

SES Research:

Aim: Determine which ML algorithm to use for date.io

Proposed model before research: intuitively suggested using a clustering Algorithm as it makes sense to cluster people into rooms based on the answers they have selected.

“A cluster refers to a collection of data points aggregated together because of certain similarities”

The technique I decided to follow is to start at the root of ML and work my way down to an algorithm that best suits our use case.

Machine learning models cheat sheet

Supervised learning	Unsupervised learning	Semi-supervised learning	Reinforcement learning
<p>Data scientists provide input, output and feedback to build model (as the definition)</p> <p>EXAMPLE ALGORITHMS:</p> <p>Linear regressions</p> <ul style="list-style-type: none">■ sales forecasting■ risk assessment <p>Support vector machines</p> <ul style="list-style-type: none">■ image classification■ financial performance comparison <p>Decision tree</p> <ul style="list-style-type: none">■ predictive analytics■ pricing	<p>Use deep learning to arrive at conclusions and patterns through unlabeled training data.</p> <p>EXAMPLE ALGORITHMS:</p> <p>Apriori</p> <ul style="list-style-type: none">■ sales functions■ word associations■ searcher <p>K-means clustering</p> <ul style="list-style-type: none">■ performance monitoring■ searcher intent	<p>Builds a model through a mix of labeled and unlabeled data, a set of categories, suggestions and example labels.</p> <p>EXAMPLE ALGORITHMS:</p> <p>Generative adversarial networks</p> <ul style="list-style-type: none">■ audio and video manipulation■ data creation <p>Self-trained Naïve Bayes classifier</p> <ul style="list-style-type: none">■ natural language processing	<p>Self-interpreting but based on a system of rewards and punishments learned through trial and error, seeking maximum reward.</p> <p>EXAMPLE ALGORITHMS:</p> <p>Q-learning</p> <ul style="list-style-type: none">■ policy creation■ consumption reduction <p>Model-based value estimation</p> <ul style="list-style-type: none">■ linear tasks■ estimating parameters

This simple cheat sheet helped me narrow down the algorithm required.

We know that Unsupervised learning is the most appropriate as we are trying to arrive conclusions based on data about a user/s.

Under Unsupervised learning, I began to research Apriori, which I had never heard of and K means clustering, which I have heard of.

K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

The '*means*' in the K-means refers to averaging of the data; that is, finding the centroid.

We will be able to apply this ML model to create a centroid as a room and match the x amount of people most closely related to that room.

An advantage of K-Means is that it is supposedly very fast with $O(n)$ complexity.

— Confirmed with Nicks research this is the way to go.

Aim: Determine which backend is best to implement K-means algorithm

The initial MERN stack may not be the best option to work with an ML algorithm

I researched K-means algorithms that work with node and found a few packages:

- <https://www.npmjs.com/package/node-kmeans>
- <https://nodepit.com/node/org.knime.base.node.mine.cluster.kmeans.ClusterNodeFactory2>

However, documentation and support is lacklustre in comparison to K-means in python.

Python backend is not difficult to setup and is relatively simple to connect to a react front end.

- Confirmed the team is happy to switch the backend to python/flask.

Aim: Find best library to move users around the room and lock them in a zone

<https://openbase.com/categories/js/best-javascript-drag-and-drop-libraries>

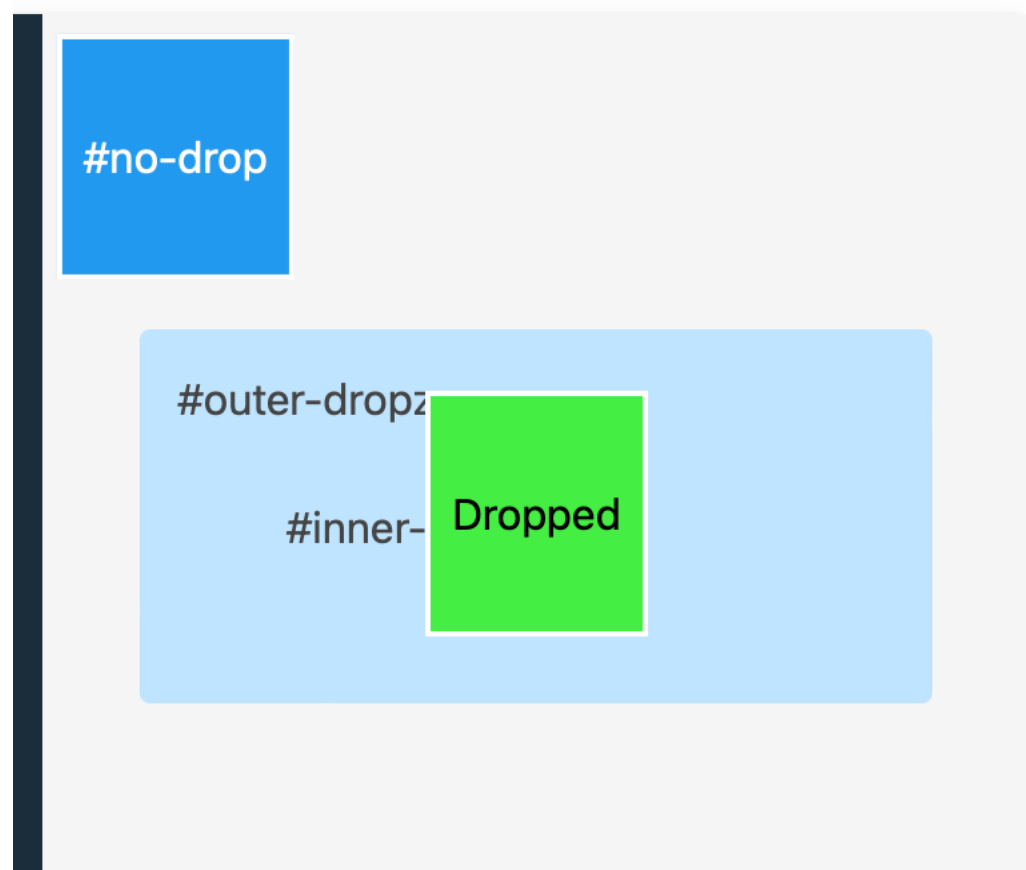
To move users around the room, I found a drag and drop library called interact.js

There were many other libraries that implemented drag and drop functionality but this one also has the zone which we will use within a room for users to connect and trigger chat.

interact.js takes a slightly different approach compared to most drag and drop libraries it gives you as much control as possible, it tries to provide a simple, flexible API that gives you all the pointer event data you'll need to move elements around.

This is in vanilla JS but I will be able to convert it into react. The implementation for the UI is fairly simple, however connecting it to users and the backend will be a challenge.

Example below:



The only issue with this, I still need to do research on is that I do know know how to make this drag and drop functionality appear in real time across multiple devices.