

# Orac Recommendation System

Thomas E. Portegys  
[portegys@gmail.com](mailto:portegys@gmail.com)

---

## Overview

The purpose of the recommendation system is to recommend resources to users based on feature classification and resource preferences. A user's features are, for example, demographic and medical conditions. A user's preference is, for example, a rating for a video. The system will be able to recommend resources that are input by an expert, such as a physician, and resources based on similarities with other users in the system.

## Functions

1. Input, update, delete and query user. A user id can be optionally created for a new user, and features with corresponding importance weights are stored for the user. The query can use optional fields to perform a search operation.
2. Input, update, delete, and query resource. A resource id can optionally be generated. A resource can optionally be categorized, e. g., video, activity, or advice.
3. Input, update, delete, and query user resource recommendation. The status of a recommendation, e. g. completed, can also be input. A recommended resource can also be applied to a group of users based on description matching.
4. Input, update, delete, and query user resource preferences and status. The status of a recommendation, e. g. completed, can also be input. A recommended resource can also be applied to a group of users based on description matching.
5. Monitoring and reporting.

## Operation

In addition to producing a list of pre-recommended resources, perhaps input by a physician, generating a set of recommended categorized resources for a user consists of:

1. Finding a set of similar users based on features and feature weightings.
2. For this set of "neighbors", further selecting those that have similar resource preferences.
3. Produce a list of resources that have yet to be completed.

## Implementation

The Weka machine learning java library contains an extensive list of libraries with a programmable API. It is also capable of interfacing with conventional relational databases, such as SQL Server and MySQL. An API for the recommendation system will be produced and implemented using these components. The recommendation system can then be used as a service for a web-based REST interface on one or more servers. In due time, a cloud-based (scaling) implementation can be done. The API to the database should be constructed with this in mind.