer outcomes.

Solution. In order to make reasonable predictions to answer the previously mentioned queries, a three-part system was created to gather data, store data, and make predictions using the data. To gather data, we implemented a multi-use mobile application that volunteers and supervisors may use to create and join events, log and approve hours, and log volunteer interests, future goals, etc. The app is easily navigable and contains many functions that volunteers and supervisors need. Any data gathered from this app is stored inside an organized cloud database. The database has functions that also allows for secure user authentication, so users can log in safely. At any time, the app may query any data from the database about the user to display on the app. A machine learning framework that gathers data from volunteers and makes predictions was implemented as well. We generated dummy data, mimicking data of real volunteers, in order to train the machine learning algorithms. Through multiple trials, we were able to find the best-performing algorithms to use in predicting each question. In the end, the chosen algorithms are run on a server that the app may access at any time to show prediction results to users. The use of dummy data allowed the machine learning algorithms [4][5] to train and produce the best possible results. Together, the three systems work together to accurately make the best predictions possible.

Various experiments were devised in order to find the top-performing algorithm for making predictions for each question. For each question, multiple experiments were run, differing each time in areas such as algorithm parameters, volunteer feature sets, and training data sizes. Changing these areas allows us to determine if a certain prediction model depends heavily on one of these areas. Training the machine learning algorithms to make the best possible prediction required tens of thousands of generated dummy data. To easily compare the prediction results with a separate dataset to prove its accuracy, the data is randomly split into two sets: a training set and a test set. Through running multiple trials, the mean accuracy of the prediction was generated for classification algorithms, and the coefficient of determination, the R2 value, of the prediction for regression algorithms. Each experiment generated multiple mean accuracies and R2 values that were then graphed in order to find the top-performing algorithm for each question. Analysis of the graphs shows that each prediction model yields a consistent

accuracy of prediction through multiple trials, however, the accuracy of prediction differs greatly between multiple models. Any further interesting finds were noted as well. It was found that generally, one prediction model trumped the other prediction models, regardless of changes in algorithm parameters, volunteer feature sets, or training data sizes.

The rest of this paper is structured as follows: Section 2 will detail the multiple challenges faced in this study and how they were overcome; Section 3 will describe the methodology and solution in greater detail; Section 4 details the experiments that were performed in this study, as well as a thorough analysis of the results; Section 5 will list any related works that have also been done regarding volunteering; and lastly, Section 6 will conclude the study and state any future work that may be done.

2. CHALLENGES

In order to build the data-driven application system, a few challenges have been identified as follows.

2.1. Challenge 1: The Heterogeneous Data Communication Between Multiple Roles

Traditional management of volunteers by event supervisors include email, texting, or online chat services. However, these methods of communication may be slow and require extra planning. This application aims to provide instant management of events and volunteers. Appropriated for multiple types of users from volunteers to supervisors, this application must facilitate data communication between the supervisors and the volunteers. Volunteers must be able to see event data and join events the supervisors have created, and supervisors must be able to view volunteer profile information, create events, and manage volunteers. When a supervisor creates an event, there must be some way for volunteers to view the event and join the event, and when a volunteer joins an event, the supervisor must be able to view the volunteer and his/her profile. Real-time data updates must be available for users, so no communication errors occur. To solve this challenge, a cloud database will be used, so the application can read and write data at any moment. Any information that users log, including hours, events, or profile changes, will be written to the cloud database. This will allow other instances of the application to read the changes that have been made.

2.2. Challenge 2: The Complexity of the User Roles and the Associated Data Structures

In order to ensure efficient and effective communication between roles, any data logged by users must be structured to allow for easy retrieval and writing of data. Since the application is targeted towards both supervisors and volunteers, there are many types of data subsets that need to be organized and collected. These may include variables such as volunteer hours log, volunteer joined events, supervisor created events, or profile changes. Without the correct organization, data may be redundant or lost, causing problems in readability and query time. The data being collected is organized into multiple collections and subcollections. For this application, there are three main entities: profile information, organizations, and events. Profile information includes any data regarding users, such as name, email, age, gender, joined/created events are logged under this entity. The data in the Organizations collection includes organization name, join code, administrators, members, etc. The Events collection includes the organization it belongs to, date, time, supervisors, volunteers, etc. Together, all three entities are connected, and often, data from one entity may be used to query data from another entity. For example, accessing the profile information will allow us to view the list of events that the volunteer is currently signed up for, and using this list of events, further information regarding each event may be queried from the Events collection.

3. SOLUTION

This application is a volunteer management platform that allows volunteers to track hours and join events and allows supervisors to create and manage activities. The application has three components that ensure the highest quality performance for the user: the frontend application, the backend cloud database, and the backend machine learning server. The application, written in Dart in Flutter [7], can run on both iOS and Android devices. It hosts many features that allow both volunteers and supervisors to keep track of their workflow. The app is also able to communicate with the cloud Firebase. The cloud database uses Google Firebase [8] and stores data as JSON objects [9]. Google Firebase contains a multitude of features, such as analytics, authentication, and storage. The application directly writes data into the Firebase and is later able to read the data. Lastly, the machine learning code is written in Python and hosted on a server. The Python server will access any data gathered by the application and use the data to make predictions about the user. There are multiple machine learning algorithms, to ensure each query uses the best prediction model to its needs. These three components constantly exchange information. The main technical challenge is keeping the three systems in check and communicating correctly, with data produced on two different occasions. It is crucial to keep all data organized and stored in the correct places to avoid any errors in communication between the systems. The following section will describe each component in further detail.

The frontend of this application was built in the IntelliJ integrated development environment (IDE) using Google's open-source UI software development kit Flutter. Flutter uses the object-oriented programming language, Dart [6].

The application, targeted towards both the volunteer and supervisors of events, will provide each user access to one of three platforms: the volunteer only platform, the supervisor only platform, and the volunteer and supervisor platform. Each platform has its respective pages, depending on the user. Upon entering the app, all users will be prompted with the same login screen, with the option to either login or sign up for an account. After logging into an account, a home screen will appear with a bottom navigation bar containing different pages depending on whether the user is a volunteer, supervisor, or both. In the case of the volunteer only, the pages Home, Events, Hours, and Profile are shown. Figure 1, shown below, displays the volunteer only platform. In each page, users will be able to view their enrolled events, view any upcoming events, log their volunteer hours, and edit their profile, among other features. Any data, including hours, registered events, and profile changes, is stored in a cloud database that the application will be able to access.

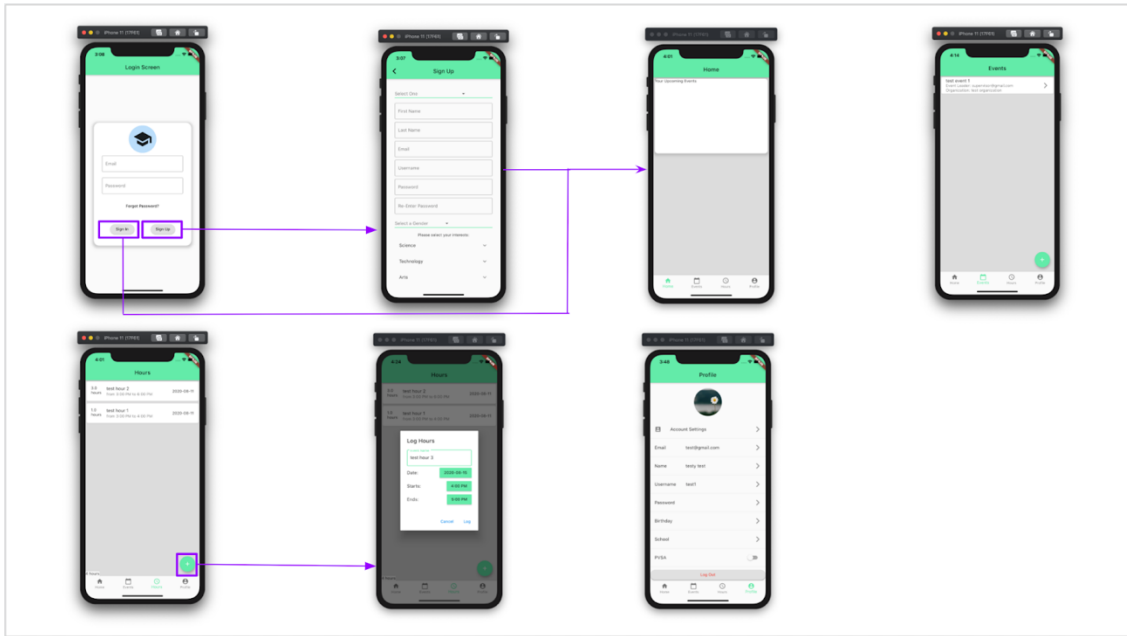


Figure 1. The Volunteer Only platform storyboard

The supervisor only platform follows a similar flow, shown in Figure 2. After logging into a supervisor only account, the home page appears, with a bottom navigation bar that contains the pages Home, Events, Organizations, and Profile. In these pages, the user will be able to see their upcoming events, approve volunteer hours, create and manage events, join and create volunteer organizations, and edit their profile, among other features. Any data shown in these pages are stored in the cloud database as well.

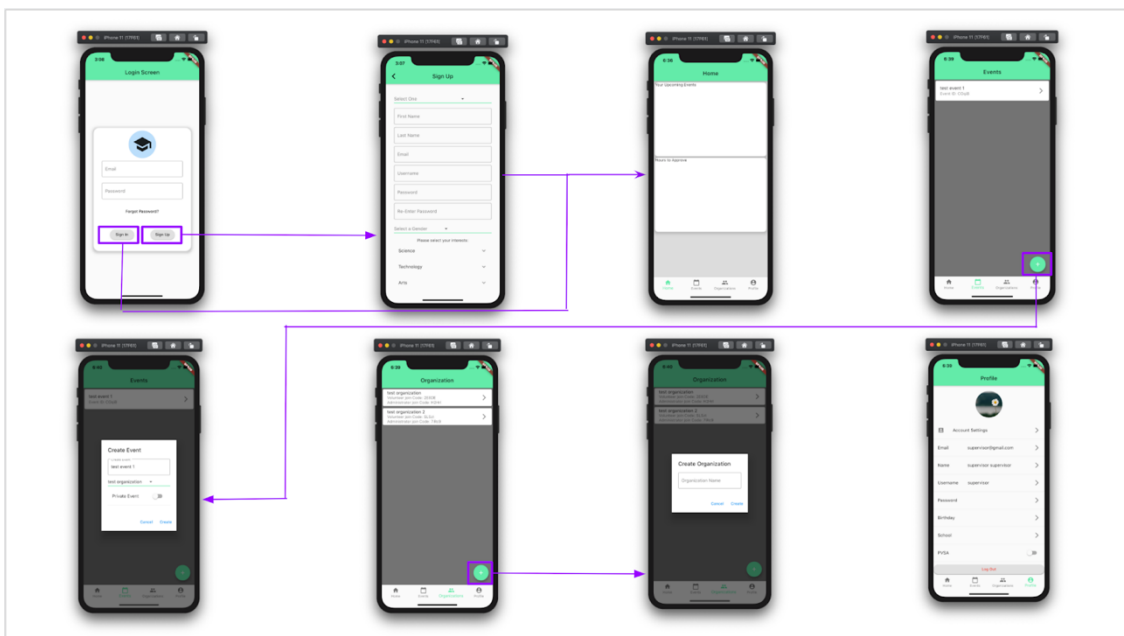


Figure 2. The Supervisor Only platform storyboard

The application includes complex code to load data from the firebase. This function is attained by employing elements such as asynchronous functions and future data types. The use of the

two components mentioned before allows the application to run multiple tasks at once and store and use data that will exist in the future. Shown in Figure 3 is an example of the code. In the snippet, a query is sent to the database using an asynchronous function. The use of an asynchronous function allows the application to make a query to the Firebase while other tasks are running or loading. Usually, when using a variable, there must be a declaration of what the variable is representing. However, this can be problematic if the data does not exist yet, such as while data is being loaded from a database. The use of a future data type solves this problem, as instead of throwing an error, it can wait for the data to be a non-null type before showing the data.

```
Future<Map> retrieveEventData(String organization, String eventCode) async {  
  var eventInfo;  
  await firestore  
    .collection('organizations')  
    .document(organization)  
    .collection('events')  
    .where('eventID', isEqualTo: eventCode)  
    .getDocuments()  
    .then((value) {  
      var documents = value.documents[0];  
      eventInfo = {  
        'organization': documents['name'],  
        'eventLeader': documents['eventLeader'],  
        'eventID': documents['eventID'],  
      };  
    });  
  return eventInfo;  
}
```

Figure 3. A snippet of the application code

The backend of this application includes two portions: the machine learning server and the cloud database. To make predictions regarding the users, the machine learning model and cloud database consistently communicate. Since user data is stored in the cloud database, the machine learning model must be able to read data from the database and produce a result and subsequently store this result into the cloud for the front end to query.

Google Firebase is used to store and read data efficiently. The Firebase contains features that allow user authentication by email, phone number, or other accounts without any additional code. Using this feature, users may create accounts and safely log in through the app. In the Firebase, there are three main types of entities stored as documents in collections of the same name: users, organizations, and events (Figure 4). Each entity has its own properties, and some have their own subcollections. The Firebase is organized sensibly as such so that the data can be queried efficiently by the application.

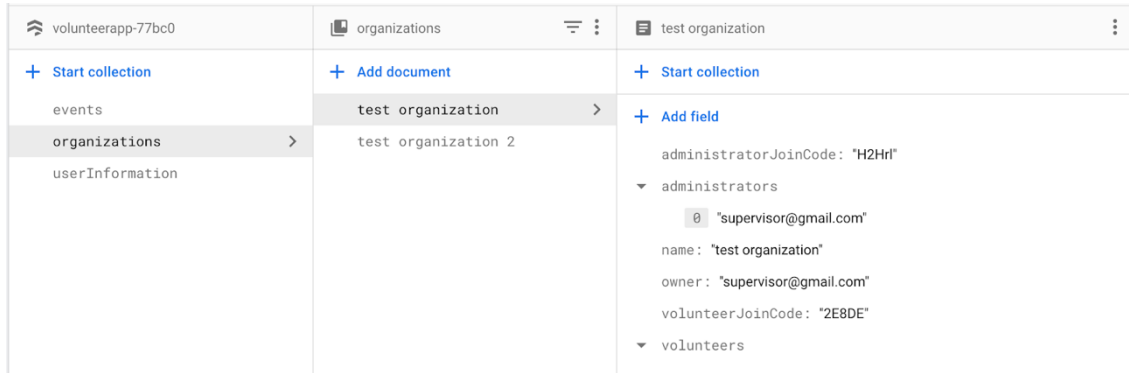


Figure 4. The “organizations” collection within the Firebase

The machine learning framework in the backend is written in Python and employs Scikit-learn, a software machine learning library for Python. Scikit-learn features a myriad of regression, classification, and clustering algorithms. The Scikit-learn is used to make predictions on the three inquiries mentioned above.

Multiple algorithms are tested to find the top-performing algorithms for each prediction. To train the prediction models, first, tens of thousands of accounts of dummy data was generated, with data including age, gender, interests, etc. The variables within the data were linked, for example with volunteer performance increasing with age. Next, to train the algorithms, a class in the Sci-kit Model Selection library named train test split was used. This class divided the dummy data into two groups; one would train the model and the other would be used to test the model. This allows the predictions of the training set data to be compared accurately with the actual results from the test data. After running multiple training tests, the algorithm that consistently performed the highest was confirmed.

4. EXPERIMENT

When signing up to volunteer for an event, a volunteer may not know how well they will perform in the event due to not completely understanding the requirements and expectations of volunteering for the event. In this experiment, we aim to find the best machine-learning algorithm to predict a volunteer’s performance on a scale of one to ten based on preceding experience, user information, and event information.

To train the algorithm, data from previous users, and their ratings after each event is used. When a prediction is required, it will take into account the current user’s profile information and past ratings to make a prediction on how well their performance on the specified event will be. Using the Scikit-learn machine learning library in Python, multiple algorithms, including linear regression, polynomial regression, random forest classifier, k-nearest neighbor classifier, Gaussian naive Bayes, and random forest bagging classifier, are tested for accuracy in prediction [13][14][15]. Each prediction model is trained with dummy data variables that include volunteer age, event type (science, engineering, math, writing, public speaking, etc.), interests, and a rating on the event. Although all volunteer profile information and the event type is randomly generated, the performance is tied to these generated numbers. For example, if the event type is science, and if the user is interested in science, then the minimum performance will be increased. Similarly, the minimum performance of the volunteer will increase with the age of the volunteer.

Multiple experiments are run, each with multiple trials. Each experiment alters one part: algorithm parameters, volunteer feature sets, or training data sizes. Through these experiments,

we will be able to determine if an algorithm depends on certain factors to produce the best results. Each algorithm is then tested and scored to find the most accurate prediction model.

Figure 5 illustrates the results of experiment 1. Over twelve trials, the random forest bagging classifier consistently outperformed the rest, followed closely by the random forest classifier. By contrast, the linear regression prediction model consistently scored the lowest.

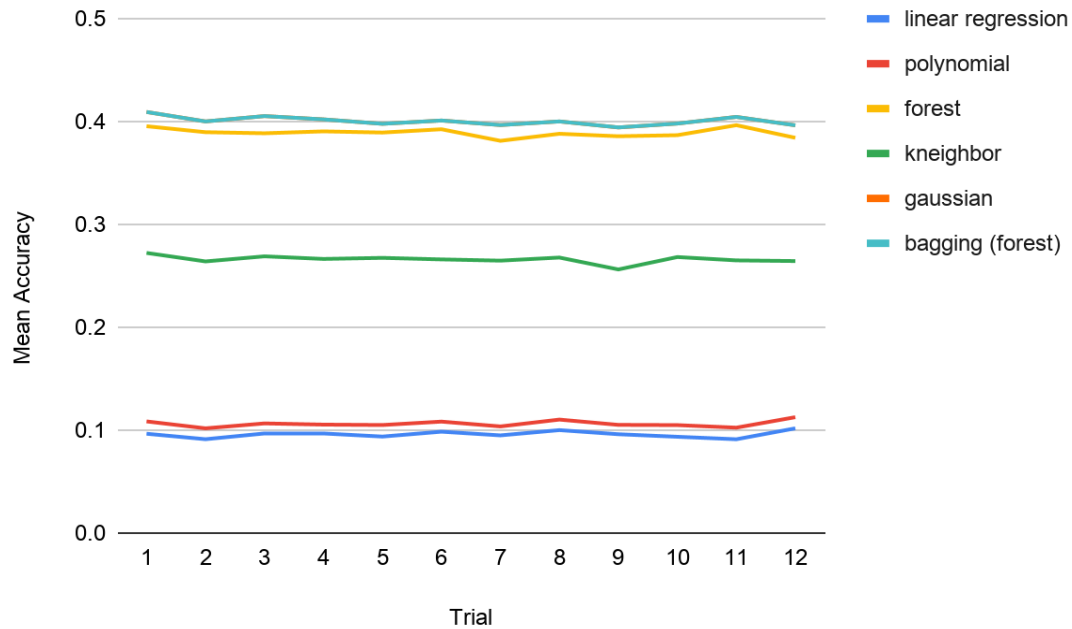


Figure 5. Experiment 1.1 Results Graph

An interesting find was that the random forest bagging classifier and the Gaussian naive Bayes algorithms scored exactly the same numbers, as can be seen in Figure 6. After further examination, it was concluded that this was an error in the Python code.

Table 1. Experiment 1.1 Results

	r ² value		mean accuracy on the given test data and labels	mean accuracy on the given test data and labels	mean accuracy on the given test data and labels	mean accuracy on the given test data and labels	mean accuracy on the given test data and labels
trial	linear regression	polynomial	svm	forest	kneighbor	gaussian	bagging (forest)
1	0.09659642072	0.1085847523		0.3956	0.2724	0.40945	0.40945
2	0.0912802714	0.1018627497		0.3898	0.26415	0.40025	0.40025
3	0.09686422492	0.1066349464		0.38875	0.2691	0.40545	0.40545
4	0.0968569656	0.1054718431		0.3906	0.26655	0.40225	0.40225
5	0.09385395767	0.1050779491		0.38945	0.2676	0.39795	0.39795
6	0.09858931497	0.108387892		0.39265	0.2661	0.40115	0.40115
7	0.09501043918	0.1037082948		0.3814	0.2649	0.39675	0.39675
8	0.1001078768	0.110352902		0.38825	0.2679	0.40025	0.40025
9	0.09611727669	0.1052227573		0.38585	0.25635	0.3944	0.39445
10	0.09364955503	0.104988994		0.3869	0.2684	0.3982	0.3982
11	0.09123142702	0.102510737		0.3966	0.26515	0.40465	0.40465
12	0.1018312775	0.112689413		0.38435	0.2645	0.3965	0.3965

To ensure that the random forest bagging classifier would be the best predictor of volunteer performance, another experiment (Experiment 2) was conducted, changing the feature set of the volunteer dummy data. In addition to the previous data set, which included volunteer age, event type (science, engineering, math, writing, public speaking, etc.), interests, and a rating on the

event, this experiment added extra variables: the number of hours required for the volunteer to volunteer in the event and the volunteer's previous average ratings in previous events. These variables were once again linked to the randomly generated performance number: the higher the volunteer's previous rating, the higher the minimum performance of the volunteer will be, and the more hours are required for the event, the lower the minimum performance of the volunteer will be.

Figure 6, the graph for experiment 2, shows that the polynomial regression, random forest classifier, and the Gaussian naive Bayes algorithms scored very similarly. In contrast with Experiment 1, where polynomial regression was one of the worst-performing algorithms, polynomial regression scored exceptionally.

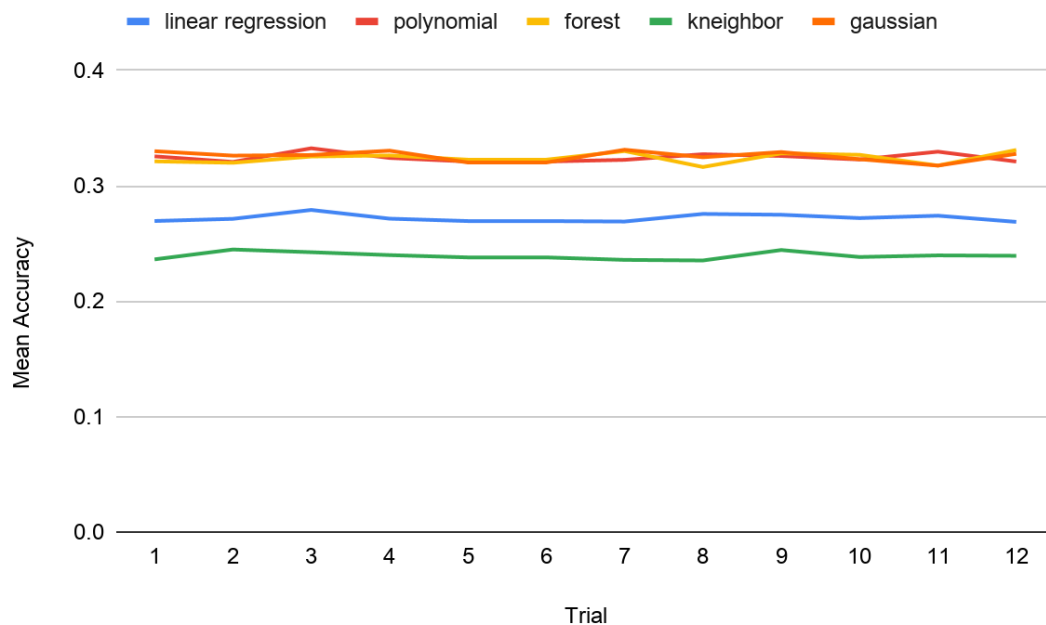


Figure 6. Experiment 1.2 Results Graph

In experiment 3, we altered the model parameters of one prediction model. Since the random forest bagging classifier did not currently include any parameters, we altered the parameters of the second-best predictor thus far, the random forest classifier. The random forest classifier requires two parameters: n-estimators and max depth.

In the results, shown in Figure 7, it is clear that most changes in parameters yielded similar results, except when n-estimators was a small number and max depth was a large number.

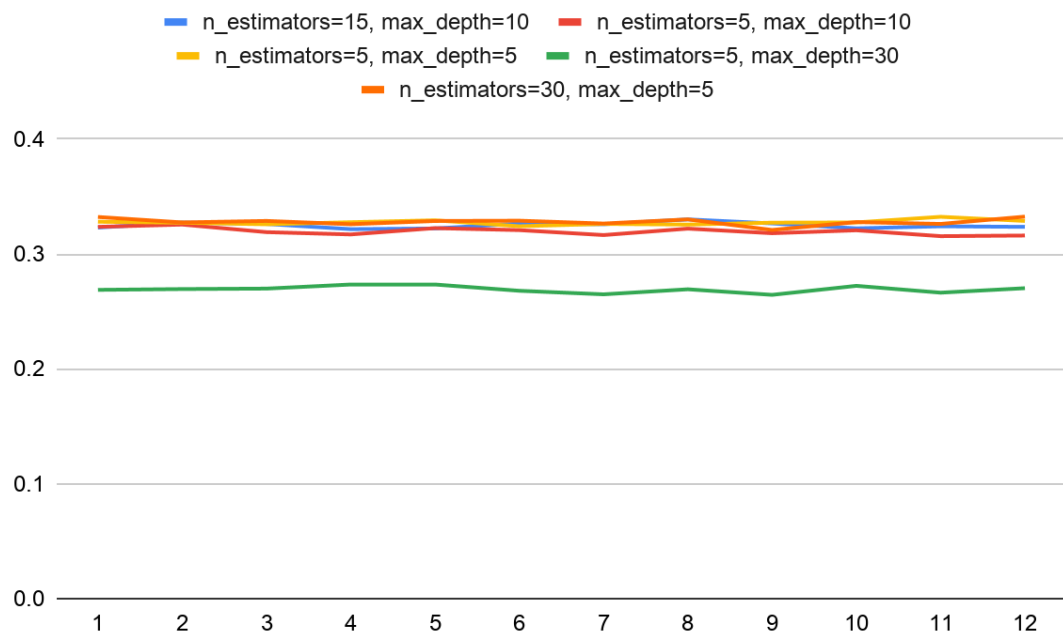


Figure 7. Experiment 1.3 Results Graph

Experiment 4 explored the impact of sample size on the accuracy of the prediction. In Figure 8, it is apparent that in the trials with 1000 volunteers, the prediction model scores less consistently than the trial with 10000 volunteers does. It was found that rather than increasing the accuracy of the prediction models, training the prediction models with more sets of data would increase the consistency of the predictions.

10000 trials vs. 1000 trials

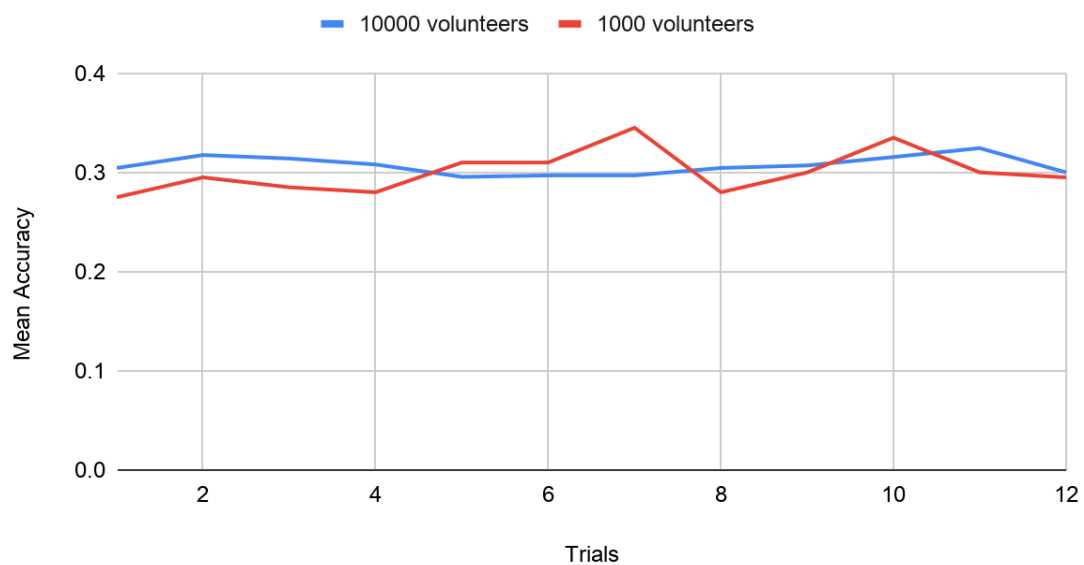


Figure 8. Experiment 1.4 results graph

Overall, it was found that the random forest bagging classifier was the highest performing prediction model when trying to predict the volunteer's performance. Furthermore, avoiding the use of random forest algorithm parameters of a low n-estimators value and a high max depth value and training the datasets with more trials will yield better results.

The application will employ event recommendations for users to volunteer in. This will improve volunteer and supervisor experience in multiple ways: 1) Volunteer exposure to suitable events will increase, allowing volunteers to serve in more events, and 2) The supervisors' events will be advertised to volunteers who are more interested and experienced, increasing the quality of volunteer service. In this last experiment, the goal was to accurately recommend a category of the event to a volunteer-based on the events they had previously volunteered in, profile information, and event information.

As performed in the previous two experiments, the machine learning algorithms are trained using previous user data. Both previous users' event data and the user's current data will be used in order to make the prediction. The Scikit-learn machine learning library in Python allows usage of linear regression, polynomial regression, support vector machine, random forest classifier (n-estimators = 30 and max_depth = 30), k-nearest neighbor classifier (n-neighbors = 3), and Gaussian naive Bayes to make predictions [13][14][15]. Each prediction model is trained with dummy data variables including the number of each specific type of event (science-oriented, computer-science-oriented, writing oriented, public-speaking-oriented, math-oriented, and engineering-oriented) that the volunteer has served in. The generated recommendation is related to the number of events of each type that the volunteer has attended.

Within this experiment, multiple experiments with multiple trials were run. Each experiment alters one of each: algorithm parameters, volunteer feature sets, and training data sizes. These experiments will allow us to determine which machine learning algorithms will predict recommendations for users the best.

In experiment 1, the only data taken into account was the amount of each type of event the volunteer participated in. The type of event that the volunteer served most in was the recommended event. Figure 9 illustrates the results of experiment 1. The two highest-scoring prediction models were the support vector machine and the random forest classifier, and the random forest classifier scored slightly higher than the support vector machine on certain accounts. Linear regression was the lowest scoring prediction model.

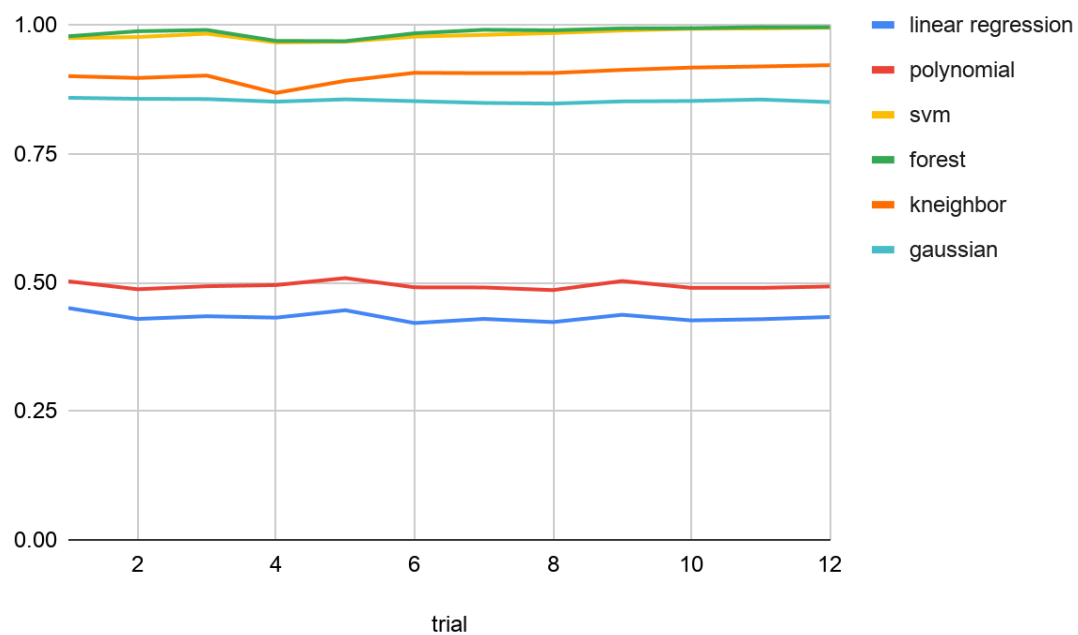


Figure 9. Experiment 2.1 results graph

During experiment 2, the highest-scoring model, the random forest classifier's two model parameters, n-estimators and max-depth, were altered. Figure 10 shows the results of the second experiment. Overall, the different groups scored similarly, with the exception of n-estimators = 5 and max depth = 5, and n-estimators = 30 and max depth = 5. The highest scoring was n-estimators = 30 and max depth = 30, which is the same model parameters used in experiment 1.

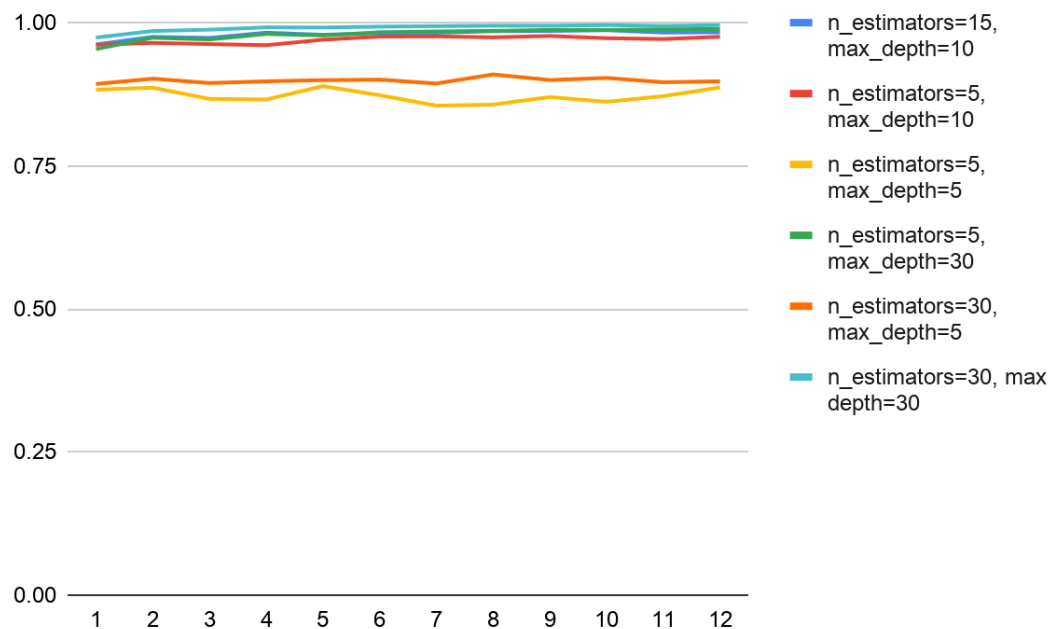


Figure 10. Experiment 2.2 results graph

In experiment 3, the impact of sample size on the result was once again tested. The highest scoring prediction model, the random forest classifier, was tested in this experiment. Figure 11 shows the unanticipated results of the experiment. With the introduction of more trials, the machine learning algorithm performed worse than with fewer trials.

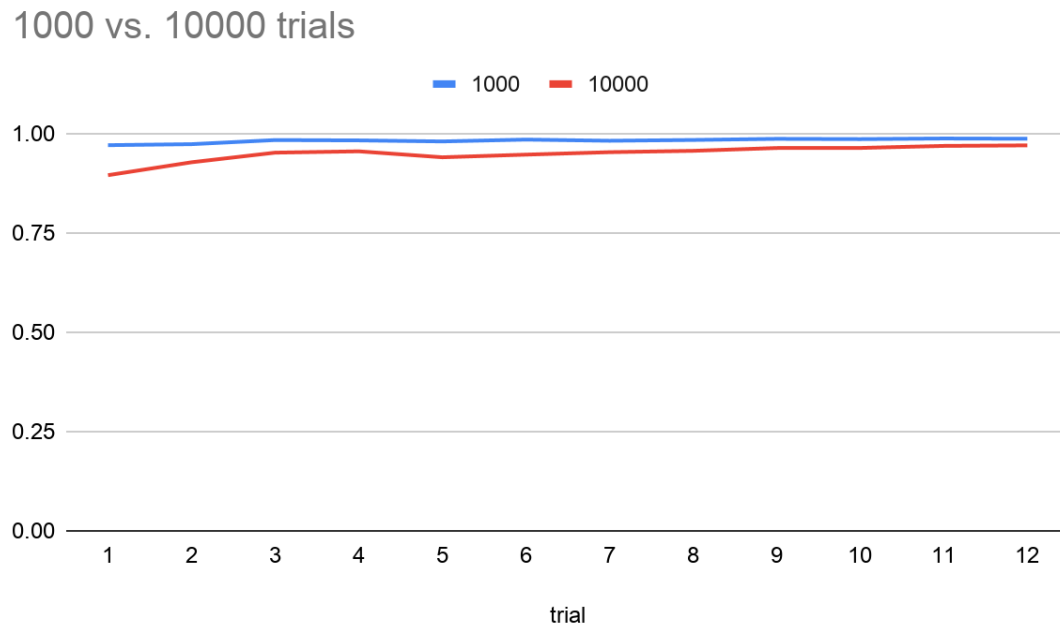


Figure 11. Experiment 2.3 Results Graph

5. RELATED WORK

Stukas, A. et al [10] propose an index that calculates the volunteer's matches to one or more of six motivational categories identified in previous research. The authors of the paper prove that the use of this index will predict volunteer outcomes better than volunteer motives and affordances alone can predict. Furthermore, the authors find the less structured organizational contexts are, the greater the total matches' effect may be. In contrast, this paper uses machine learning algorithms to predict multiple outcomes, one of which is volunteer performance based on traits like age, gender, interests, etc. Rather than categorizing volunteers, this study connects the volunteer and his/her information directly to volunteer performance. Connections between volunteer performance and organizational structures are not covered in this paper.

Finkelstein, M. et al [11] predicted how volunteerism is affected by multiple factors, such as initial motives, personality, and role identity. The paper found that a volunteer's identity and expectations were the greatest predictors of the length of volunteering. Instead of searching for the basis of volunteerism, this paper aims to find patterns between a volunteer's disposition and the types of events volunteers may want or should volunteer in. These patterns will be used to predict events to recommend to the volunteer or the effectiveness of volunteering in an event to the volunteer's final goal.

Davis, M et al [12] explores how volunteer behavior is influenced by subjective experiences during volunteering and preceding experiences before volunteering. Two-hundred-thirty-eight real volunteers over nine organizations were studied over the course of a year. Findings indicate that, consistent with their model, volunteer involvement is influenced by the satisfaction felt from volunteering, but the continuance of volunteerism is not. This study uses dummy data and

aims to predict how both objective and subjective factors influence the performance of the volunteers. These factors may include age, gender, and interests.

6. CONCLUSION AND FUTURE WORK

Volunteering events are common occurrences, providing opportunities for individuals to benefit the community. By finding the best machine learning algorithms to predict the two queries introduced previously, volunteers will be able to have a more satisfying and beneficial volunteer experience, while supervisors will be more content with the outcomes of their volunteers. The predictions of these algorithms are implemented as user-viewable features within the application. By viewing these predictions, volunteers will be able to have a better understanding of what events to volunteer in and how they will perform. As the bridge between volunteers and supervisors, this application will both facilitate the communication between volunteers and supervisors and improve volunteer experience.

Since the data used in this project was dummy data that was generated by code, there may be some inconsistencies between the dummy data and real volunteer data. Although the dummy data was generated to correlate between interests, performance, and event participation, real humans are often much more complicated. Future research will examine further connections between real humans and performance and event recommendations. With real people using the application, future research will prove to have more accurate results.

REFERENCES

- [1] Van Willigen, Marieke. "Differential benefits of volunteering across the life course." *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences* 55.5 (2000): S308-S318.
- [2] Morrow-Howell, Nancy, Song-Iee Hong, and Fengyan Tang. "Who benefits from volunteering? Variations in perceived benefits." *The Gerontologist* 49.1 (2009): 91-102.
- [3] Wilson, John. "Volunteering." *Annual review of sociology* 26.1 (2000): 215-240.
- [4] Michie, Donald, David J. Spiegelhalter, and C. C. Taylor. "Machine learning." *Neural and Statistical Classification* 13.1994 (1994): 1-298.
- [5] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *the Journal of Machine Learning Research* 12 (2011): 2825-2830.
- [6] Bracha, Gilad. *The Dart programming language*. Addison-Wesley Professional, 2015.
- [7] Wu, Wenhao. "React Native vs Flutter, Cross-platforms mobile application frameworks." (2018).
- [8] Moroney, Laurence, Moroney, and Anglin. *Definitive Guide to Firebase*. Apress, 2017.
- [9] Peng, Dunlu, Lidong Cao, and Wenjie Xu. "Using JSON for data exchanging in web service applications." *Journal of Computational Information Systems* 7.16 (2011): 5883-5890.
- [10] Stukas, A. A., Worth, K. A., Clary, E. G., & Snyder, M. (2009). The Matching of Motivations to Affordances in the Volunteer Environment: An Index for Assessing the Impact of Multiple Matches on Volunteer Outcomes. *Nonprofit and Voluntary Sector Quarterly*, 38(1), 5–28. <https://doi.org/10.1177/0899764008314810>
- [11] Finkelstein, Marcia A., Louis A. Penner, and Michael T. Brannick. "Motive, role identity, and prosocial personality as predictors of volunteer activity." *Social Behavior and Personality: an international journal* 33.4 (2005): 403-418.
- [12] Davis, M. H., Hall, J. A., & Meyer, M. (2003). The First Year: Influences on the Satisfaction, Involvement, and Persistence of New Community Volunteers. *Personality and Social Psychology Bulletin*, 29(2), 248–260. <https://doi.org/10.1177/0146167202239050>

- [13] Segal, Mark R. "Machine learning benchmarks and random forest regression." (2004).
- [14] Bunea, Florentina, Alexandre B. Tsybakov, and Marten H. Wegkamp. "Aggregation for Gaussian regression." *The Annals of Statistics* 35.4 (2007): 1674-1697.
- [15] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." *Emerging artificial intelligence applications in computer engineering* 160.1 (2007): 3-24.