ORPHEUS:

A MULTI-USER MUSIC RECOMMENDATION SYSTEM

BY THE DUMMY VARIABLES

OUTLINE

Motivation

Process Flow

Recommender Systems

Collaborative Filtering

Model testing

Aggregation strategy

Final product

Matrix Magic

MOTIVATION

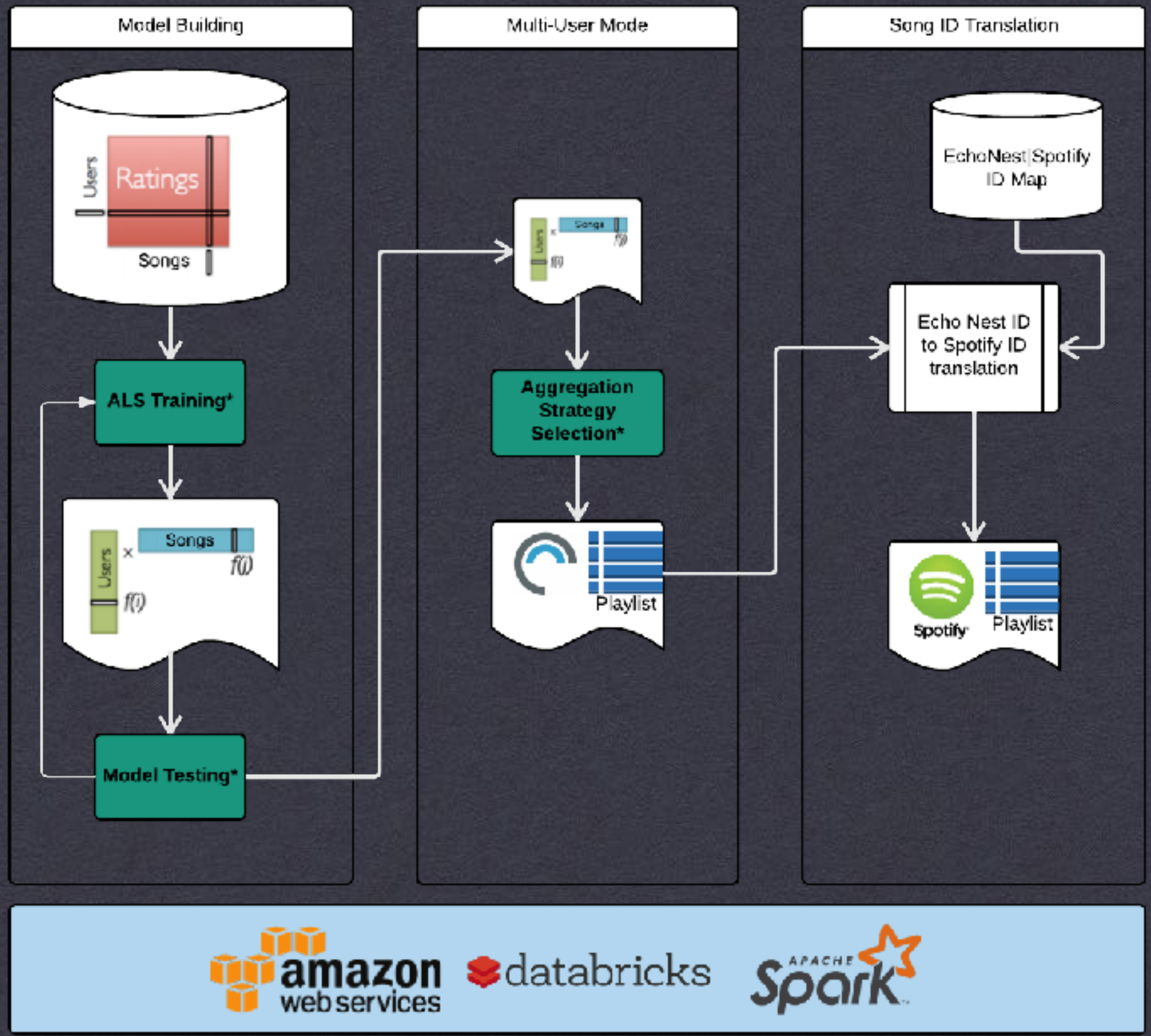
- Create a playlist for multiple users with different tastes and preferences and provide a uniformly fantastic listening experience using:
    - Collaborative filtering on implicit feedback
    - Aggregation strategy
    - Multi-users spotify accounts
    - Flask

THE DATASET

The Echo Nest Taste Profile Subset

- 3 GB
- 1,019,318 unique users
- 384,546 unique songs
- 48,373,586 unique user, song, playcount triplets
- On Average each user listens to 47 unique songs or 50% of users listen to 27 unique songs

PROCESS FLOW



RECOMMENDER SYSTEMS

ALS Training*

Types of recommender systems:

1. Content-based:
    a. Looks at characteristics of users or items. For example;
        i. Find similarity of songs based on music tempo,
        ii. Find similar users based on demographics.
    b. Difficult to collect
2. Collaborative filtering (CF):
    a. Modelling users' past behavior or the items' rating. For example;
        i. Identify users who rated the same item similarly.
    b. Cold start problem.

COLLABORATIVE FILTERING

ALS Training*

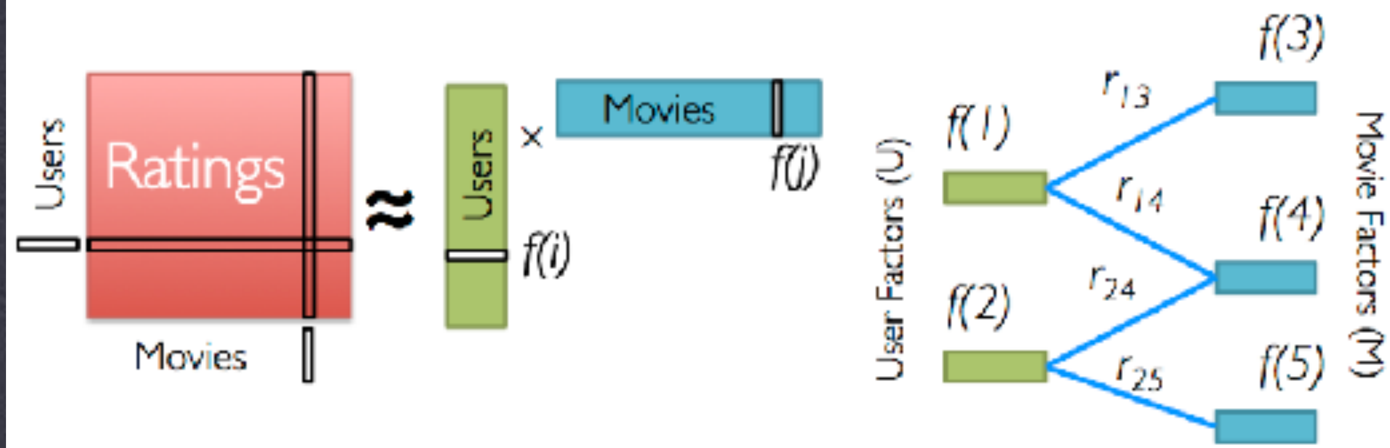
1. Explicit feedback
   a. User actively input their preference of the product. For example;
      i. Netflix star rating
2. Implicit feedback
   a. Based in consumption of the item.
      i. Number of times a song is heard.
      ii. Repeat purchase of product.

ALS Training*

COLLABORATIVE FILTERING

LATENT FACTOR MODEL:

Low-Rank Matrix Factorization:



Iterate:

EXPLICIT MODEL

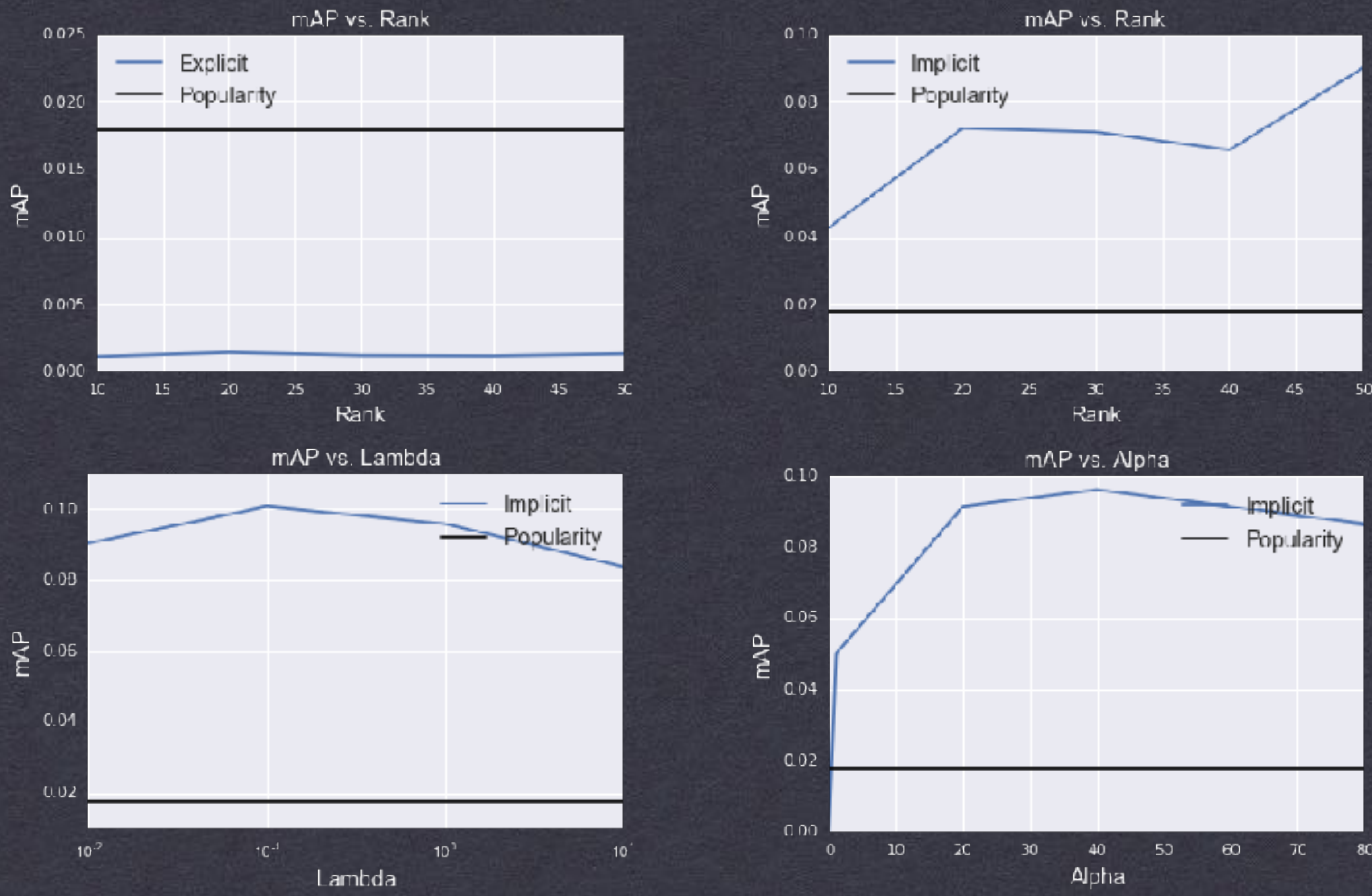
$$f[i] = \arg\min_{w \in \mathbb{R}^d} \sum_{j \in \mathrm{Nbrs}(i)} \left(r_{ij} - w^T f[j]\right)^2 + \lambda ||w||_2^2$$

IMPLICIT MODEL

$$\min_{x_\star, y_\star} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

$$c_{ui} = 1 + \alpha r_{ui}$$

MODEL TESTING



Model Testing*

MODEL TESTING

20	↓1	WiscoDisco	0.11206	5	Tue, 26 Jun 2012 01:21:55 (-28.5h)
21	↓1	sedielem	0.10952	11	Thu, 09 Aug 2012 17:52:45 (-50.4d)
22	↓1	Francie	0.10812	3	Fri, 08 Jun 2012 11:27:52
23	↓1	MML	0.10778	16	Wed, 06 Jun 2012 07:33:46 (-15.3h)
24	—	Superman89	0.10117	5	Fri, 20 Jul 2012 15:20:35 (-18.4h)
25	—	eggershead	0.09923	5	Thu, 17 May 2012 00:41:58
26	—	AppleCakeMining	0.09655	12	Mon, 21 May 2012 10:29:26 (-34.8h)
27	—	Andrew Ostapets	0.09132	17	Sun, 29 Jul 2012 21:13:41 (-10d)
28	↑69	seventeen	0.08998	4	Thu, 09 Aug 2012 16:32:37
29	↓1	holy shit	0.08657	13	Mon, 30 Jul 2012 03:09:46 (-46.1d)
30	↓1	jose	0.08657	6	Sat, 30 Jun 2012 17:33:16 (-14.4d)

25th out of 150 teams!

Aggregation Strategy Selection*

AGGREGATION STRATEGY

How to adapt recommendations to a group based on individual members' preference?

When forming recommendations for the group:

Average strategy: average happiness of the members.

Most pleasure: happiness of the most happy member.

Least misery: happiness of the least happy member.

Multiplicative: product of each member's happiness.

	INDIVIDUAL RECOMMENDATIONS				
USER	Lifehouse -	Lifehouse - From	Lifehouse - Falling In	Lifehouse - Storm	Lifehouse - All In
84250	0.95	0.77	1.00	0.97	0.64
92650	0.71	0.98	0.89	0.84	0.97
193650	0.90	0.86	0.86	0.75	0.88
	GROUP RECOMMENDATIONS				
Least Misery	0.71	0.77	0.86	0.75	0.64

AGGREGATION STRATEGY

Which strategy performs best?

What strategy to use based on group characteristics?

- Best strategy for a group with very similar members?
- Best strategy for a small group vs big group?

How to measure group satisfaction?

$$S(u,R) = \frac{\sum_{i \in R} \hat{p}(u,i)}{|R|}$$

$$S(g,R) = \frac{\sum_{u \in g} S(u,R)}{|g|}$$

Cluster users

- K-means clustering using latent features.
- Create homogeneous and heterogeneous groups of 3, 5, and 7 members.
- Derive S(g,R) for each aggregation strategy.

Aggregation Strategy Selection*

AGGREGATION STRATEGY ANALYSIS

Average aggregation strategy works best especially for heterogenous groups!

HOMOGENOUS: 3-members				
	LM	MP	AV	MU
means	0.51	0.49	0.56	0.52
sds	0.14	0.13	0.12	0.16
ANOVA p-value	0.548			

HOMOGENOUS: 5-members				
	LM	MP	AV	MU
means	0.48	0.45	0.53	0.51
sds	0.11	0.09	0.09	0.12
ANOVA p-value	0.0719			

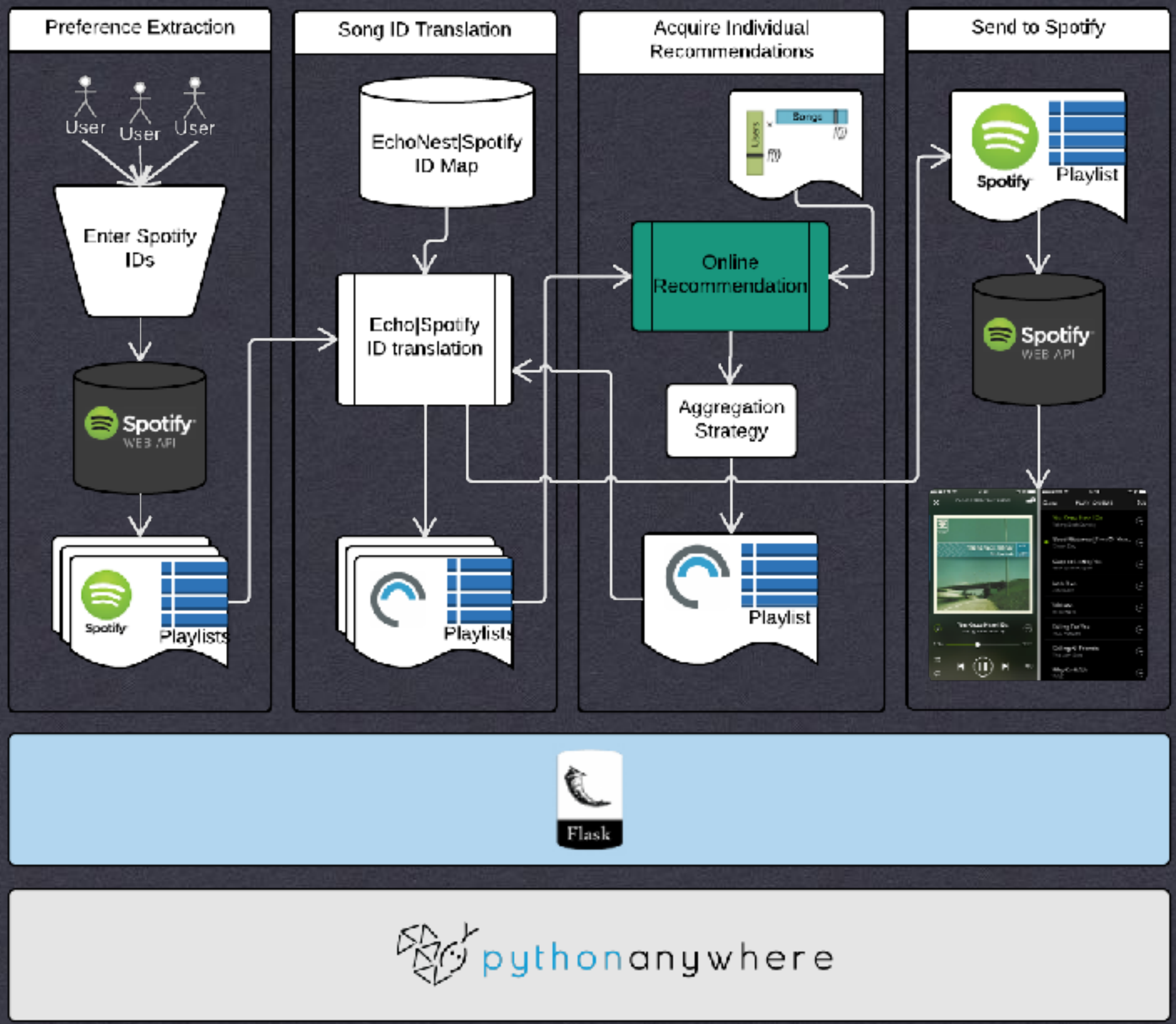
HOMOGENOUS: 7-members				
	LM	MP	AV	MU
means	0.51	0.46	0.56	0.52
sds	0.15	0.15	0.13	0.18
ANOVA p-value	0.277			

HETEROGENOUS: 3-members				
	LM	MP	AV	MU
means	0.39	0.38	**0.47**	0.35
sds	0.08	0.07	0.08	0.15
ANOVA p-value	**0.00481**			
TukeyHSD Post-hoc test: p-value < 0.05				
variables	**AV-**	**MU-**		

HETEROGENOUS: 5-members				
	LM	MP	AV	MU
means	0.25	0.23	**0.36**	0.19
sds	0.06	0.06	0.06	0.1
ANOVA p-value	**3.86E-10**			
TukeyHSD Post-hoc test: p-value < 0.05				
variables	**AV-LM**	**MU-**	**AV-**	**MU-**

HETEROGENOUS: 7-members				
	LM	MP	AV	MU
means	0.2	0.19	**0.32**	0.17
sds	0.04	0.04	0.04	0.07
ANOVA p-value	**8.83E-16**			
TukeyHSD Post-hoc test: p-value < 0.05				
variables	**AV-LM**	**AV-**	**MU-**	

FINAL PRODUCT



TRY ORPHEUS

CLICK ME!

REFERENCES

CARVALHO, L., MACEDO, H.: USERS' SATISFACTION IN RECOMMENDATION SYSTEMS FOR GROUPS: AN APPROACH BASED ON NONCOOPERATIVE GAMES (2013)

HU, Y., KOREN, Y., VOLINSKY, C.: COLLABORATIVE FILTERING FOR IMPLICIT FEEDBACK DATASETS

MASTHOFF, J.: GROUP RECOMMENDER SYSTEMS: COMBINING INDIVIDUAL MODELS (2011)

O' CONNER, M., COSLEY, D., KONSTAN, J.A., RIEDL, J.: POLYLENS: A RECOMMENDER SYSTEM FOR GROUPS OF USERS. ECSCW, BONN, GERMANY (2001)

SEGARAN, T.: PROGRAMMING COLLECTIVE INTELLIGENCE (2007) CHAPTER 2

SHANI, G., GUNAWARDANA, A.: EVALUATING RECOMMENDATION SYSTEMS

Model Testing*

MODEL TESTING

- How to Evaluate Implicit Model?
  - RMSE doesn't make sense!
  - Neither regression nor classification
- Mean Average Precision (mAP)
  - Precision-at-k (P)
    - Proportion of correct recommendations within top-k of the predicted ranking
  - Average Precision (AP)
    - Precision at each recall point k
  - Mean Average Precision (mAP)
    - Average over all users
  - Used in Million Song Kaggle Competition

Evaluate mAP

Train Users	Test Users (Hidden Songs)
	Test Users (Visible Songs)