XTrainRec

Gym Exercise Recommendation Engine

Federico Loguercio - 22 March 2019

Introduction

Gyms are booming all over the world. With an increasing portion of the population working in office jobs, more and more people choose to go to the gym both to remain healthy and work on their appearance.

Xtrain is revolutionising the way people are working out. The Xtrain app, offered for free to every Xtrain member, provides predefined workouts, curated by Xtrain's trainer team, alongside dietary recommendations. Currently, recommendation engines are in place to provide these two items to the user.

Recently, Xtrain introduced videocameras to every area of each gym belonging to the chain. The initiative began as a response to problems arising from injuries happening at the gym. Unclarity about whether a certain injury was the user's or Xtrain's fault have resulted in high legal costs for the company.

This report aims to point out possibilities to further extract value from the videocameras. In particular, the prototype for an exercise-based recommendation engine will be proposed. The videocameras allow us to know exactly what kinds of exercises are being performed by which kinds of users, such that we can extract the true workout routines of Xtrain's customers and offer suggestions to bring their workouts to the next level.

Dataset

In order to portray the potential of this prototype, a simulated dataset will be generated.

Personal data about every user is already being recorded when they sign up. This includes age, weight and gender.

Further data will eventually be gathered through **computer vision** on the video material gathered by the cameras. One more user-related datapoint will be **strength**, measured as the average of the maximum weight in kg which the user has used on the most comparable and common exercises, being bench press, squat and biceps curls. Thereafter, a **binary ratings matrix** will be constructed based on which exercises a user has actually performed in the gym. More precisely, it will be assumed that a user liked the exercise if they performed it at least two times and did not like it if they performed it only once.

Lastly, for each of the 300 exercises possible with the equipment present in the gym, the main body part trained in that exercise will be recorded.

All of these variables were simulated within reasonable value ranges. No exploratory data analysis will be performed as that would make little sense on simulated data.

Recommendation Goal

One of the main problems users face in gyms is that they **fall into the same workout-routine** and then rarely change up the exercises they do. Providing users with **predefined workouts** through Xtrain's app has been welcomed by certain users, but many users do not wish to completely change what they do at the gym. Instead, they wish to **receive recommendations for exercises which they could add to their routine**, or switch in for a similar exercise they are already doing.

This also implies that there is no real **cold start** problem in this case. We are facing two kinds of cold starts: Users who are new to going to the gym in general and users who are new to Xtrain but have gone to the gym before. The first one will be bridged by the predefined workouts which are already being recommended within the app. The second one should not be a problem at the beginning because the user already knows what they want to do at the gym. It is not necessary to suggest exercises to them right at the beginning. There is no need for a transition or cut point between this generic recommendation system and the one we are outlining. Instead, the second one will start adding onto the first one from a certain point on.

The idea is that the recommendations provided by the engine will **help users evolve their workouts based on which exercises they already do**. **Two main users** can be pictured: The first one has a rather **limited repertoire** of exercises, potentially lacking some of the staples. We would want to be recommending the staples to this kind of user. The second one has an **extensive repertoire**. This user should be recommended niche exercises.

Broadening the range of exercises performed by users will have major positive effects on the results they will be able to obtain, improving customer satisfaction and potentially attracting new customers. Soreness after trying a new exercise is commonly known. Then why do users not already switch up their exercises a lot? On one hand there is sheer laziness. Exploring new exercises require additional effort, first to find them and then to learn how to perform them. In addition to that, it often is the case that one does not like the new exercise one tries, leading to frustration about wasted time. This new recommendation engine will ideally be recommending new exercises that the user will actually like and keep using, while at the same time having an interface that explains them well, lowering the effort needed to try them and nudging the user towards this good behaviour of exploring new exercises. A short video should be the best way to introduce the user to a new exercise.

The next section will discuss the optimal design of the system in order to best profit from other users' workouts.

Technical Requirements and Solutions

The first decision to be made regards **diversity** of recommendations. We want to **exploit the known best exercises as much as possible**, suggest exercises which we know will fit the person well. However, there is also an incentive to explore less known, **innovative exercises**. Luckily, this issue should resolve itself for the most part. The key characteristic is that a **user should not be recommended an exercise they have tried recently**. (That already happens in the predefined workouts.) If there is an exercise which the user does not already perform and which is recognised to be a very good fit, it should without doubt be recommended to the user. Only after the user has tried these exercises should the system begin to recommend more exotic exercises. Thus, **exploitation shall be prioritised.**

Stability refers to a quality of recommender engines to repeat the same recommendation to a user in more or less the same circumstances. The workout case is particular: **If a user is planning to train legs on a given day, they would not want to follow a recommendation for biceps curls**. On another day, however, that could be a good recommendation. Hence, a certain degree of stability is desirable. This will be solved through the choice of the main channel of delivery of the recommendations. Each time the user visits the gym, one exercise will be recommended to them. At the same time, **the past 4 recommendations will remain visible**. Ten exercises at the time will be generated through one recommendation, as a **medium length list** is predictable far better than very few items.

Explainability is another frequent requirement to recommender engines. In our case, it might be desirable to be able to mention that "Exercise A was recommended to you since you performed Exercise B". While it in most cases is self-evident, which body parts are trained through an exercise, a user might be **more likely to try an exercise which apparently is similar to one they already perform**; not only in terms of muscles trained, but also in terms of intensity for example.

This leads to another important design point; prior to the introduction of the system, all customers should be **thoroughly briefed** about it, as to avoid them being freaked out when the app knows which exercises they perform.

Evaluation

The system's evaluation is not entirely straightforward. A possible **offline evaluation** would be prediction accuracy or RMSE on unseen data. This will yield a **first impression** of how well the system can predict exercises that a user is missing and which are performed by most other similar users. The **final validation**, however, should happen through **A/B testing**. A fraction of all users will be shown the new recommendations in the Xtrain app. The **KPI** proposed to measure the goodness of the recommendation engine is the **percentage adoption of recommended exercises,** to be compared with the percentage use of the predefined workouts. As a **long-term KPI**, increase in **strength** of recommendation-adopters as opposed to non-adopters shall be measured. As mentioned before, improved results for Xtrain members will the number of users leaving and attract new customers. Albeit being the ultimate goal, increase in members will not be used as a KPI as it is too dependent on other factors such as trends, season and temperature, which would lead to an imprecise measurement.

The long-term KPI (strength) will be a superior measure of success compared to offline accuracy, especially for measuring the recommendations to users with an already vast repertoire of exercises. Recommendations made to them will be based on the few users which are one step further in workout diversity.

Algorithm

Different algorithms will provide the best recommendations for different users. We will first outline these algorithms and then discuss how to combine them.

To begin with, **popularity**-based recommendations can be expected to perform well for users with a small repertoire who are lacking **staples**; staples correspond to the most popular exercises in general. Defining popular exercises as the one performed most often on average should be sufficient. Alternatively, association rules using support and confidence should yield a similar result and will also be explored.

Lift also is a valid measure to detect good recommendations in our case. We wish to find exercises that other users are significantly **more likely to do,** based on the fact based on the fact that they have several **exercises in common** with the user we are recommending for. For example, take a tennis player who routinely performs a rare lateral abdomen exercise which is beneficial for playing tennis. At the same time, most other users performing this exercise, also tennis players, additionally perform a chest exercise which ideally targets the muscles needed for tennis playing. As a result, the **conditional probability** of a user performing that second exercise is very high, given they perform the first one. Lift is high, and indeed we would want to recommend the user this additional exercise.

Content based filtering would be **weak** in this scenario. The “content” known for each exercise is little, and recommending exercises based on it would basically result in recommending further shoulder exercises to someone already working out shoulders, for example. In the future, with a more exhaustive dataset on the characteristics of the exercises content based recommendations could be considered. Nevertheless, such rules would be based off experts’ opinions on which exercises have common characteristics, rather than on the stronger, implicit information contained in the user patterns observed.

Collaborative filtering really is the way how a recommendation engine can add extra value in this case. It is the approach through which we can **unveil the hidden relationships** between items and between users, making according predictions. **User based collaborative filtering** (UBCF) intuitively can be expected to **best** discover the top items to recommend. In our initial statement, we outlined how we would aim to recommend exercises to a user based on analysing users **performing similar exercises** - or, more in depth, users with similar fitness goals. Conversely, **item based collaborative filtering** (IBCF) would essentially connect exercises which are similar in terms of **which users perform them**. There could also lie considerable value in these recommendations, as they would for example recommend really complex exercises to a user who is into complex exercises. IBCF would come with the advantage of far less frequent need for retraining. With a set of items remaining more or less the same, the model will only need to predict on new users without need for retraining.

For obvious reasons, **Re-recommender** systems do not make any sense in this scenario. **Random** recommendations run against the decision made that exploitation should be prioritised over exploration.

Context Awareness

It is reasonable to assume that several contextual variables will have an influence on which exercises should be recommended to a user.

One evident feature is **time-awareness**. An option which we will explore is to have the algorithm **forget about exercises a user has tried a long time ago**. The rationale behind this can be that a user got stronger and now would like to try an exercise they previously did not.

The **time of the day** could be considered for predictions as well. Perhaps, people are more likely to run in the morning and lift weights at night. A simple implementation of this would be to train two separate recommendation engines for morning and afternoon, including only exercises performed at that time by users. In a similar approach, with preferences varying from country to country, location-awareness, in the form of different engines for different countries, could be explored.

Ensemble

While each algorithm has different advantages, most of them come with drawbacks as well. **Popularity** based algorithms, for instance, tend to lead to **feedback loops**, recursively enforcing the most popular items, leading to polarisation on the most popular items. Combining different algorithms can circumvent these drawbacks while extracting the most relevant recommendations from each algorithm.

In practice, we **implemented all of the algorithms selected earlier** by themselves and now will build upon them to construct ensemble recommenders. When applying the algorithms to the real data, before proceeding we would want to evaluate the individual algorithms' performance offline through accuracy, and then use the best ones in the ensembles. Owing to the fact that we are working with simulated data, users and items actually are uncorrelated within themselves, meaning that no meaningful rules can be found. Therefore, comparing accuracy would be illusionary. Either way, the construction of the ensemble methods allows for easy substitution of algorithms.

We begin with a simple **weighted ensemble using UBCF and IBCF**, weighing the predictions of the two engines, with slightly higher weight (0.6) on UBCF.

A more advanced ensemble with a stronger business foundation is created as an **ensemble over experience**. If the user's **strength is below a set threshold**, **popularity**-based recommendations are returned, aiming to provide the user with the most common exercises. For **stronger** users, on the other hand, a **UBCF** recommendation is given, as they can be expected to already have a more optimised workout routine which includes staples.

Recommendation Delivery

The main way of delivery proposed for the recommendations is to include them in the app, alongside the predefined workouts. They would be **pop-up like**. Alternatively, an implementation where users can **actively request recommendations** could be considered. One advantage of this is that it would allow the user to **personalise the recommendation** at the time it is queried. For example, they could specify that they wish to see recommendations for calves exercises.

Limitations and Improvements

One problem that will be encountered is that our definition of strength, being the average of weights used in the most common exercises, will not work for everyone as not everyone has performed them. However, imputing the missing values with zero should produce a reasonable proxy, assuming that those users who have never performed these exercises will be rather weak on average. Nonetheless, strength can have some limitations when used as a KPI. The way it is currently defined, it will not capture increases in strength of abdominals, for instance. A more in-depth measure designed by Xtrain's fitness experts should eventually replace it.

Another issue is that certain users perform rare, unidentified exercises, which will not be captured by the computer vision. While the recommendations for these users will be off, it can be assumed that the reason why they perform these exercises is that they already opt for a very diverse set of exercises, such that there is no need for recommendations.

Conclusion and Example Results

The table below shows the output of exemplary recommendations for the ensemble over experience.

	User_3995	User_3996	User_3997	User_3998	User_3999	User_4000
Strength	78	196	26 (weak)	85	191	190
Rec 1	side lying groin stretch	atlas stone trainer	hanging bar good morning	anterior tibialis-smr	anterior tibialis-smr	posterior tibialis stretch
Rec 2	overhead triceps	all fours quad stretch	side to side box shuffle	atlas stone trainer	atlas stone trainer	standing overhead barbell triceps extension
Rec 3	on-your-back quad stretch	atlas stones	cable hammer curls - rope attachment	all fours quad stretch	all fours quad stretch	atlas stones
Rec 4	hack squat	otis-up	barbell deadlift	atlas stones	atlas stones	box jump (multiple response)
Rec 5	good morning	landmine linear jammer	plate twist	otis-up	otis-up	seated hamstring

The next table provides a brief overview of the evolution of rules through the different recommendation engines, for the same user.

	Popularity	AR (lift)	UCBF	ICBF	weighted ensemble	ensemble over exp.
Rec 1	standing olympic plate hand squeeze	spinal stretch	linear depth jump	band hip adductions	linear depth jump	low cable crossover
Rec 2	box squat with bands		dumbbell squat	looking at ceiling	dumbbell squat	alternating kettlebell row
Rec 3	reverse barbell curl		dumbbell bench press	peroneals stretch	standing military press	circus bell
Rec 4	return push from stance		standing military press	bent-knee hip raise	dumbbell bench press	incline cable chest press
Rec 5	tricep dumbbell kickback		balance board	dancer's stretch	balance board	upper back-leg grab

The recommendations, based on algorithms selected through a thorough process of evaluation which looked at the reasoning which would underlie a recommendation from each algorithm, will provide Xtrain's users with the power to step up their workouts. By learning the profiles of each user through implicit data, the recommender will be able to provide each user with the exercises they truly need to achieve their fitness goals and bring Xtrain's app to the next level.