

ENHANCING CLINICAL TRIAL ACCESS AND DIVERSITY: AN INNOVATIVE APP FOR PERSONALIZED TRIAL RECOMMENDATIONS AND EQUITABLE HEALTHCARE PARTICIPATION

Jasmine Xing¹ and Ang Li²

¹Buckingham Browne and Nichols School, 80 Gerrys Landing Road,
Cambridge, MA 02138

²Computer Science Department, California State Polytechnic University,
Pomona, CA 91768

ABSTRACT

This project focuses on creating an app to recommend clinical trials to users based on their profiles, improving access and diversity in trial participation. It addresses disparities in trial representation and aims to provide easy access to trial information. The app utilizes web scraping, a recommendation system, and a saving mechanism for trials. An experiment evaluates the recommendation system's accuracy, highlighting the challenges of ethnicity clustering. The app aims to promote equitable healthcare access and expand participation in clinical trials.

Shortcomings of existing solutions include limited scope, such as being specific to cancer trials or certain patient groups. In contrast, the app aims to provide recommendations for a broader range of diseases and profiles, improving accessibility and diversity.

Future improvements include refining the recommendation model, increasing the number of clusters, adding location-based recommendations, and expanding the trial database. The ultimate goal is to enhance the diversity of trial participants and improve healthcare equity.

KEYWORDS

Clinical Trial Recommendations, Healthcare, Trial Participation, Personalized Healthcare Access

1. INTRODUCTION

Researchers conduct clinical trials to determine the effects of the use of a new treatment, such as a drug. They observe a group of people over the course of time and see how they change, whether it's physically, mentally, or in another way. Clinical trials often employ new treatments and drugs that have the potential to cure people and change their lives [3]. According to the National Institutes of Health, trials in the past have often "relied almost exclusively on White male study participants" [1]. According to the FDA's report on clinical trial participation demographics, from 2015 and 2019, 76% of the participants in clinical trials were white, with 11% being Asian, 7% Black or African American, 1% being American Indian or Alaska Native, and 5% being other

[4]. However, the demographic of the US is 75.5% white, 13.6% Black or African American, 1.3% American Indian or Alaska Native, and 6.3% Asian [5]. There are disparities between which groups are represented in clinical trials, especially in representation of African Americans in clinical trials.

Diversity is critical to the results of these studies, because different groups of people can have different responses to the same medical treatment [2]. Many factors could affect how people respond to a treatment. For example, the presence of different cultures could affect a participant's lifestyle and their experiences. Other factors that are important when considering diversity are "age, biological sex, pregnancy status, unhealthy behaviors, health-promoting behaviors, environmental conditions, genetic variation and geographic ancestry, and underlying medical problems or presence of comorbidities" [1]. When clinical trial participants differ in all of those factors, it gives the researchers more insight on what factors the subject of their study might interact with [2].

When a group is not represented in a trial, the results can be biased against these groups. Researchers will have a less comprehensive understanding of the treatment they're testing in a trial, due to lack of data in such groups. This will in turn decrease equity in access to potentially innovative treatments.

Barriers to diversity in clinical trials could be due to a plethora of reasons, such as lack of access because of socioeconomic status, distrust in the medical system, or simply not being aware of what is available. A prime example of a cause for distrust is the Tuskegee Syphilis study, which nonconsensually used black men in a study on many diseases [6].

The "Evaluation of an artificial intelligence clinical trial matching system in Australian lung cancer patients" study created an AI model to match "Australian lung cancer patients" with eligible clinical trials, achieving 91.6% accuracy. However, it's limited to Australian patients and has a relatively small dataset. Our project aimed to improve on this by providing recommendations for all diseases and profiles, ensuring broader access to trials, and using larger and more diverse datasets for better generalizability.

The discussed study by the Leukemia & Lymphoma Society's Clinical Trial Support Center aimed to reduce barriers to clinical trial access for 906 cancer patients, offering guidance on trial registration. However, it relied on medical professionals and focused solely on cancer trials. Our project improved on this by providing broader, non-cancer-specific clinical trial recommendations accessible to a wider audience without the need for medical assistance.

The mentioned study aimed to create a machine learning model to match patients with clinical trials using 891 cancer trials from ClinicalTrials.gov. It achieved a 0.90 precision score but was limited to cancer trials, primarily benefiting pregnant or HIV patients. Our project improved on this by offering a broader recommendation system encompassing various diseases and profiles, increasing accessibility to a wider range of clinical trials.

My proposed solution is the creation of an app that recommends people to clinical trials they are eligible for, which also has a filtering system and saving system that makes it easier to access clinical trial data. Many people are unaware that clinical trials exist, especially if they're not near larger medical centers. Doctors in large medical centers are often involved in research and clinical trials, so they're able to inform their patients of any new clinical trials that they could benefit from. However, in more rural areas where there are less large medical centers, doctors might not be informed of new clinical trials, so they can't spread that information to their patients. My app provides an easy, accessible way for people to look at trials around them and get recommendations, instead of only giving them the ability to filter through the trials themselves

and look for ones they're eligible for. The FDA has mandated diversity in clinical trials and created clinicaltrials.gov [12], but many people are still unaware of the site and how to find clinical trials. The site has an overwhelming amount of trials on it, making it hard for people to choose a trial to enroll in. By creating recommendations based on the user's data and profile, the user has less of a need to scroll through endless trials, wondering what to do with all of them. The app also offers the user the option to save a trial into the user's "My Trials" tab, which allows them to keep track of trials more easily. It creates a centralized area that recommends users trials, allows users to browse through and filter trials, and save them.

The program features a crucial machine learning recommendation system that uses an unsupervised K Means algorithm to sort trials into five clusters based on user profiles [7]. The accuracy of this algorithm is essential as it influences user engagement. An experiment was conducted to assess the accuracy of these recommendations by manually altering patient profiles and recording the accuracy of trial recommendations. The experiment was repeated five times, with each run generating five recommendations. The highest accuracy observed was 0.4, indicating that 2 out of 5 recommended trials matched the user's profile. While age and gender recommendations were generally accurate, ethnicity recommendations were less so, possibly due to the complexity of clustering ethnicities [8]. However, since ethnicity is not a strict eligibility criterion, users can still participate in recommended trials. Analyzing accuracy in relation to each characteristic can guide model improvement. Another experiment with a fixed patient profile was conducted similarly, and it revealed that age and gender recommendations were consistently 100% accurate, while ethnicity recommendations mirrored the overall accuracy. The model excelled in clustering ages and genders, likely due to fewer gender categories, while the complexity of ethnicity categories posed challenges.

2. CHALLENGES

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

2.1. Web Scraping the Data

One major component of the app is web scraping the data. The data is going to be scraped from sites, like the UCLA Clinical Trials site that lists all of their ongoing and future trials [9]. Each of these trials have a focus, which is the main focus of the study, like a disease being studied. However, each trial has a different name for focuses that are very similar to each other, meaning there's no standardized way to list focuses. This could be fixed by using a natural language processor to standardize the focus. There are also various formats that the age range of eligibility for each trial is displayed, meaning the web scraping needs to cover all those formats. This could be fixed by adding methods to standardize the most common formats to one format, for internal processing.

2.2. The Recommendation System

Another important component of the app is the recommendation system. This recommendation system will use a machine learning algorithm to classify different trials into different groups to be recommended for users of the app. Thus, the system needs to get all the trials' data it needs to make an adequate and accurate prediction. Thus, the system will be trained with over 4,000 trials spanning many different focuses, instead of limiting it to one type of disease. It should take in the important information that defines each trial, like its location, age, gender, and more. It could use K-Means to split the trials into clusters and recommend different users trials from different clusters.

2.3. Retrieving and Saving the User's Saved Trials

An important component of the app is retrieving and saving the user's saved trials. Each user needs to have certain saved trials saved with their user ID [10]. Each user's information is stored in a Firebase database. The user's saved trials will be saved under a saved trials tab with their user ID. They will be displayed in a separate page in a StreamBuilder to allow it to update regularly, so new trials being added will show up. They will be displayed and be able to be added, deleted, and opened up to see more information about each trial. Thus, the app will be convenient for the user to track their trials.

3. SOLUTION

This app has a few main components. When a user first opens the app, they will see a splash screen and then the authentication screen, which asks them to either login or signup. Their data is sent to a larger database, which Firebase was used for [11]. Firebase is very efficient and quick in updating when a user sends a Https request to change their data [13]. The code for authentication is in Flutter, coded in a dart file. After the user creates an account or logs in, they're brought to the home page, which has information about clinical trials and recommendations based on the user's information. There are also three buttons on the bottom of the screen, which are labeled "Find Trials," "My Trials," and "My Info." When the user clicks on the "My Info" button, they open another screen. This screen displays their information, including ethnicity, age, and gender. This screen gives the user the opportunity to change this information, logout, or delete their account. The "Find Trials" button on the home page opens a screen where users are able to scroll through a database of trials and filter through them by any information. The database of trials are stored on a server run by AWS. The server code is in Python and is permanently run in AWS. The "My Trials" button opens a screen that allows the user to open a list of trials they've saved. The last component is the recommendation system, which is kept on the home page, and recommends the user trials that fit their profile, based on the information they keep in their "My Info" tab.

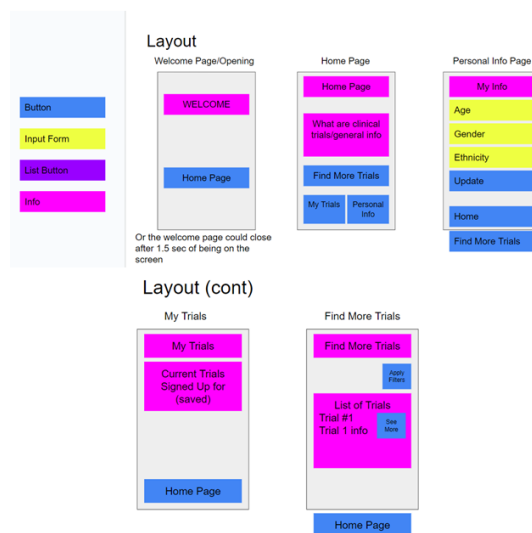


Figure 1. Overview of the solution

The authentication component of the app was created using Firebase's authentication system. This system uses the user's email and password to authenticate the user. It then stores each user's information along with their email and password. Updating a user's information in Firebase

updates it in realtime on the app, and vice versa. Thus, Firebase is quick and efficient for the authentication component.

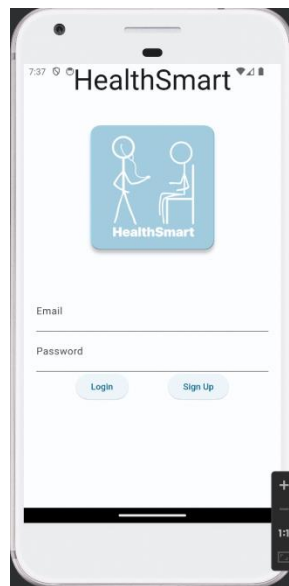


Figure 2. Home page

```
void onPressedLoginButton() async {
  if(_formKey.currentState!.validate()){
    try {
      String email = emailController.text;
      String password = passwordController.text;
      await FirebaseAuth.instance.signInWithEmailAndPassword(
        email: email, password: password);
      navigateToHomePage();
    }
    on FirebaseAuthException catch(e){
      if(e.code == 'user-not-found'){
        showSnackBar('That Email is not registered');
      }
      else if(e.code == 'wrong-password'){
        showSnackBar("Wrong password.");
      }
      else{
        print(e.code);
        showSnackBar("Can't connect, please try again later.");
      }
    }
  }
}
```

Figure 3. Screenshot of code 1

The method on Pressed Login Button runs when a user presses the “Log In” button on the authentication screen. This button is pressed after a user inputs their email and password. This method gets the email and password that the user entered from the loginForm, which sends it to a TextEditingController that is called in this method. It then stores these values as email and password, respectively. These variables are fed into the function that Firebase’s Authentication system provides to log into an account. Then it brings the user to their home page. If the Firebase Authentication system catches an error, it will return the kind of error it finds. If a user has not created an account, the system will catch that as user-not-found. If the password entered is wrong, the system returns wrong-password. The last case is where the system does not specify what kind

of error it has run into, so the app returns a default message, “Can’t connect, please try again later.” The ultimate return when a user properly logs in, is to log into the user’s account and bring the user to their home page. When a user is logged in and wants to change their information, they should navigate to their account page, which they can do by pressing “My Info” in the home page. Then, they can change different information stored in their profile, like their ethnicity, age, and gender, which will automatically update in the database.

The filtering system has a couple different pieces to it. The app displays the trials stored in a database, which is stored in a server run by AWS. The server code is all in Python. The trials are web scraped from various clinical trial sites, like UCLA and UCI. It was not possible to web scrape the trials from the official clinicaltrials.gov website, as the code to use was hidden. Each trial is stored as a clinical trial object, which contains the attributes of a clinical trial and which participants are eligible for it, including but not limited to the age range, gender, type of patient, and more. They are all stored in a larger trials list, which has unique properties. When the list is displayed on the server, it is converted into json format, making it easier for the computer to read it when it is used. In the app, which is coded in Flutter, the server is used for the list of trials, which are subsequently displayed on the screen. The filtering system allows users to only display trials that fit their interests. They can filter by the age, gender, or focus of the study and its participants. There’s a pop-up when the user clicks the “Filter” button, which has a drop down menu for the parameters they want to filter by, and when they click the “Update Filter” button, their parameters are imputed and only the trials that fit those parameters show up.

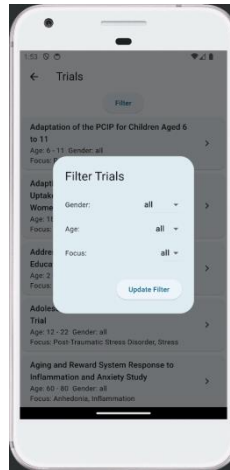


Figure 4. Filter

```

void showFilterDialog() {
    List<String> genderList = ["all", "male", "female"];
    List<String> focusList = ["all"];
    List<int> ageList = List<int>.generate(101, (i) => i);

    String genderSelected = filterData["gender"];
    String focusSelected = filterData["focus"];
    int ageSelected = filterData["age"];

    showDialog(
        context: context,
        builder: (context) {
            return StatefulBuilder(
                builder: (context, setState) {
                    return AlertDialog(
                        title: const Text("Filter Trials"),
                        content: SingleChildScrollView(
                            child: Column(
                                children: [
                                    // Gender Row
                                    Row(
                                        children: [
                                            const Text("Gender:"),
                                            const Spacer(),
                                            DropdownButton<String>(
                                                value: genderSelected,
                                                onChanged: (String? value){
                                                    genderSelected = value!;
                                                    setState((){});
                                                },
                                                items: genderList.map<DropdownMenuItem<String>>{(
                                                    String value){
                                                        return DropdownMenuItem<String>(
                                                            value: value,
                                                            child: Text(value),
                                                        ); // DropdownMenuItem
                                                    }
                                                }).toList()
                                        ],
                                    ),
                                ],
                            ),
                        ),
                    ),
                ),
            ),
        ),
    );
}

```

Figure 5. Screenshot of code 2

The show Filter Dialog opens when a user wants to filter trials. It has a list of the possible values for each parameter. For example, for genders, it has “all,” “female,” and “male.” There are three parameters: gender, focus, and age. Then, there are variables that represent the values that the current display is showing for. By default, they’re set to filter Data [parameter_name]. These are values set at the top of the _State class. At default, the age is set to 0, the gender set to “all,” and the focus set to “all” as well. Then, the method shows the dialog and returns an Alert Dialog, which is where the user changes the parameters they want to filter by. It starts with gender, which is a dropdown button. It changes the gender Selected (initiated at the top of the method) to whatever the user chooses from the list of genders (genderList). It repeats these same steps for both age and focus, with their respective lists and selected values. At the end, there’s an “Update Filter” button. When pressed, it changes the values of the parameters in the map created outside of the method. This changes the list of trials displayed on the user’s page to the list of trials that fit the user’s values for the parameters.

The saving system of the app allows users to have easy access to specific trials they choose. If a user is recommended for a trial and they want to save it to their own list so they can see it afterwards, they can. It uses a TrialsListView class to display all the saved trials. The saved trials are saved on Firebase under the user’s account ID so they can be easily accessed for display. From the recommended trials and filtered trials, users can also press the save button which will save them to the same place.

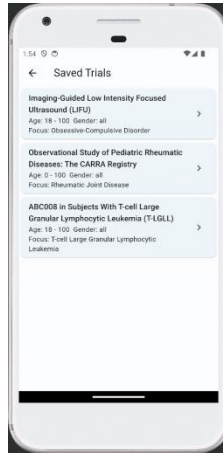


Figure 6. Saved trials

```

Widget savedTrialsBody() {
  String uid = FirebaseAuth.Instance.currentUser!.uid;
  String path = "savedTrials/$uid";

  return StreamBuilder(
    stream: FirebaseDatabase.Instance.ref(path).onValue,
    builder: (context, snapshot) {
      if (snapshot.hasError) {
        return const Text("Error");
      }
      if (snapshot.hasData) {
        if (snapshot.data == null || snapshot.data!.snapshot.value == null) {
          return Center(child:
            Text(
              "No Trials Saved",
              style: Theme.of(context).textTheme.displaySmall,
            ), // Text
          ); // Center
        }
        else {
          Map<dynamic, dynamic> mapOfSavedTrials = snapshot.data!.snapshot.value as Map<dynamic, dynamic>;
          return savedTrialsListView(mapOfSavedTrials);
        }
      }
      return const Text("loading...");
    }, // StreamBuilder
  );
}

```

Figure 7. Screenshot of code 3

This function displays all of the user's saved trials. First, the widget collects the users' ID and the path to their information. The widget returns a stream builder, which is constantly updating. All of the saved trials for an individual user are stored in Firebase, in the "savedTrials" folder under the specific user ID. If there are no trials saved, the "My Trials" page displays "No Trials Saved." If there are trials, it will load them as a savedTrialsListView. If there's an error, it will display an error. Otherwise, the screen will display "loading..." until it can fetch the trials and display them properly.

The recommendation system of the app takes in the user's information as parameters and inputs them into a K Means machine learning model, trained with the trials from the database. The code for the system is in the Python server run by AWS. K Means is a machine learning AI model that takes in a bunch of different values and splits them up into a few different clusters, and the amount of clusters is decided by the programmer.

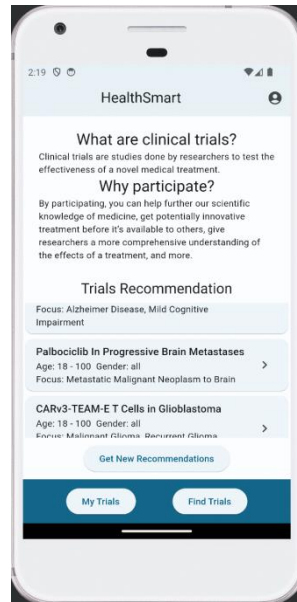


Figure 8. Q&A page

```
def get_recommendation(all_trials, age, ethnicity, gender, numRecommendations = 5):
    ethnicity_mapping, gender_mapping, kmeans, cluster_name_mapping = prep_data(all_trials)
    encoded_ethnicity = ethnicity_mapping[ethnicity]
    encoded_gender = gender_mapping[gender]
    input_data = np.array([[age, age, encoded_ethnicity, encoded_gender]])
    cluster_id = kmeans.predict(input_data)[0]
    recommended_names = cluster_name_mapping[cluster_id]
    if len(recommended_names) == 0:
        return []
    size = min(len(recommended_names), numRecommendations)
    return np.random.choice(recommended_names, size)
```

Figure 9. Screenshot of code 4

The `get_recommendation` function takes in the full list of trials being used for the recommendation system. It also takes in the user's information, like their age, ethnicity, and gender. It also has a parameter called `numRecommendations` which is preset to 5, meaning the function returns 5 different trials as a recommendation for the user. The function first creates the model by calling the `prep_data` function. This will create a K Means model trained on the list of trials inputted into this function. It uses this model and the user information inputted into this function to create a prediction, which is a cluster. After, it will randomly choose five of the trials in the cluster to return and recommend to the user. They are returned as a numpy array. These functions are called on the server in a new page. The app sends a request to the server, calling this function and returns the trials to be displayed in the same way they were for the filtering system.

4. EXPERIMENT

4.1. Experiment 1

An important component of the program is the machine learning recommendation system. Currently, it runs an unsupervised K Means algorithm that is trained to sort trials into five clusters. The accuracy of the algorithm is very important, because they need to fit the user's

profile. They show up on the home page and if a user sees trials recommended for them that truly fit their profile, they'll be more inclined to participate and continue using the app.

The experiment to test the recommendation system will be evaluating the accuracy of the recommendations. The profile information for each patient will be changed manually and then the recommendations and their accuracies will be recorded. If the recommended trials fit the user's profile, namely their ethnicity, age, and gender, it will be recorded as accurate. If something in the trial doesn't fit the user's profile, for example the patient is female and the recommended trial is for males only, it's recorded as inaccurate. For this experiment, all of the user's information (age, gender, and ethnicity) will be specified and kept at a random constant for the experiment.

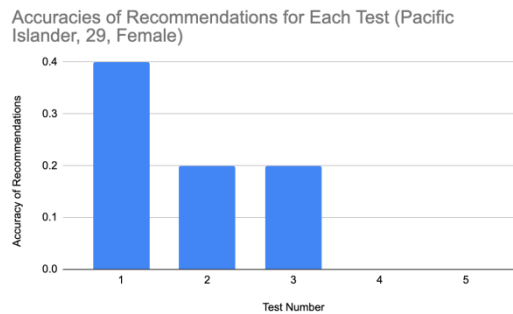


Figure 10. Figure of experiment 1

Overall, the experiment was run 5 times and the recommendations were recorded for each test run. There were five recommendations per test run. The percentage of those five trial recommendations per test run that exactly match the user's profile is the accuracy of the recommendations. They were then graphed against the test number. The highest accuracy was 0.4, meaning 2 out of the 5 recommended trials fully fit the user's profile. Some recommendations were unfitting, while some others were. Many of the recommendations were accurate with respect to the age and gender eligibility, however they were inaccurate when it came to ethnicity. This could mean that the model was unable to split the ethnicities into clear clusters. However, ethnicity is not a required eligibility to participate in a trial, even though it increases the trial diversity, so the user can still participate in the recommended trials, regardless of ethnicity.

4.2. Experiment 2

The accuracy of the recommendations in respect to age, gender, and ethnicity are also very crucial. Seeing the specified accuracies in relation to each characteristic can help further development of the model by seeing where it needs improvement.

This experiment will be conducted very similarly to the previous one. Different characteristics of the patient will be specified, like their age, and the age, gender, and ethnicity of the recommended trials will be recorded. They will each be compared to the user's profile for accuracy, and the percentage of the five recommended trials per test run that are accurate will be recorded and graphed. For this test, the patient will be 29, a female, and Pacific Islander. Each attribute will have a graph that portrays the accuracy of that specified attribute. If one attribute is less accurate in the recommendations, the model can be fixed to work better in that aspect.

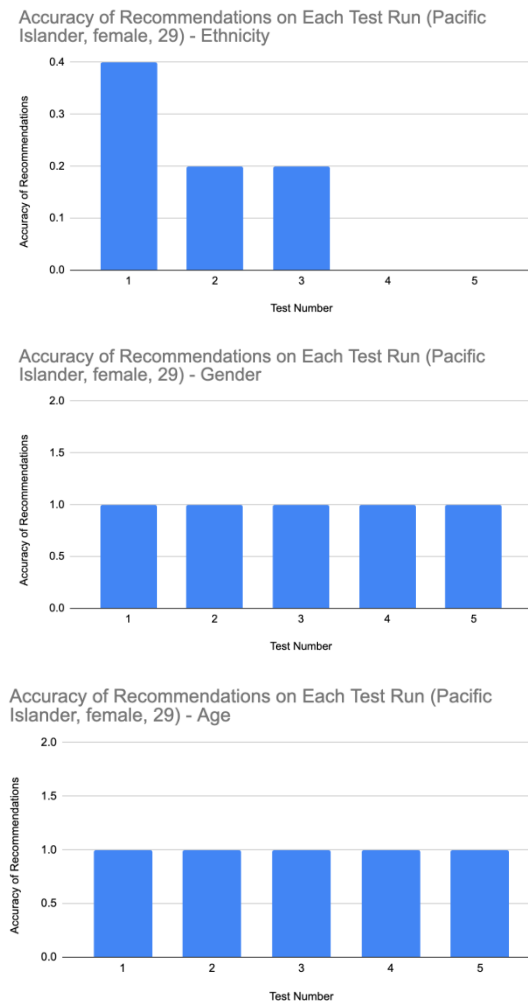


Figure 11. Figure of experiment 2

Overall, this experiment was run five times, similarly to the other one. The age and gender graphs show that the recommendations are 100% accurate in respect to those two attributes. However, the ethnicity mirrored the overall accuracy of each of the respective tests. The first test gave 2 out of 5 accurate recommendations, in respect to ethnicity, which is the same as the accuracy overall. Thus, ethnicity seems to be the reason for any inaccurate recommendations. The age is 100% accurate for each recommended trial, meaning that the model does a good job in clustering the ages. The same goes for gender, as the gender has 100% accuracy as well. This could be because there are only 3 different qualifications for gender at the moment: females, males, and all (both male and female). Thus, the five clusters can be more accurate to the respective genders. There are many more ethnicities, so they may be harder to break into clusters.

5. RELATED WORK

A study published by the International American Medical Informatics Association in the Oxford University Press addresses the same problem by creating an artificial intelligence (AI) model that matches “Australian lung cancer patients” to clinical trials they’re eligible for [14].

The researchers named Marliese Alexander, Benjamin Solomon ... use a variety of clinical trials in the government database of trials. They also took the clinical data of 102 patients with lung cancer to use in the study. The data was run through a model and then the model predicted the eligibility of a patient for a certain trial. The overall accuracy was 91.6%, which is relatively high. However, this model was only created for “Australian lung cancer patients”, not for all patients around the world. This does not help expand access to trials to other types of patients. There are a plethora of other studies that need more diverse participants, meaning this model may not work on them. My app provides recommendations for everyone, no matter their disease or profile. One hundred and two patients is also a relatively low testing data set.

A study published in the American Society of Clinical Oncology discusses an approach used by The Leukemia & Lymphoma Society’s Clinical Trial Support Center (CTSC) to lower the barriers to accessing clinical trials. The study took 906 patients and analyzed their eligibility for various cancer clinical trials. The CTSC also provided guidance to many patients on how to register and find clinical trials, increasing accessibility to these trials. However, this approach has limited scope, as it requires a medical professional’s aid and is specific to clinical trials having to do with cancer. My app is able to help more people find more clinical trials without needing a medical professional, and can find trials not in cancer.

A study in the Journal of the American Medical Informatics Association, Volume 24, Issue 4, July 2017 created a machine learning model to aid in matching clinical trials to patients [15]. These researchers used 891 cancer clinical trials from ClinicalTrials.gov. Then, they used regular expressions to analyze the clinical trial eligibility requirements, which they then trained a support vector machine learning model on. Overall, the method had a 0.90 precision score. However this model mainly classifies patients in relation to clinical trials for cancer. The model also works particularly for patients who are pregnant or have HIV. Thus, its scope is somewhat limited, and only has the potential to help certain people. There are also different trials that are not cancer-focused.

6. CONCLUSIONS

If given more time, I would continue to work on improving the accuracy of the model. At the moment, the model only splits the trials into 5 different clusters, which may not be enough to cover the large diversity of the trials. I would do this by adding more clusters and then testing to see if the accuracy of the system is better than before, to find the most optimal model. At the moment, the recommendation system only uses the user’s age, gender, and ethnicity to pair them with the best-matched cluster of trials. It would be great to also add the user’s location, so the recommended trials are more convenient for them to get to. This would consist of adding a location attribute to each person and adding that attribute to the AI recommendation system. I also hope to add more trials to the database it is pulling from, which will make the list of trials more comprehensive. I would do this by finding more websites to scrape trials from.

Overall, I hope that this app can help expose more people to the potentially innovative treatments that are being used in clinical trials. I hope this can increase the diversity of participants in clinical trials. This will help scientists gain more comprehensive results from the trials, which will be applicable to a diverse population therefore improve equity in access to innovative healthcare solutions.

REFERENCES

- [1] Bibbins-Domingo, Kirsten, Alex Helman, and Victor J. Dzau. "The imperative for diversity and inclusion in clinical trials and health research participation." *Jama* 327.23 (2022): 2283-2284.
- [2] Clark, Luther T., et al. "Increasing diversity in clinical trials: overcoming critical barriers." *Current problems in cardiology* 44.5 (2019): 148-172.
- [3] Li, Guowei, et al. "An introduction to multiplicity issues in clinical trials: the what, why, when and how." *International journal of epidemiology* 46.2 (2017): 746-755.
- [4] da Cruz, Lyndon, et al. "Five-year safety and performance results from the Argus II retinal prosthesis system clinical trial." *Ophthalmology* 123.10 (2016): 2248-2254.
- [5] Ratcliffe, Michael, et al. "Defining rural at the US Census Bureau." *American community survey and geography brief* 1.8 (2016): 1-8.
- [6] Centers for Disease Control and Prevention. "Tuskegee Study—Timeline—CDC-NCHHSTP." Centers for Disease Control and Prevention (2021).
- [7] McKenney, Nampeo R., and Claudette E. Bennett. "Issues regarding data on race and ethnicity: the Census Bureau experience." *Public health reports* 109.1 (1994): 16.
- [8] Kaneshiro, Bliss, et al. "The challenges of collecting data on race and ethnicity in a diverse, multiethnic state." *Hawaii medical journal* 70.8 (2011): 168.
- [9] Alexander, Marliese, et al. "Evaluation of an artificial intelligence clinical trial matching system in Australian lung cancer patients." *JAMIA open* 3.2 (2020): 209-215.
- [10] Sae-Hau, Maria, et al. "Overcoming barriers to clinical trial participation: Outcomes of a national clinical trial matching and navigation service for patients with a blood cancer." *JCO Oncology Practice* 17.12 (2021): e1866-e1878.
- [11] Zhang, Kevin, and Dina Demner-Fushman. "Automated classification of eligibility criteria in clinical trials to facilitate patient-trial matching for specific patient populations." *Journal of the American Medical Informatics Association* 24.4 (2017): 781-787.
- [12] US Food and Drug Administration. "FDA Takes Important Steps to Increase Racial and Ethnic Diversity in Clinical Trials." (2022): 245-255.
- [13] Moroney, Laurence, and Laurence Moroney. "The firebase realtime database." *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform* (2017): 51-71.
- [14] Mussone, Lorenzo, Andrea Ferrari, and Marcello Oneta. "An analysis of urban collisions using an artificial intelligence model." *Accident Analysis & Prevention* 31.6 (1999): 705-718.
- [15] Vartak, Manasi, et al. "ModelDB: a system for machine learning model management." *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 2016.