

# Video Game Recommendations using Machine Learning

## Darragh Fahey G00352047

### Project Description

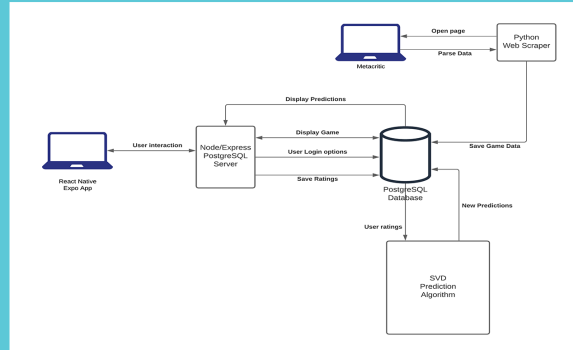
Machine Learning is becoming one of the most talked about and important topics in Computer Science and Engineering. For this reason I wanted to focus on it for my final year project.

I started to investigate recommendation systems using machine learning and found that there is already a lot of systems in place to use it the most well know of which would be for big companies like amazon recommending new products based off other users, or Netflix's tv and movie recommendations. I knew then I wanted to make one that catered to games.

For the project I used PostgreSQL for database management, React Native for front end webapp development and various python libraries for the machine learning algorithm and web scraper portion of the project.

Throughout the project I learned a lot about how to manage the various components and make sure they all work together.

### Technologies and Skills



Architecture Diagram



The front-end webapp was designed using React native and Expo. Using these two frameworks allowed me to create a webapp that would work on the web, android, and iOS without having to develop separate codebases for them

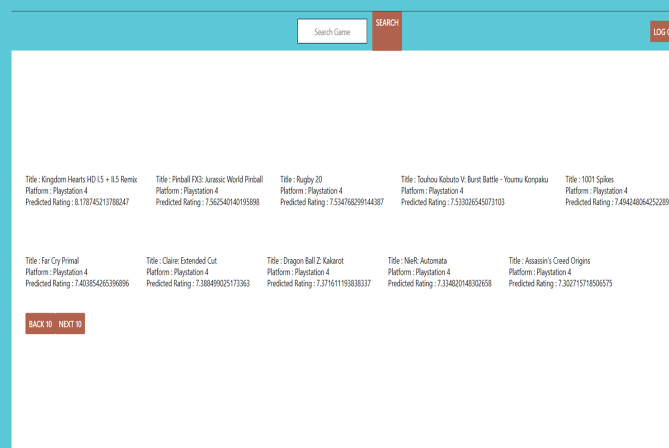


For the Machine learning algorithm and the Web Scraper I used python. For the Web Scraper I used the BeautifulSoup library, this allows the code to pull data from HTML and XML files which is then easier to parse. For the Machine Learning algorithm I used the Surprise Scikit, which allowed me to feed the relevant information into an SVD algorithm



I decided on using PostgreSQL for the database of my project as it is a well known Relational Database, and it uses a structured query language, and scales well. This means that it will continue to work well for an ever growing dataset.

### Results



Using the Surprise scikit I was able to feed user recommendations into the singular value decomposition algorithm (SVD) algorithm which made predictions for games that a user has not played or recommended.

These predictions are saved into a PostgreSQL database, which the backend server looks up using a fetch API call and displays the found information on the front end webapp.

The user can scroll through pages of ten games of personalized predicted recommendations.

If a user is not logged in they will be shown the top metacritic rated games instead of a personalized list.

A user can search individual games and see more information about them whether they are logged in or out of the system.