When approaching the problem predicting future adopted users, the number one issue was isolating the dataset to not only include users that fit not the explicitly stated requirements but to extrapolate and do further curating of the data set to ensure we are only building a model from users that make intuitive sense to include. The complete curating process includes:

- Removing users from the data set that have not logged in since creating the account
- Removing users that have an account created within 7 days of the last pull
    - I did this because I am working under the assumption that the latest date a user had logged into the product is the date that the data was pulled as the acquisition date was not explicitly given in this problem.

From here, we begin the predictive model build. I started by importing the engagement CSV file and converting the time_stamp column into a datetime format. I then created a new column in the dataframe that records only the date of the login and not the time of day. Finally, I created a mapping dictionary that loops through engagement csv in these steps:

- Creates a temporary dataframe that stores all the login of a particular user
- Records the first login and the last login date timestamps
- A set of 2 conditional statements:
    - If the latest login timestamp is less than 7 days of the first login timestamp then the map will record the user as 0 (not adopted)
    - If the number of days of logins are less than 3 the map will record the user as 0 (not adopted)
- If both conditions are not satisfied, then I concluded that the user is an adopted user and map it as such

Now I moved onto the CSV holding the feature information to build the predictive model with. Importing the CSV file, I start by dropping all users that had not logged in. This was done by dropping all rows with NaN values in the last login column. The next thing I did was to convert both columns containing datetime info into datetime columns. I then used the map I created earlier to create our labels, labeled as “adopted”. Looking at the user invitation column and seeing values missing, I worked under the assumption that the missing values were due to the user creating an account on the product without an invitation and would therefore have no information available, so I set all the missing values to 0. To solve the 2nd non explicit requirement of needing to remove accounts younger than 7 days since account creation, I used np.where to create a conditional column populating it with values of 1 if account creation date was earlier than the cutoff date and 0 otherwise and dropped all rows that did not satisfy the requirement. Finally, I set the user ids as the index and dropped all the non-pertinent columns for building the model: creation_time, last_session_creation_time, requirement, name, and email

Moving onto building the predictive model itself, I followed the standard procedure of splitting the dataset into training, validation and test sets. I also built a dendrogram to take a quick look at what features might be the most important for determining adoption rate. It indicated that the organization group and the creation source were the closest correlated values with adoption rate. Tweaking the hyperparameters, I built a Random Forest Regression model after recombining the validation and test set and fit it to our data. Finally, we get:

- R^2: -0.22922049742990414
- Mean Squared Error: 0.2295234997259423

A bar graph showing feature importance of the random forest regressor was also plotted. Based off this barplot, we see the most important feature of adoption rate was which organization the user belonged to.