# MLND-Capstone project

## Book Recommender System Proposal

***1. Domain background**

In 2010-2012, we created more data than the the entire history. *(1) there is lot of information bombardment on the internet users. In the midst of this information abundance, Recommender Systems serve as agents that help users in getting the relevant information. Increasingly, lots of people are using internet as a source of information for making comparative analysis of products that they would like to buy online. As such lot of websites are also offering books as one of the product and some sites are dedicated websites for online book shopping, People are increasingly using Internet to discover the books that they will like.

***2.Problem statement**

As websites are using recommender systems widely, online book buyers are beginning to see books they did not know existed but were 'recommended' to fit their individual tastes. As a avid reader,I want to build a book recommender system.

***Datasets and inputs**

Collected by Cai-Nicolas Ziegler in a 4-week crawl (August / September 2004) from the Book-Crossing community with kind permission from Ron Hornbaker, CTO of Humankind Systems. Contains 278,858 users (anonymized but with demographic information) providing 1,149,780 ratings (explicit / implicit) about 271,379 books.

BX-Users Contains the users. Note that user IDs (`User-ID`) have been anonymized and map to integers. Demographic data is provided (`Location`, `Age`) if available. Otherwise, these fields contain NULL-values.

BX-Books Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (`Book-Title`, `Book-Author`, `Year-Of-Publication`, `Publisher`), obtained from Amazon Web Services. Note that in case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavours (`Image-URL-S`, `Image-URL-M`, `Image-URL-L`), i.e., small, medium, large. These URLs point to the Amazon web site.

BX-Book-Ratings Contains the book rating information. Ratings (`Book-Rating`) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.

***4. Problem solution**

Recommendations made by websites are not made at random, but are based on other similar readers' preferences or purchase histories. Recently this phenomenon is becoming a powerful marketing tool that retailers deploy to meet reader expectations and generate sales through up-selling or cross-selling. And the engine to generate recommendations is powered by algorithm based Recommendation Systems using an array of techniques such as Collaborative Filtering and Markov Chains.

The goal of this project is to build a Recommendation System for book buyers through Collaborative Filtering, Collaborative filtering produces recommendations based on the knowledge of users' attitude to items, that is it uses the "wisdom of the crowd" to recommend items.

***5.Benchmark model**

Create a model that makes recommendations using item popularity. When no target column is provided, the popularity is determined by the number of observations involving each item. When a target is provided, popularity is computed using the item's mean target value. When the target column

contains ratings, for example, the model computes the mean rating for each item and uses this to rank items for recommendations. It has no personalization but can be a good benchmark model.(3)*

***6.Evaluation Metrics**

A distance metric commonly used in recommender systems is cosine similarity, where the ratings are seen as vectors in n-dimensional space and the similarity is calculated based on the angle between these vectors. Cosine similarity is recommended to use when there is a sparse data. A problem with Cosine similarity is that it does not consider the differences in the mean and variance of the ratings made to items. On the other hand In our case as we have a dataset with high sparsity so We are going to use cosine similarity .

***7.Project design (workflow)**

(1)Data cleaning with pandas, sframe

(2) data visualization with graph lab

(3)cosine similarity to build recommender with graph lab

(4) cross-validation measures how good the model's performance is.

The most widely used evaluation measurement is the RMSE or Root Mean Squared Error. It is a straightforward difference measurement on predicted vs expected rating value. In other words RMSE measures how good the model's prediction is. The lower the RMSE the closer the prediction is to the actual rating.

Another evaluation measurement is **precision** is the fraction of relevant instances among the retrieved instances, while **recall** is the fraction of relevant instances that have been retrieved over total relevant instances

Of the three evaluation metrics, we chose RMSE as the key criteria to make a judgement on the quality of the model. This is primarily based on the context of the problem where the accuracy was important so as to generate models

(5)The final Recommender Model

# refrences

(1)https://www.forbes.com/sites/bernardmarr/2015/09/30/big-data-20-mind-boggling-facts-everyone-must-read/#f2b3ff117b1e

(2) http://www2.informatik.uni-freiburg.de/~cziegler/BX/

(3)https://turi.com/products/create/docs/generated/graphlab.recommender.popularity_recommender.create.html#graphlab.recommender.popularity_recommender.create