For the movie recommender system app I decided to use the Python programming language, since it is the most popular choice for building machine learning applications, also having the largest amount of libraries dedicated to data science and machine learning. Other available choices were R and C++, however R’s user community is way more limited compared to that of Python and C++’s tooling and libraries for machine learning are not yet mature enough.

As for Python libraries, I am currently using pandas for data manipulation and scikit-learn as the general framework for machine learning, since it is easier to manipulate exclusively tabular data (which I am working with), compared to PyTorch and Tensorflow. Tensorflow and PyTorch are typically used more in Deep Learning and Neural Networks, while scikit-learn is used more in general machine learning. However, these frameworks are not exclusive and designed to be able to integrate with each other.

There are almost no recommender systems that are standalone applications, instead many of them are integrated within movie-related apps. For example, Netflix has a built-in recommendation system (and the company itself is one of the more invested in this area of research), where the app analyzes what movies you have watched and recommends similar movies. Of course, the system is also very complex, being capable of deducing issues such as recency bias and user bias. Netflix uses Python, MySQL and AWS for its machine learning and data science components. Netflix’s recommender system cannot specifically recommend less popular movies and shows.

My application will be deployed as a web application built with React and Flask, so that it is compatible with most modern web browsers. I will use Flash instead of Django as it is more lightweight and don’t need the majority of Django’s tools and React since I am personally more familiar with it and I prefer React Hooks to Angular Components or other object based approaches. The backend will be deployed on Google Cloud instead of AWS, since I prefer the ease of use and the power of the VMs of Google Cloud to the versatility of AWS. The frontend will be deployed using Netlify.