Final Project on

**Matchbox Recommendation System: A case study on banglaflix.com.bd**

Instructor: **Ashok Rangaswamy**

Submitted By:

|  |  |
| --- | --- |
| Name | Email |
| M A A Mehedi Hasan | [mamehedi.hasan@gmail.com](mailto:mamehedi.hasan@gmail.com) |
| Babul Miah | [Babul.cu.cse35@gmail.com](mailto:Babul.cu.cse35@gmail.com) |

12-January-2017

Belancer, Dhaka

# Introduction

Recommender systems or recommendation systems (sometimes replacing "system" with a synonym such as platform or engine) are a subclass of [information filtering system](https://en.wikipedia.org/wiki/Information_filtering_system) that seek to predict the "rating" or "preference" that a user would give to an item.

We are working on a video streaming app (banglaflix.com.bd) where thousands of bangla movies, video songs, dramas, and documentaries are there. Objective of this project is to implement a content recommender system in banglaflix app. We will use matchbox recommender for this. This experiment demonstrates the use of the Matchbox recommender modules to train a recommender engine. We use a pure collaborative filtering approach: the model learns from a collection of users who have all rated a subset of a catalog of contents. Matrix factorization allows us to infer from this latent user preferences and content traits. These preferences and traits are then used to predict what rating a user will give to unseen content, so that we can recommend content that the user is most likely to enjoy.

# Data Understanding

Banglaflix is a LIVE app and users are providing their ratings. For this project we have extracted content ratings and content metadata in csv format. There are two datasets required for matchbox recommender:

* 1. User rating: We have total 18,996 user rating data
  2. Content metadata:

**Sample Data Format:**

Match box recommender uses data triplet in <user, content, rating>

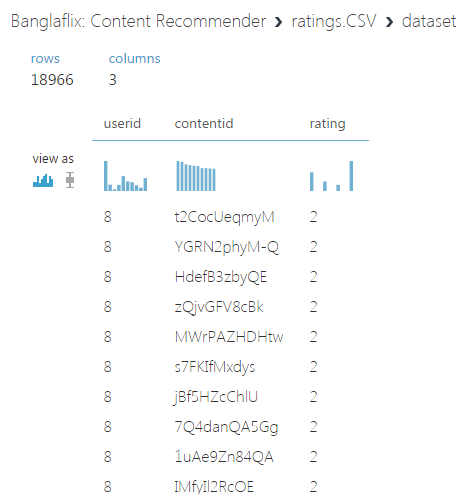


Image: - User rating data

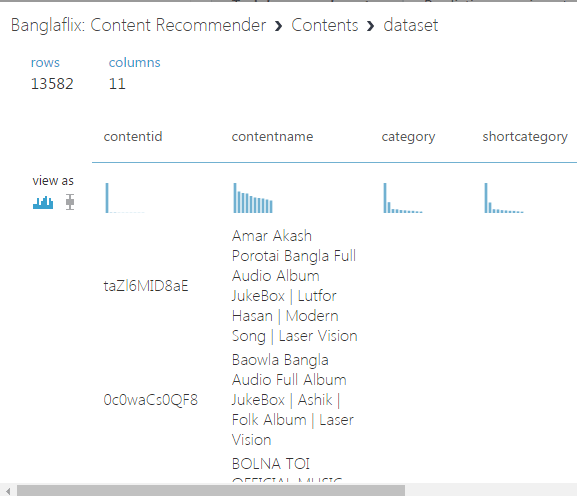


Image: - Content metadata

# Model Description

The training data has 18,996 unique ratings for 13,582 contents by 18,996 users. Each instance of data is a tuple consisting of a user identifier, a movie identifier, and the rating. First, we need to prepare the data for use with the Train Recommender module. It requires triplets in this format: <user, content, rating>.

1. The rating field is an integer
2. The Train Recommender module requires that the input contain three fields used for training, so we use Columns Selection to select only the user ID, Content ID, and rating fields.
3. We removed conflicting data using Remove Duplicate Rows.
4. We fit parameters on one set of data and test accuracy on a remaining set. In a collaborative filtering approach we need to make sure that we can learn something about each user and each item, so we cannot simply take a random sample of all the observations. Fortunately, Azure ML Studio provides a special **Recommender split**option in the **Split**module that lets you control how the train and test samples are selected.

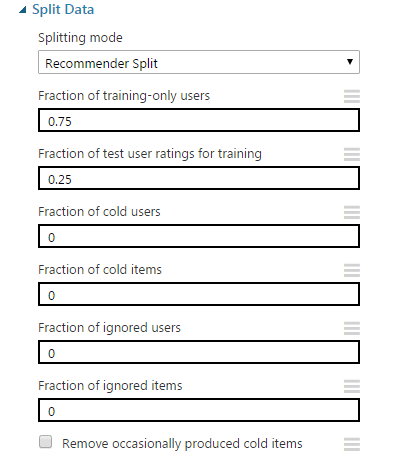


Image: - Split option

For this experiment, we used these settings:

* Fraction of training-only users: 0.75. This means that for 75% of the users we will use all the ratings to train. For the other 25% we will hold out some ratings for testing.
* Fraction of test-user ratings for testing: 0.25. For each user in the test group, we will hold out 25% of that user's ratings for testing the model.
* Fraction of cold users: 0. Cold users are users for whom we have no prior training data. In general the Matchbox algorithm can use optional user metadata to make recommendations for users even before we've seen a single rating. However, for this problem we do not have user metadata, so we will not evaluate on cold users.
* Fraction of cold items: 0. We will treat cold items the same as cold users, and evaluate only on movies for which we've received ratings.
* Fraction of ignored users: 0. In some cases we might want to test an algorithm or settings on a subset of the data. Here we'll train on the full set.
* Fraction of ignored items: 0. Same as for users.

# Model Deployment

We deployed the model as a web service in azure ml studio. This model takes 2 variables as parameters which are userid and contentid and gives the probable rating that could be given by the user.

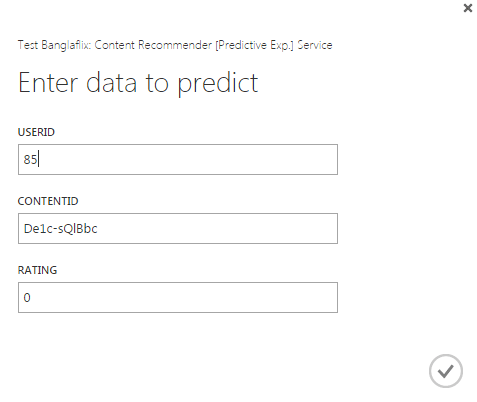


Image: - Web service input

Below is output extracted from the web service.

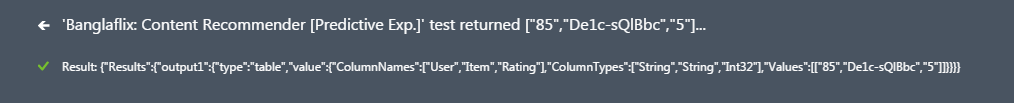


Image: - Web service output

# Conclusion

Matchbox recommender system is a very powerful system which can be used for predicting customers’ rating. We used this for banglaflix app as a case study. We will enrich the content meta data to get a better recommendation from matchbox system.