## COMP9321: Assignment 2 Meeting One

### Aims of Service

**Machine learning service**:

The service will recommend similar board games given a board game chosen by the user. This prediction is based on historical preferences from other users. It will use a collaborative filtering model, the implementation will rely on the FastAI library built on pytorch. <https://docs.fast.ai/collab.html>  
*(Note: similar reviewers are expected to give similar ratings for similar types of games. This is expected to be modelled mathematically by the vector space model)*.

**REST server functionality**:

* ***GET /board\_game***

Get a list of all board games (with IDs and URLs).

* ***GET /board\_game/id***

Get board game information directly from the .csv data. This includes information about the board game and any of its existing reviews (with IDs).

* ***GET /board\_game/id/rec?n=30***

Get the *cached result* of the machine learning service on a chosen board game. The number of entries can return up to a limit (e.g. 30).

* ***GET /usage***

Get all recorded API usage metadata, for both aggregate results and for each URL.

* **POST request** (optional)

Submit a new review for a specific game. However, this will not re-train the machine learning service.

* **PUT request** (optional)

Update an existing review (given its ID) for a specific game.

All REST data output will be formatted in JSON. All functionality will be documented in Swagger.

**REST server usage metadata**:

As mentioned above, this will be both aggregate results and results for each URL.

**Authentication**: will not be considered for minimum-viable product.

**Client interface**:

* HTML/CSS (see mock-up).
* Designed to consume API endpoint.

### Datasets

From Kaggle. Contains three .csv files: <https://www.kaggle.com/jvanelteren/boardgamegeek-reviews>

**2019-05-02.csv** - BoardGameGeeks review meta data

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Name | Year | Rank | Average | Bayes average | Users rated | URL | Thumbnail |
| 1 | Gloomhaven | 2017 | 1 | 8.9 | 8.611 | 24574 | /boardgame/174430/gloomhaven |  |

**Games\_detailed\_info.csv** - information about the board games for RESTful service

* Game ID (id)
* Name (Primary)
* Publisher (boardgamepublisher) / Publish Date (year published)
* Image (image)
* Description (description)
* Expansions (boardgameexpansion)
* Category (boardgamecategory)
* Board Game Mechanics (boardgamemechanic)
* Min Player / Min Age / Minimum Play Time / Play Time (minage/minplayer/minplaytime/ playingtime)

**Bgg-13m-reviews.csv**

* Used for training machine learning model on user-item interactions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | user | rating | comment | ID | name |
| 0 | user\_name | 10 |  | 13 | Catan |

### Communication Channel

Facebook messenger for initial contact and Slack for development.

### Code Repository

Github: <https://github.com/JerryyZhu/analytics-teamwork>

### Team Roles

*The Mix* - <https://webcms3.cse.unsw.edu.au/COMP9321/19T3/groups/20556>

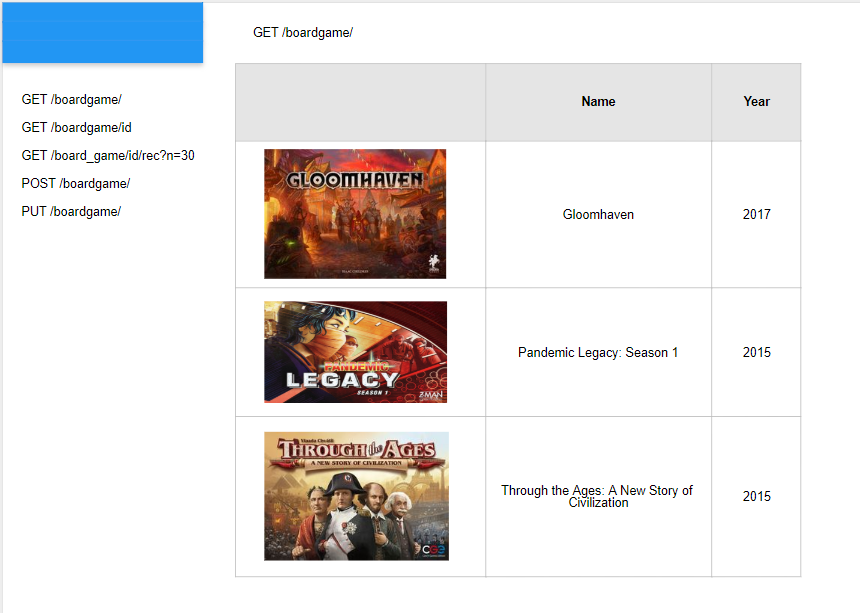
|  |  |
| --- | --- |
| **Member** | **Role** |
| Bowei Zhu | Developing and refining the machine learning model |
| Jiaying Yi | Developing and refining the machine learning model |
| Niriksha Mutharia | Client-side request calls  Client-side user interface |
| Samer Alhaddad | Server-side REST functionality to interface with the data / metadata  Swagger and deployment documentation |
| Victor Tse | Data cleansing  Server-side data query interface (function wrappers)  API metadata storage (function wrappers) |

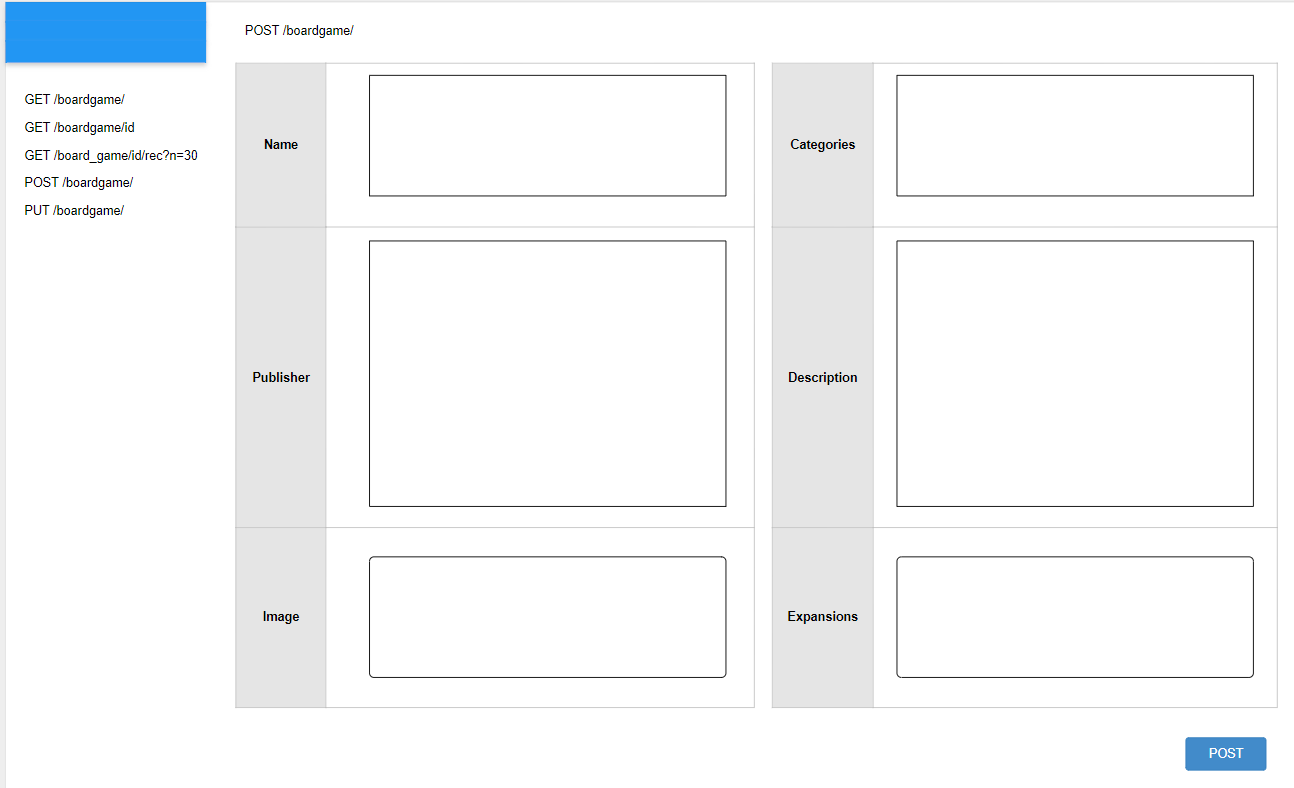
### Use Case Summary

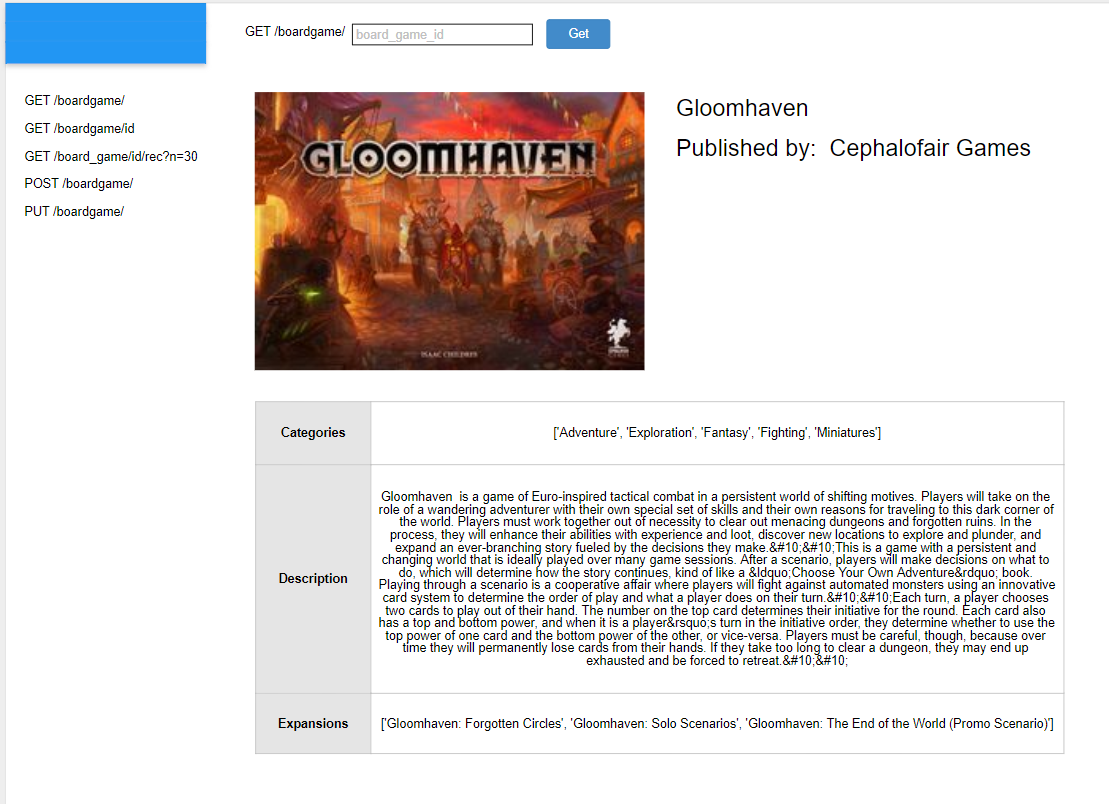
* Provide board game information such as year published, publisher, etc.
* Users can get suggestions/recommendations for board games they are interested in.
* Analytics are available for assessing endpoint usage.
* Top 10 rated games (Google Charts, Chartist.js).
* (Extension) From a business standpoint, an affiliate link to purchase the board game could be provided, thereby generating income from the API.

### Mock-up

<https://app.moqups.com/ztKjDE2ZSS/view>



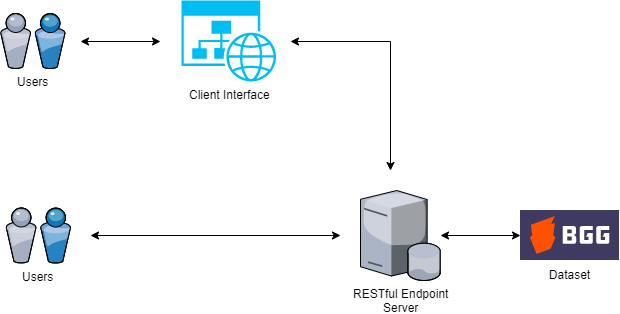




### Technical Architecture

<https://drive.google.com/file/d/1sQ-qHTB7_zNPVp6HcZMxkU6olAAC-N0J/view?usp=sharing>

Both client (user interface) and server (data endpoint) will be implemented with Flask.



### Performance Assessment

* Minimise loss function (RMSE)
* Empirically look at the results and see how similar suggestions are

**Meeting 1 Discussion Points**:

Has the group established an effective communication channel (only in English) for group

work? (e.g., messaging, code repository)

* Facebook messenger for initial contact
* Slack: <https://app.slack.com/client/TPQN5RKGS/CQ3MS465A>
* Github: <https://github.com/JerryyZhu/analytics-teamwork>

Has the group established the roles and tasks necessary for the group work?

* See documentation above

Has the group identified datasets to use the Data Analytics Service?

* Link to the datasets: <https://www.kaggle.com/jvanelteren/boardgamegeek-reviews>
* Dataset: 2019-05-02.csv, games\_detailed\_info.csv, bgg-13m-reviews.csv

Has the group identified the use‐case scenario for the Data Analytics Service application? (i.e., what is the service going to offer for the users?) Is it documented well?

Has the group identified the (high‐level) system architecture and components necessary for the

group work? (not to be detailed, high‐level plan)

**INITIAL MACHINE LEARNING NOTES**: from Jerry (Week 9)

* Columns of interest: user, game, rating. 13 million entries.
* Make a sparse matrix of users vs. games. The actual element will be the rating. The matrix is sparse as we don’t have predictions for most things.
* Each game and user has a weight associated with it. In our model, we chose n = 50.
* Each game and user has its own vector representation of 50 points, and we do the dot product of the two vectors to predict the rating.
* The machine learning model changes the vector weights until you get a small enough error (via RMSE). The loss function is to minimise RMSE. All weights are initially generated randomly.
* After you do the training to randomise/adjust the weights to minimise the error between the prediction and actual value.
* The games are the categories for the data. Each user and their review is one entry for this game category.

**Code walkthrough**:

* CollabDataBunch: some abstraction of the data the library uses…
* EmbeddingDotBias model: basically a giant matrix. Think of each user review as a row and each
* Learning rate: how quickly the model can change/update its weights. Need a trade off between how quickly model achieves optimal performance
* Train the model on the data (takes 1-2h). We do training in batches - basically the number of training iterations.
* We train on as much information as possible. However, from a business standpoint, for more reliable results, we make use of only games with a certain amount of popularity (determined by how many reviews they have: in this case, if we have >= 100 reviews).
* KNeighbours finds the n\_neighbours = 5 of the closest neighbours of all the top games in that list. KNeighbours is needed to find similarity between the games.
* “List of top games” is the list of games with >= 100 reviews.
* Pass that into the bias model. If we know the vector point for our game of choice, we run KNeighbours
* Users serve as game characteristics about a game
* We train the model based off of that

Given a board game you have already selected, choose recommendations for other board games (based off of user review preferences)

**INITIAL BRAINSTORM**: last updated 13/11/19 (Week 9)

This section contains a listing of all the various ideas that were generated. The final idea that is chosen for our project should be written up in a more formalised manner.

* What is the aim of the service?
  + Predict rating of the recipe -> allows a user to determine whether the recipe can be introduced to their restaurant's menu or not.
  + Predict the healthiness of the cuisine based on the ingredients used in the recipe.
  + Predict the cost for making the cuisine based on the approximate price of the ingredients used in the recipe.
  + Sentiment Analysis on amazon reviews.
  + Board game recommender system based on user’s preferences, who else likes it?, rating of the game. - **Jerry is looking into it**
* What are the datasets?
* What is your communication channel?
  + Slack and Facebook messenger
* What is your code repository?
  + Github link? - **Jerry is looking into it**
* A very brief description of each member’s role in the project
  + Perform Data Integration and pre-processing
  + Building a machine learning model
  + Designing a RESTful API
  + Designing a Simple Client with GUI
  + Provide documentation for your service (swagger doc and deployment instructions)
* Project documentation; correctly described use cases using notations, such as sequence, diagrams, mock‐ups, or plain natural language; the method is optional

**MEETING MINUTES**: last updated 13/11/19 (Week 9)

Board Game

Dataset - <https://www.kaggle.com/jvanelteren/boardgamegeek-reviews?fbclid=IwAR2kU8Czaw0ryTcO8rvksR0i0gW97gjr11-tCCHW6wlLbGzUuP_S_QjdnHo#games_detailed_info.csv>

Github - <https://github.com/JerryyZhu/analytics-teamwork>

Food.com dataset

<https://www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions>

Food idea -> Kaggle -> food.com

* predict ratings of the recipes beforehand
* Predict food trends?
  + Would need social media to keep it up to date
* predict where the cuisine comes from based on the ingredients

<https://github.com/stevenjson/CuisineClassifying/tree/master/Data>

* selection of different recipes a user has -> predict what other recipes they might like <https://www.kaggle.com/hugodarwood/epirecipes>
* if financial data available -> predict how expensive is it to make the recipe

<https://www.kaggle.com/jboysen/global-food-prices>

* nutritional info -> classify as healthy/unhealthy (range)

<https://www.kaggle.com/trolukovich/nutritional-values-for-common-foods-and-products>

* <https://www.kaggle.com/snap/amazon-fine-food-reviews#Reviews.csv>

Nutritional info - classification

Classification of healthiness of a recipe?

* <https://world.openfoodfacts.org/product/00004559>
  + View individual items here: <https://world.openfoodfacts.org/>
  + Website shows A-E health ranking - not sure if in dataset, but most entries should have a ranking
* <https://hackernoon.com/machine-learning-food-datasets-collection-fz5i130w0>