**Profile Based Recommendation System for Airbnb**

**Problem Statement:**

Airbnb is an online market place that enables users to list, find and rent vacation homes. Airbnb connects people to unique travel experiences, at any price point, in more than 34,000 cities and 191 countries. Airbnb has a diverse customer base, it is used by people from different age groups, professions and interests. Some customers prefer places which are closer to the city whereas some others love to enjoy the natural beauty by living in cottages. Since every customer has different interests it is essential that the user is recommended houses on the basis of their profile.

**Proposed System:**

We have developed a profile based recommendation system using supervised learning techniques where user’s future trips are predicted on the basis of their past trips and preferences. User's age group, profession, interests and most frequently traveled seasons are taken into consideration for building a more accurate model. The proposed system would build a good customer experience by reducing the average time of booking and better demand forecasting

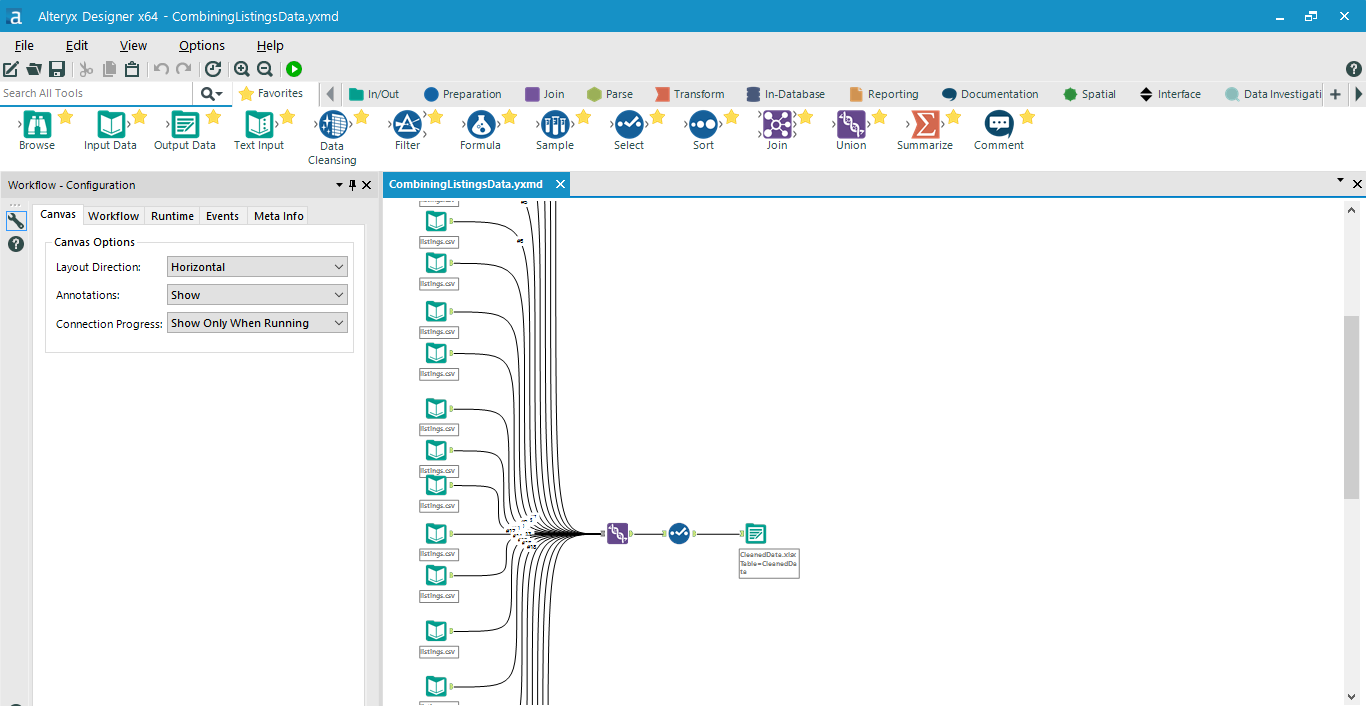
**Approach:**

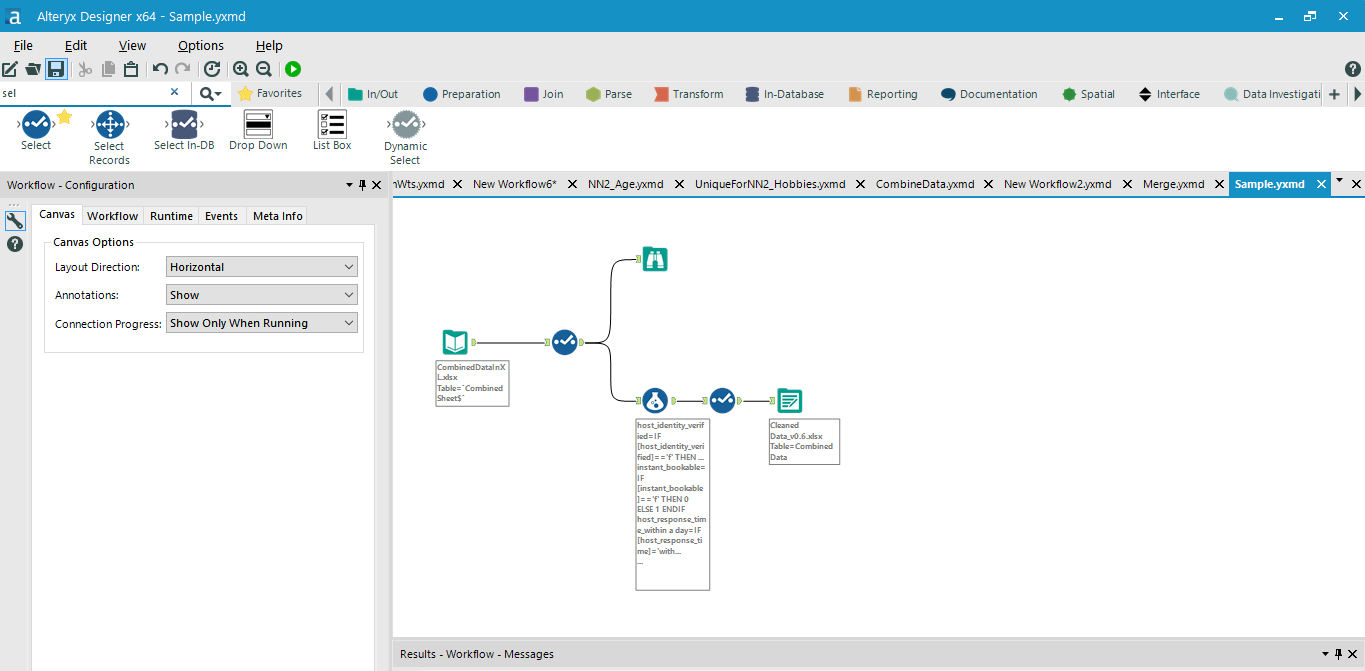
**Step 1: Data Cleaning and Data Transformation**

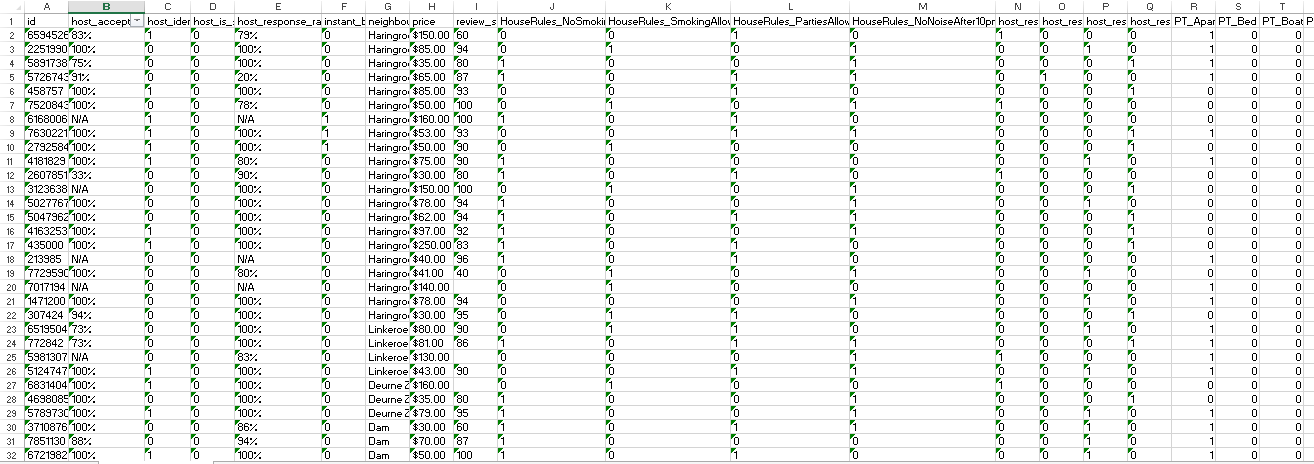
Airbnb provides an open source dataset which contains the 1.3 million listings that are posted on their website and Airbnb users and their profile details. The listings data contains information like number of bed rooms, amenities, ratings, house rules, GPS location, etc. The users data contains information like age, profession, past trip details, etc.

**Data Cleaning:** We performed data cleaning on the dataset using Alteryx tool. Data cleaning included handling missing values and garbage values, outlier detection and data normalization.

**Data Transformation**: Using the user’s travel history, we computed the seasons in which the user most frequently travels in. We identified the type of apartments the user stays in to identify user’s interests and preferences.







**Step 2: Calculating weights for every listings using Neural Network**

For every listing we computed weights to assign how suitable a listing is on the basis of age-group, profession, seasons and interests

Step 2.1. For every listing we gathered the user’s that have already stayed in them. We identified the age group they belonged to, their profession, the seasons in which they like to travel and their hobbies.

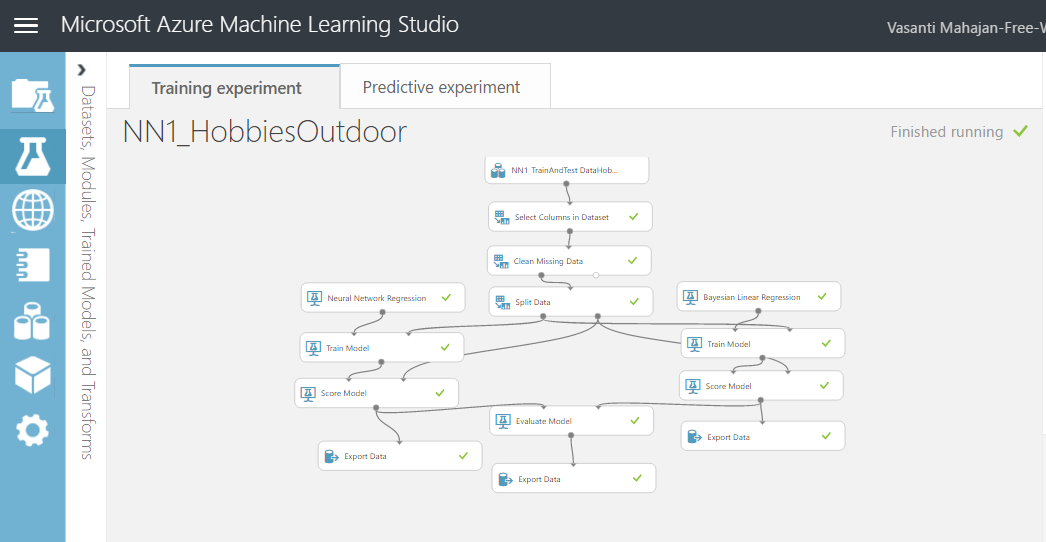
On the basis of min max normalization technique we computed weights for each of these parameters and fed it as the training data for our neural network.

Min max normalization= 

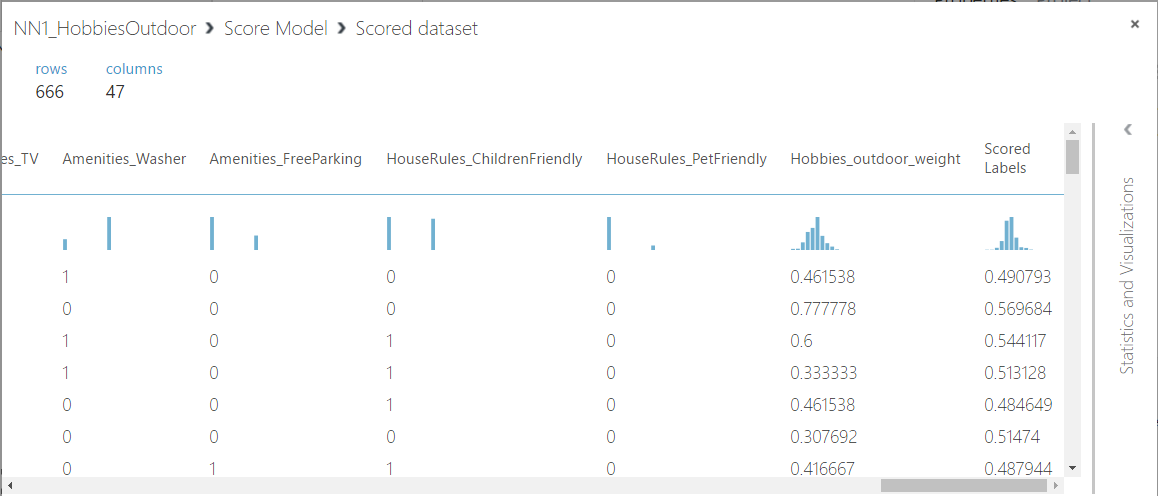
This technique ensures that the data is scaled within the range of 0-1.

We used Multi Layer Perceptron with Back Propagation which had 42 inputs and 13 outputs, We used 12 hidden layers for training the Neural Netwrok.

***Our Neural Network was trained with the root mean squared error of 0.022119.***

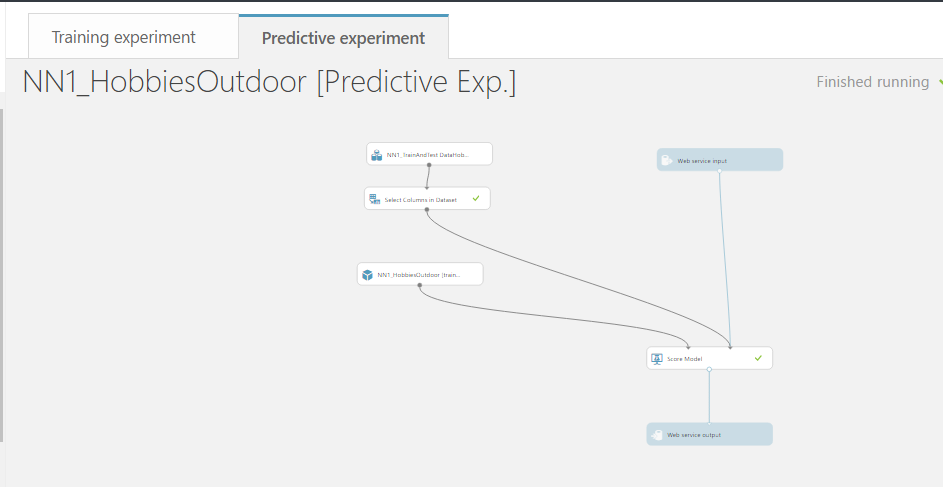


*Training model for computing Hobbies Weight*

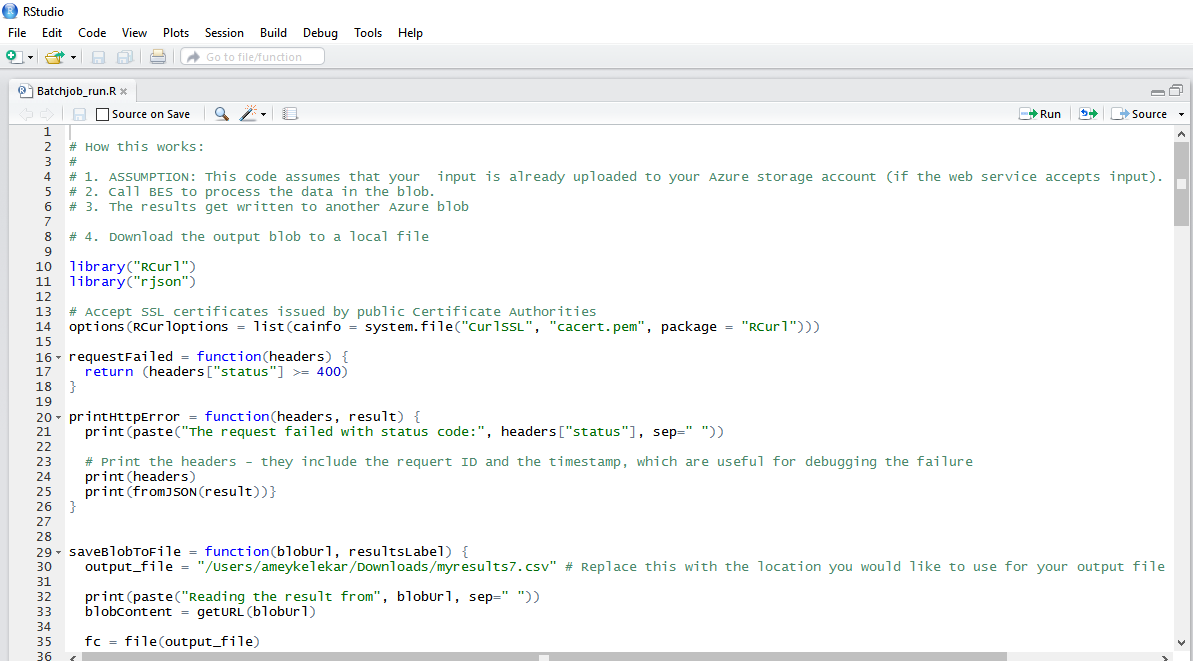


*Predicted results on the Trained Model*

Step 2.2. Once our neural network was trained a web service was created with the trained Neural Network Algorithm. Using an R script we consumed the web service, which computed the results on the basis of trained model, thus computing the weights for every listing.



*Creating a web service for running our test data on the Trained Model*

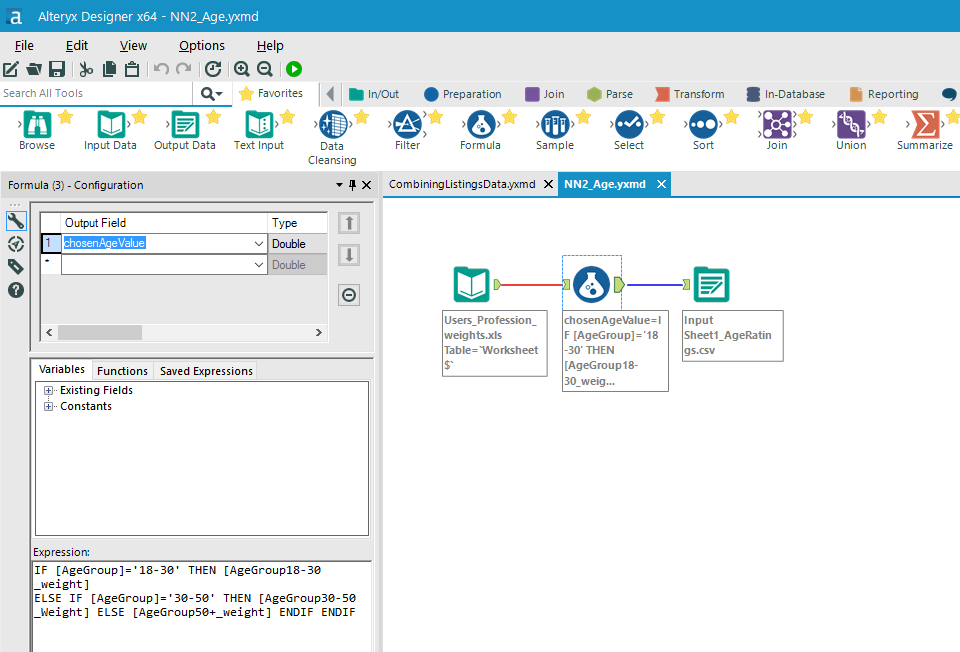


*R Script for consuming the web service and computing the weights on the entire dataset*

Thus by computing the weights for every listings we identified the most suitable listings for every age group, profession, seasons and hobbies.

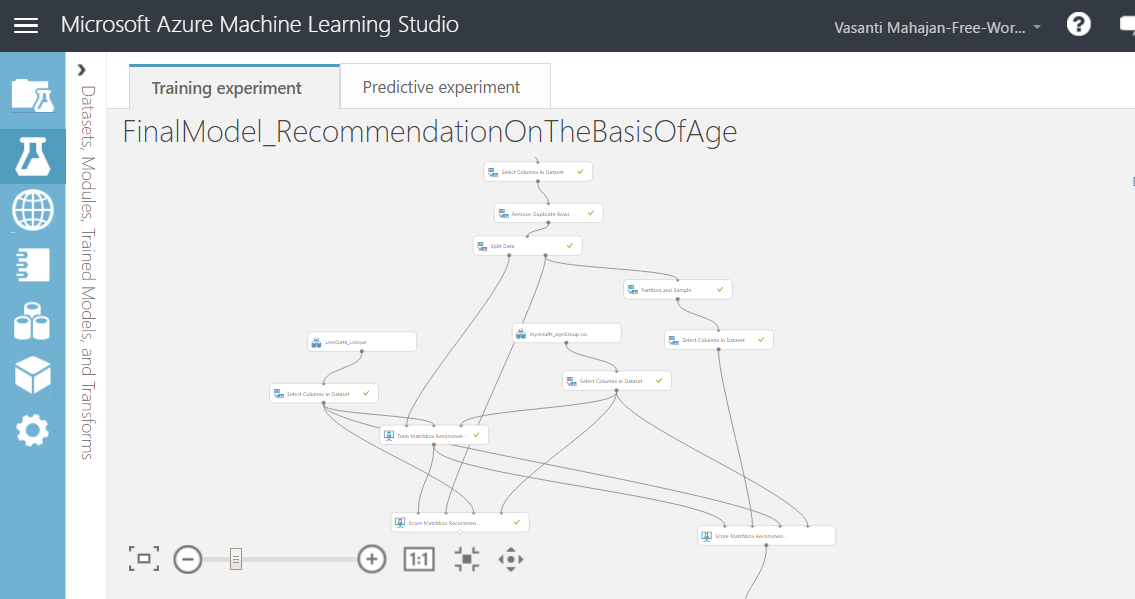
**Step 3: Predicting user’s future trips**

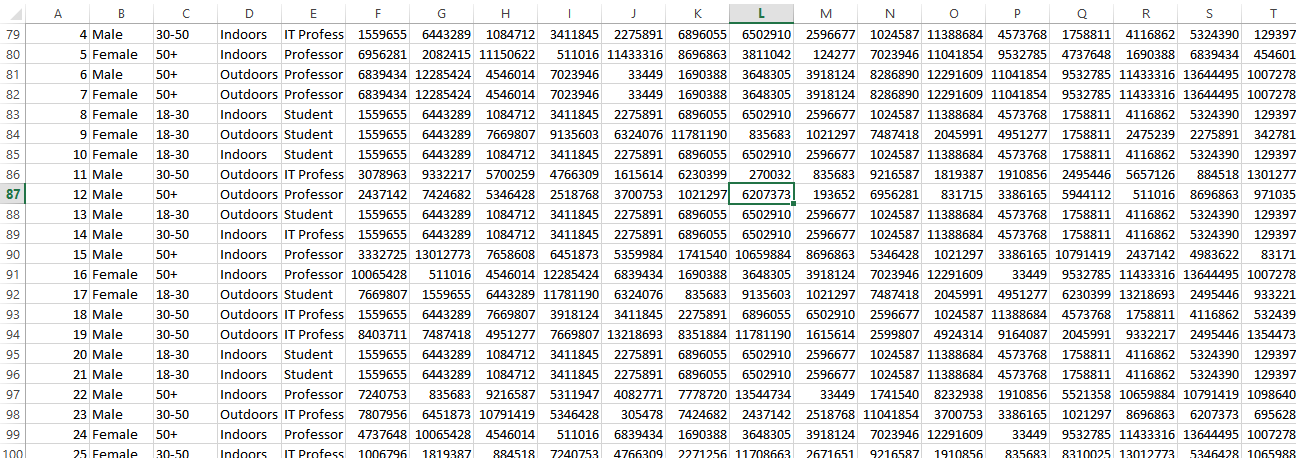
Step 3.1 Using Matchbox recommendation algorithm provided by ML studio, we computed triplets of UserId, ListingId and AgeWeights. The age weights are the weights that were computed by our neural network for every listings. We chose AgeWeights of the age group in which the user belongs to.



*Choosing the AgeWeight for every listing in which the user belongs to*

We trained our data on the training data set. We also fed the user’s data and listings data for training the model. We computed top 25 recommendations for the user based on their age weights.

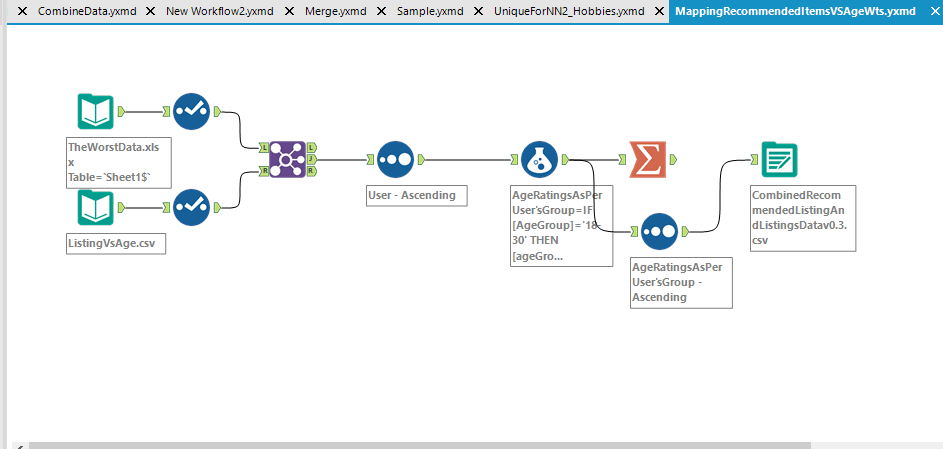




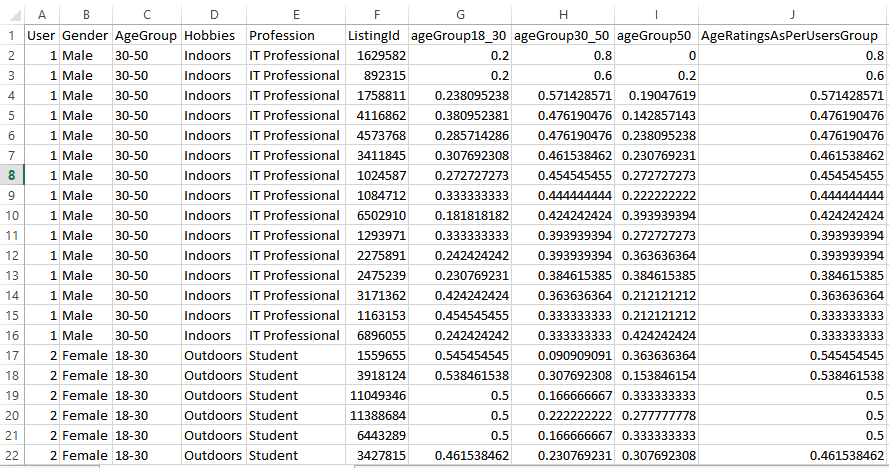
*Top 25 listing recommendations on the basis of AgeWeight*

Step 3.2

Once we got the top 25 recommendations, we sorted these recommendations by their age weights and top 15 listings were chosen for the user. This was performed in R

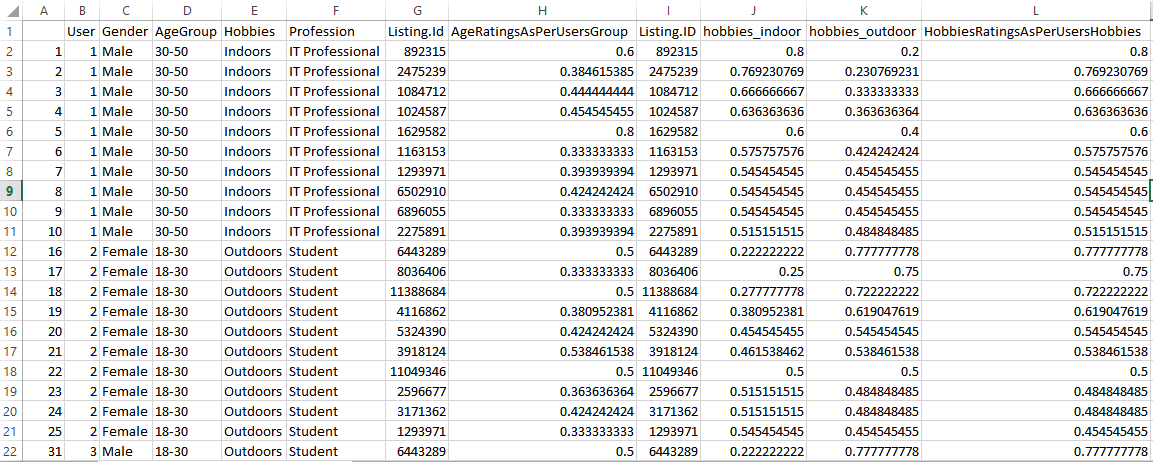


*Sorting data on the basis of Age Weights*



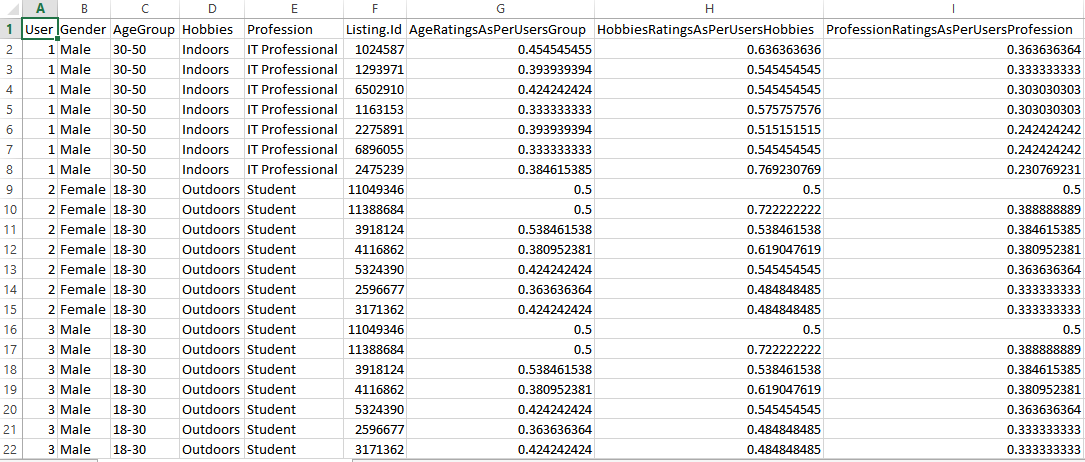
*Top 15 listing recommendations sorted on the basis of AgeWeight*

Among those top 15 recommendations, we mapped the hobbies weight for every user and identifies the top 10 listings for the user

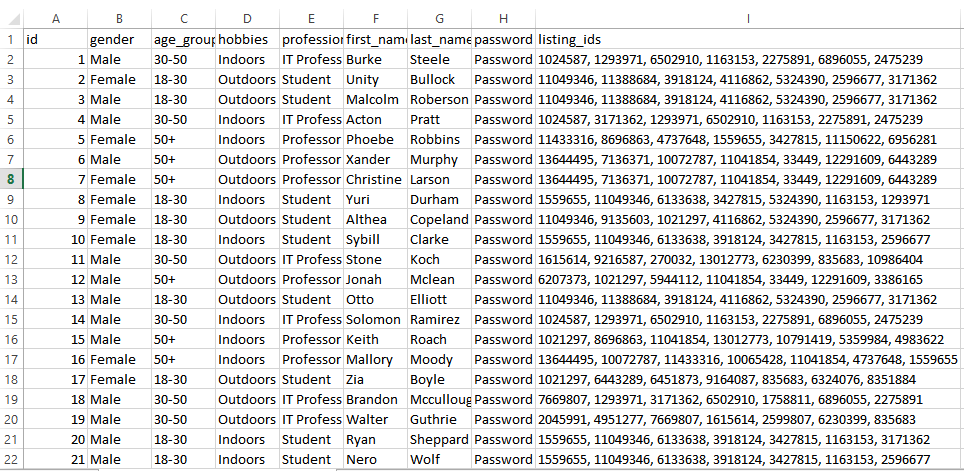


*Top 10 listing recommendations on the basis of HobbiesWeight*

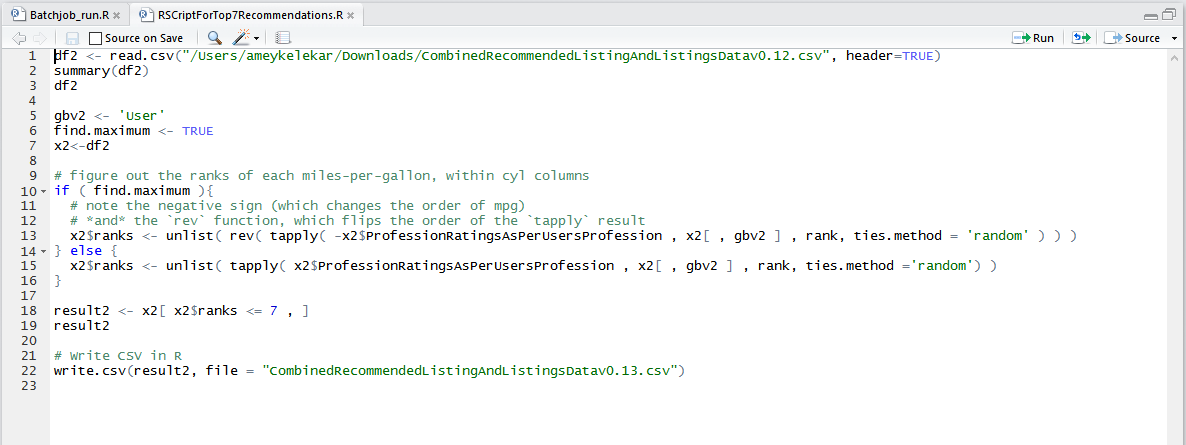
Among these top 10 recommendations, we mapped the profession weight and identified the top 7 listings for the user.



*Top 7 listing recommendations on the basis of ProfessionWeight*



*Top 7 listing recommendations on the basis of AgeWeight, HobbiesWeight and ProfessionWeight*



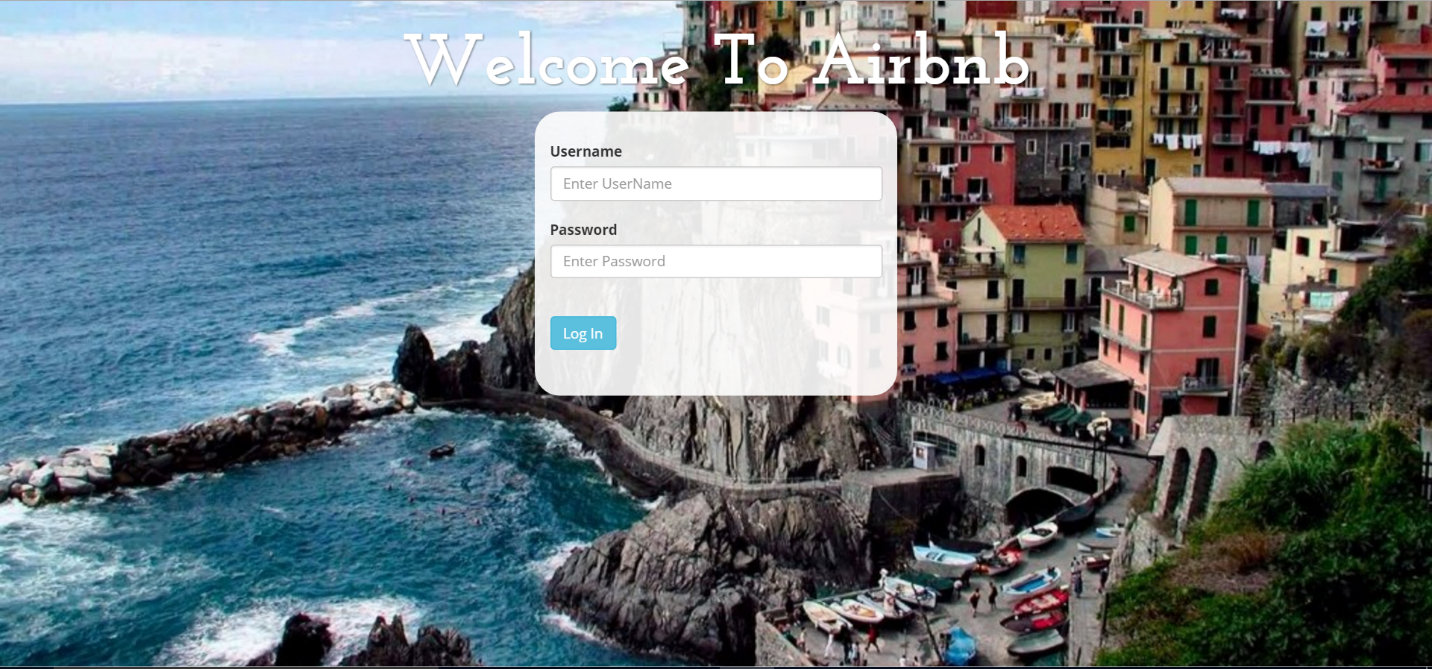
These 7 listings were perfectly in sync with user’s profile and were recommended to the user. We observed 89% accuracy in the user’s profile and listings that were recommended.

**Implementation:**

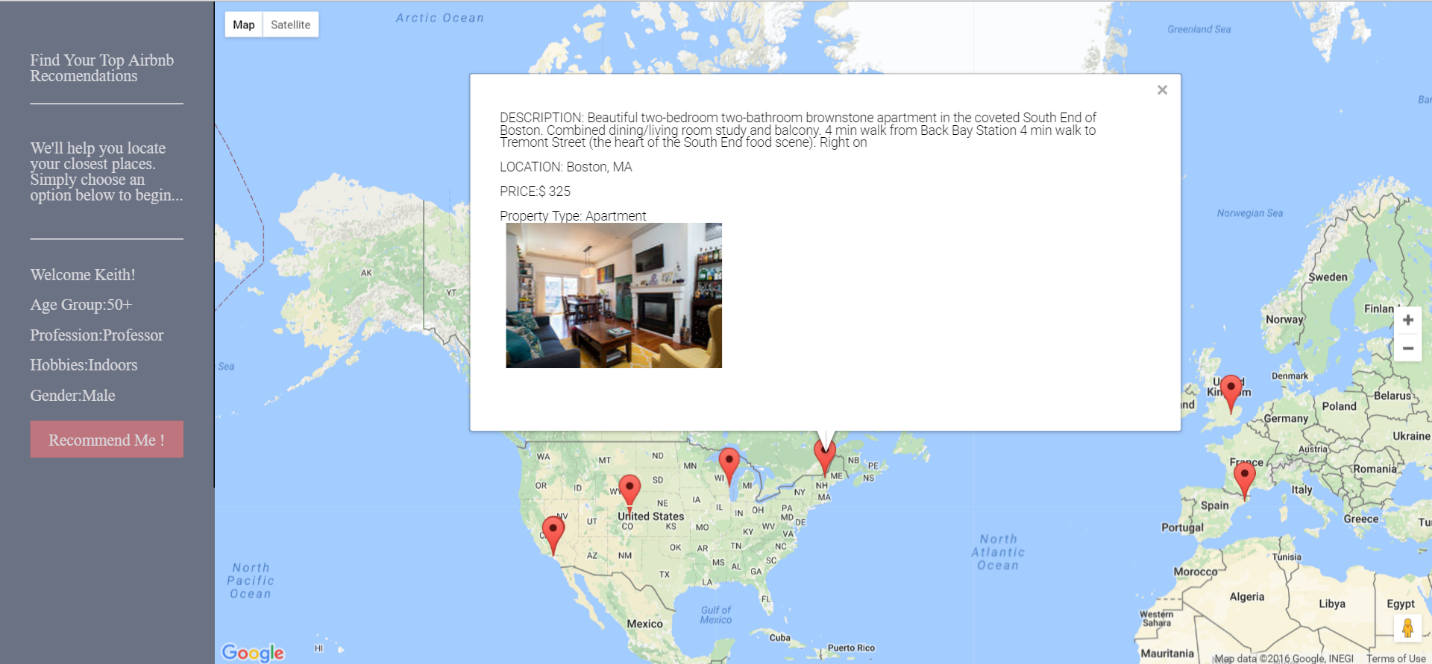
We build a live spring application using Mongo DB and Morphia. 1.3 million listings and the users data were stored in MongoDB. The listings that were recommended to the user were fed into Mongo DB and were displayed in the application on Google Maps.

We implemented Google Maps API to display the details of the recommended listings for every user. An asynchronous call was made using Ajax to fetch all the listing objects as JSON. By iterating on all the listing objects we extracted each and every listing’s latitude and longitude and fed them to the marker objects of the google maps. These markers were send to the API so that they can be displayed on the map. Each marker was binded with a click event which was done using closures. This entire function was perfirmed by JavaScript alone (Functional Programming).

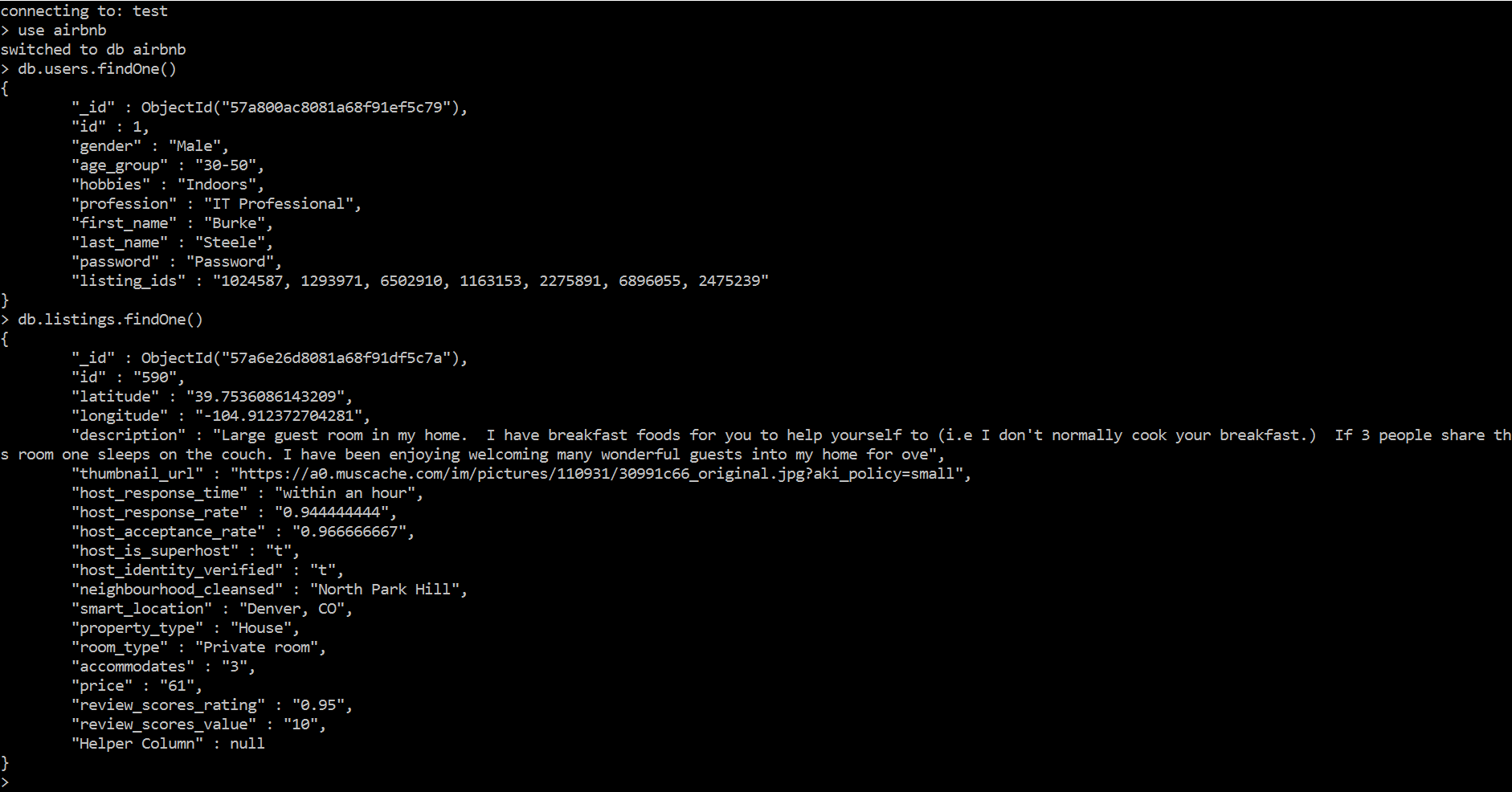
**Project Demo:**

****

*Login screen*

****

*Personalized profile based recommendations for user Keith*

****

*Data stored in MongoDB*

**Challenges Faced:**

The project was really challenging to work with since we used a different approach for recommendations. Below are few of the challenges and how we overcome each of them.

1. **Choosing the parameters for profile based recommendation:** Choosing the parameters of the user’s data and performing data transformation was one of the initial transformation. We referred to the profile based recommendation system build by Spotify to analyze the techniques they use to segregate their customers based on their profile
2. **Normalizing data:** Since the Neural network requires the data to be normalized. We performed several types of normalization techniques like Min Max normalization, Z score normalization, to name a few
3. **Computing the results for multiple parameters in one Neural Network with minimum mean square error:** Since Neural Network provides prediction on only parameter at a time we had to segregate our data and compute the weights for every parameters individually. We used several combinations of hidden layers, learning rate and momentum to get accurate results
4. **Using Morphia:** Learning Morphia in order to connect with Mongo Server was challenging
5. **Modelling the data model for the application:** Since our project involved many fields and relationships, it was challenging to model the back end data model to fit our application
6. **Transforming data before inserting in Mongo:** The data had to be cleaned and transformed at several layers before inserting into MongoDB.Also, there were numerous debugging attempts on the data to find junk data being fed to the application, causing it to the crash

**Future Scope:**

* Add REST API - We already have a web service deployed by azure. We need to connect our application with the web api, so that we can compute real time recommendations for a user.
* Add Neo4j - Connecting mongo server with the neo4j server, so that instant virtualization is achieved.

**Conclusion:**

Despite of the challenges faced while building profile based recommendation, but the suggested listings were approximately 89% accurate. Being Airbnb user's ourselves, we would like Airbnb to provide such profile based recommendations for better customer experience