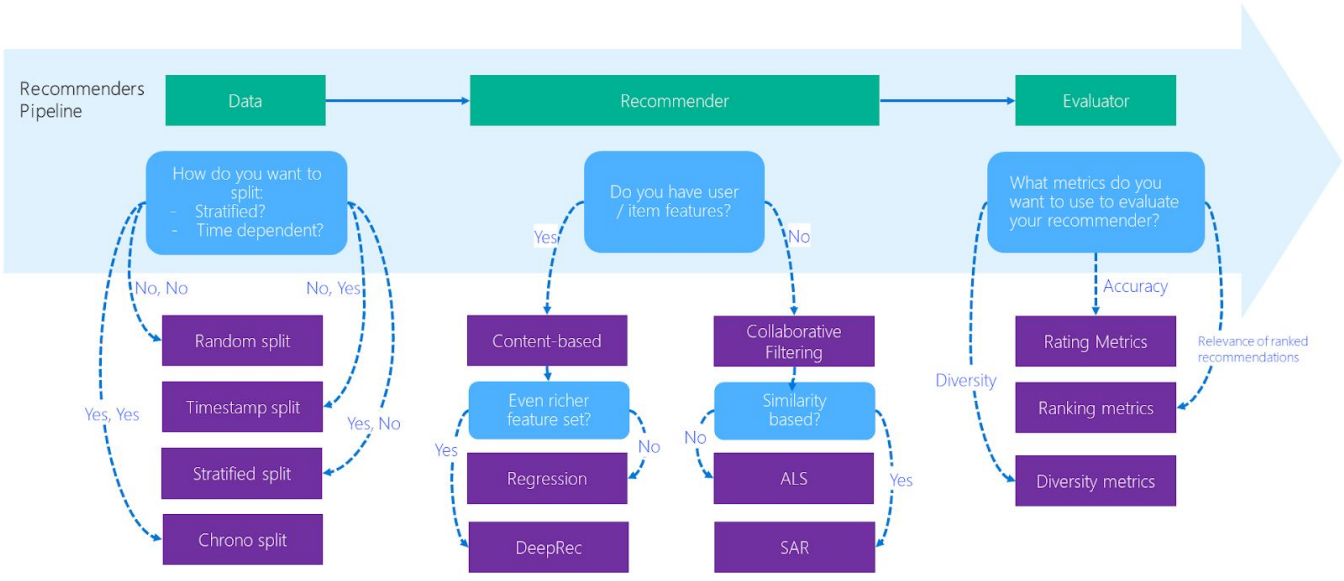
Algorithmic Marketing
Spring Semester 2020

INSTRUCTOR: Sri Krishnamurthy
analyticsneu@gmail.com

Recommendation systems

You are working at Snackfair, a custom snack shipping company that only has an online presence. The company competed against many online subscription services (https://www.mysubscriptionaddiction.com/best-snack-subscription-boxes) and wants to differentiate themselves with their superior recommendation system. The Data science team is interested in upgrading their recommendation system and have shortlisted 11 methods they could use. The have asked your team to put together a prototype of a recommendation system for one of the systems(See Appendix for team allocations). They have also reached out to 10 other teams and want to build a prototype using Streamlit to illustrate how to use it.



The goal of this assignment is to prototype, build, evaluate and deploy a recommendation system.

1. Since the company recognizes there are many approaches, the goal of this assignment is to replicate an experiment, understand when to use the algorithm using the default data given in https://github.com/microsoft/recommenders. For the allocated recommender, set up a working python environment and execute the algorithms.
2. In addition, we want to have a prototype that covers 0-3 stages of a typical recommendation workflow (https://github.com/microsoft/recommenders/tree/master/notebooks). Create or modify notebooks to customize the four stages (0-3) for your recommender using the examples provided in https://github.com/microsoft/recommenders/tree/master/notebooks
3. You are also expected to propose a sample schema of the data needed for building a snack recommendation system. Create a sample dataset in csv format and illustrate how your recommender can be used for that dataset. You can be as creative as you want. You can also use python packages like Faker to create a dataset. Also check out Kaggle's datasets if you can use any of the existing datasets as a proxy.

Deliverables (Due March 27 6.00pm):

1. A 2-5 page report in https://github.com/googlecodelabs/tools format to illustrate your understanding of how various algorithms are integrated in websites.
2. Be as technical as possible and discuss what algorithms and frameworks have facilitated the development of the site.
3. You will be given 5-10 minutes to present your company analysis in class on March 27th

Appendix:

Recommendation systems team allocations (See https://github.com/microsoft/recommenders)

TEam	Algorithm	Environment	Type	Description
1	Cornac/Bayesian Personalized Ranking (BPR)	Python CPU	Collaborative Filtering	Matrix factorization algorithm for predicting item ranking with implicit feedback
2	Extreme Deep Factorization Machine (xDeepFM)*	Python CPU / Python GPU	Hybrid	Deep learning based algorithm for implicit and explicit feedback with user/item features
3	Factorization Machine (FM) / Field-Aware FM (FFM)	Python CPU	Content-Based Filtering	Algorithm that predict labels with user/item features

4	FastAI Embedding Dot Bias (FAST)	Python CPU / Python GPU	Collaborative Filtering	General purpose algorithm with embeddings and biases for users and items
5	LightGBM/ Gradient Boosting Tree*	Python CPU/ PySpark	Content-Based Filtering	Gradient Boosting Tree algorithm for fast training and low memory usage in content-based problems
6	Neural Collaborative Filtering (NCF)	Python CPU / Python GPU	Collaborative Filtering	Deep learning algorithm with enhanced performance for implicit feedback
7	Restricted Boltzmann Machines (RBM)	Python CPU / Python GPU	Collaborative Filtering	Neural network based algorithm for learning the underlying probability distribution for explicit or implicit feedback
8	Riemannian Low-rank Matrix Completion (RLRMC)*	Python CPU	Collaborative Filtering	Matrix factorization algorithm using Riemannian conjugate gradients optimization with small memory consumption.

9	Simple Algorithm for Recommendation (SAR)*	Python CPU	Collaborative Filtering	Similarity-based algorithm for implicit feedback dataset
10	Surprise/Singular Value Decomposition (SVD)	Python CPU	Collaborative Filtering	Matrix factorization algorithm for predicting explicit rating feedback in datasets that are not very large
11	Vowpal Wabbit Family (VW)*	Python CPU (online training)	Content-Based Filtering	Fast online learning algorithms, great for scenarios where user features / context are constantly changing