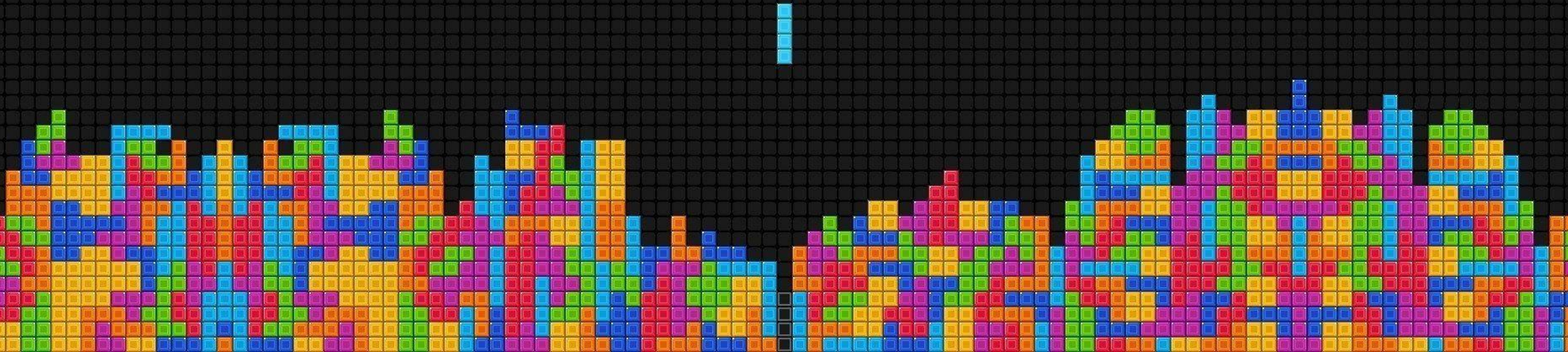


Steam Video Game Recommendation System

A Springboard Data Science Career Track Final Capstone Project

Joshua Ogden-Davis

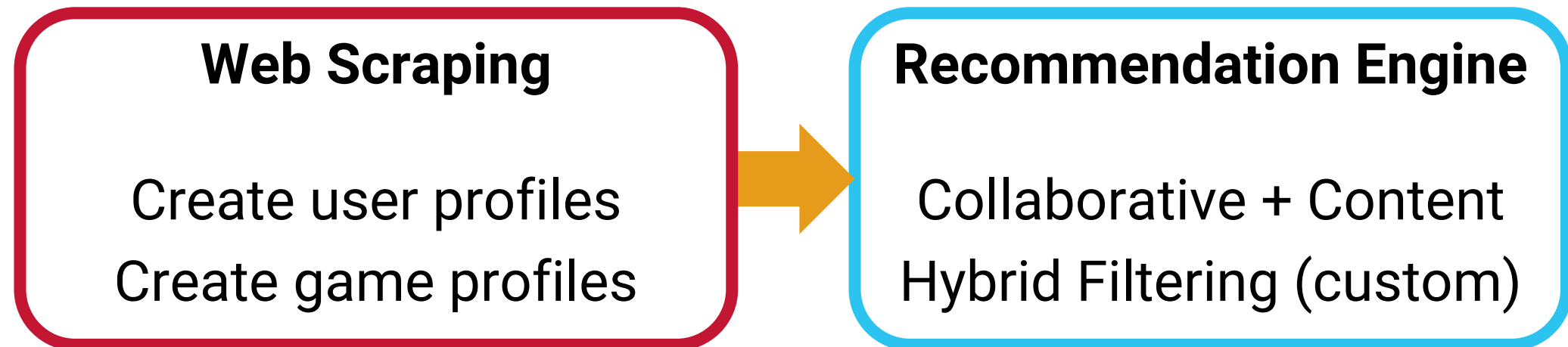
April 2024



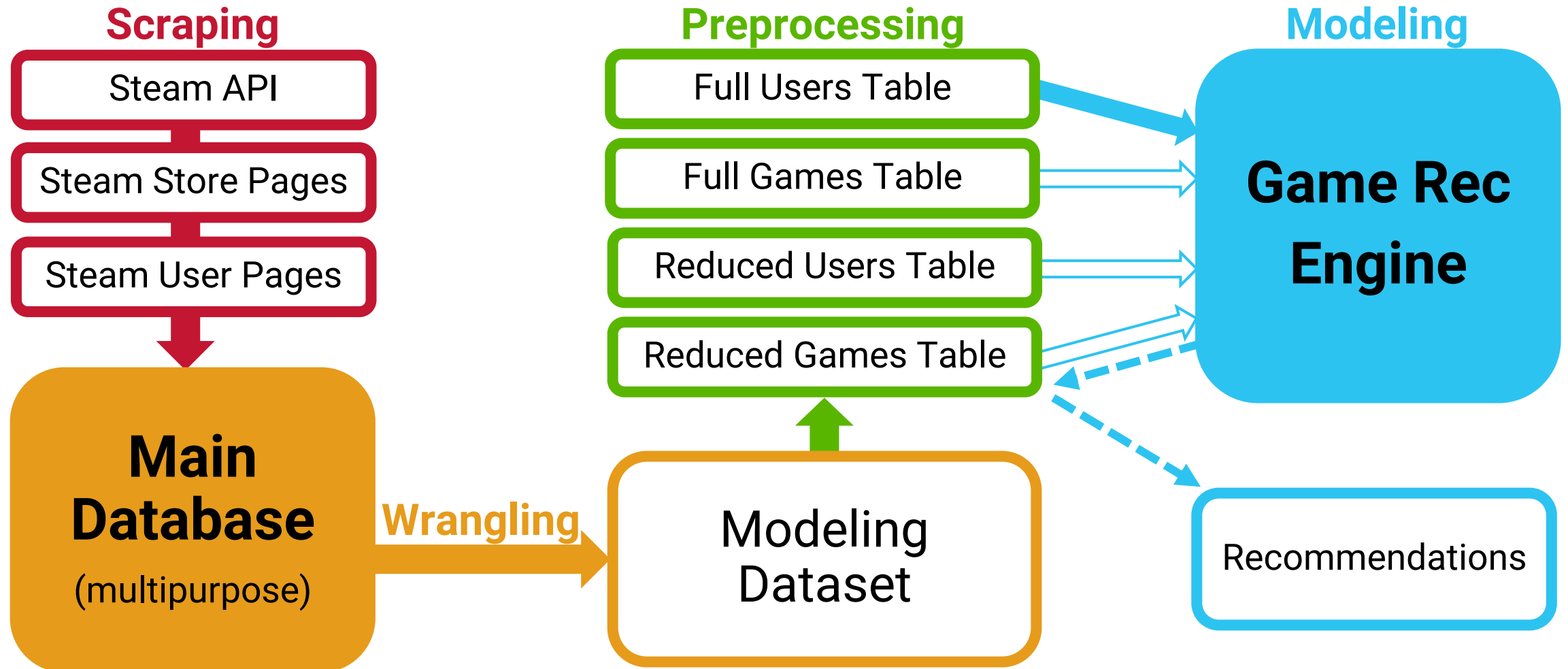
Purpose

1. **Generate an extensive dataset** from **publicly-available data**
2. Develop a **game recommendation system** for **existing users**

Two Main Sub-Projects



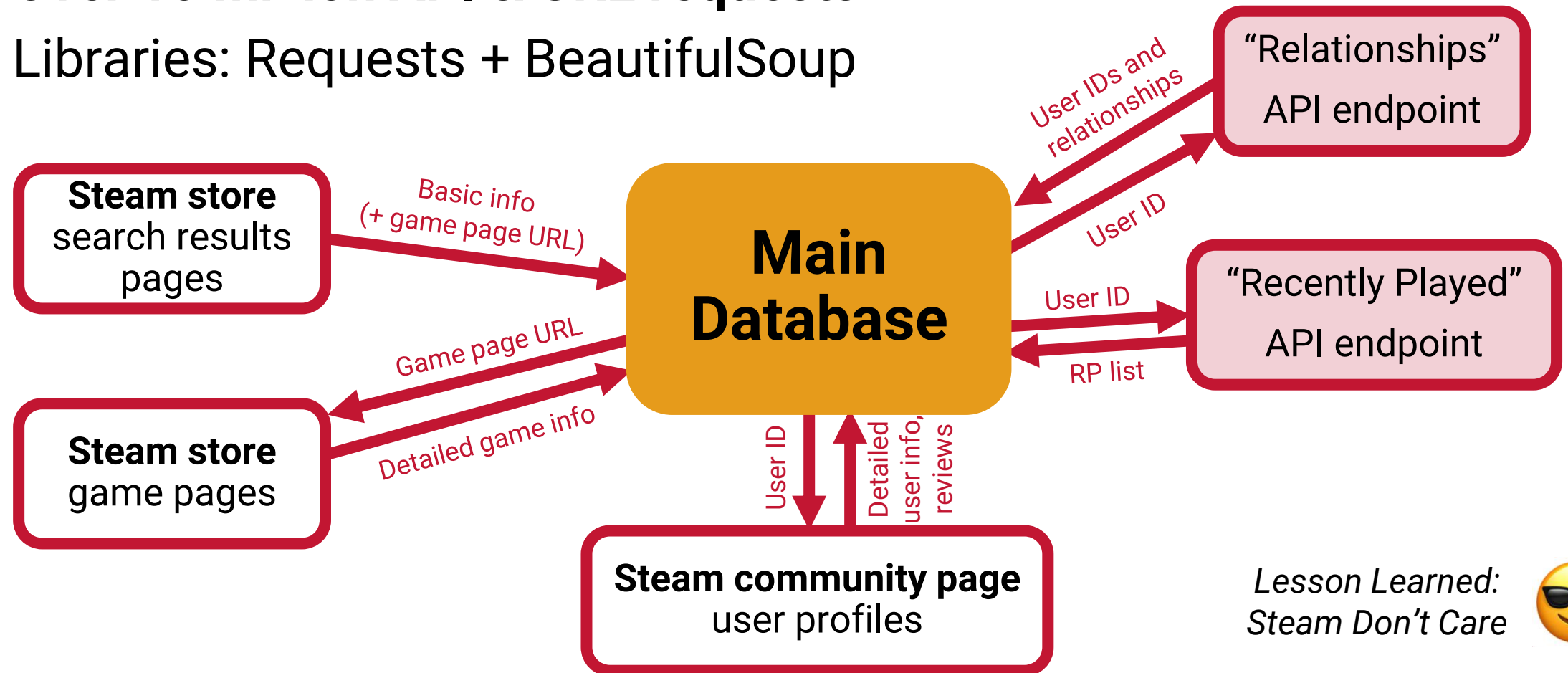
Overall Flow




Scraping Methods

Over 10 million API & URL requests

Libraries: Requests + BeautifulSoup



Lesson Learned:  Steam Don't Care

Database Schema

Games (100k, ~90%)	
app_id	int
title	
release_date	datetime
positive_review_percent	float
number_of_reviews	int
game_page_link	str (url)
price	int
tags	list(str)
tag_list	list(str)
developer	list(str)
publisher	list(str)
description	str
interface_languages	list(str)
full_audio_languages	list(str)
subtitles_languages	list(str)
english, german, french, spanish, brazilian, russian, italian, schinese, japanese, koreana, polish	int
date scraped	datetime

Data source: scraped from Steam store

Reviews (5.7mil)	
user_id	str
app_id	int
positive	int(0-1)
total_playtime	datetime
review_playtime	datetime
text	str
helpful_count	int
review_date	datetime
edit_date	datetime
date_scraped	datetime

Data source: scraped from Steam user profiles

Recently Played (3.6mil)	
user_id	str
app_id	int
playtime_2weeks	int
playtime_forever	int

Data source: Steam API

Relationships (100k)	
user_id 1	str
user_id 2	str
friend_since	datetime

Data source: Steam API

All Users (1.9mil, ~1.5%)	
user_id	str

Data source: Steam API

Gray background means the field was used for modeling.

Games Tables

Condensed Sparse Row Format

	Adventure	Indie	Fem. Protagonist	... (446)
game index 1	0	1	0	...
game index 2	0.8	0	1	...
game index 3	0	0	0	...
... (100k)

- “Reduced” games table omits games with <5 tags (arbitrary).
- Because CSR has no separate index, we must maintain:
 - Bidict of full table index <-> app_id
 - Bidict of full table index <-> reduced table index
 - (Steam already indexes tags starting at 1)

Index Hell



Users Tables

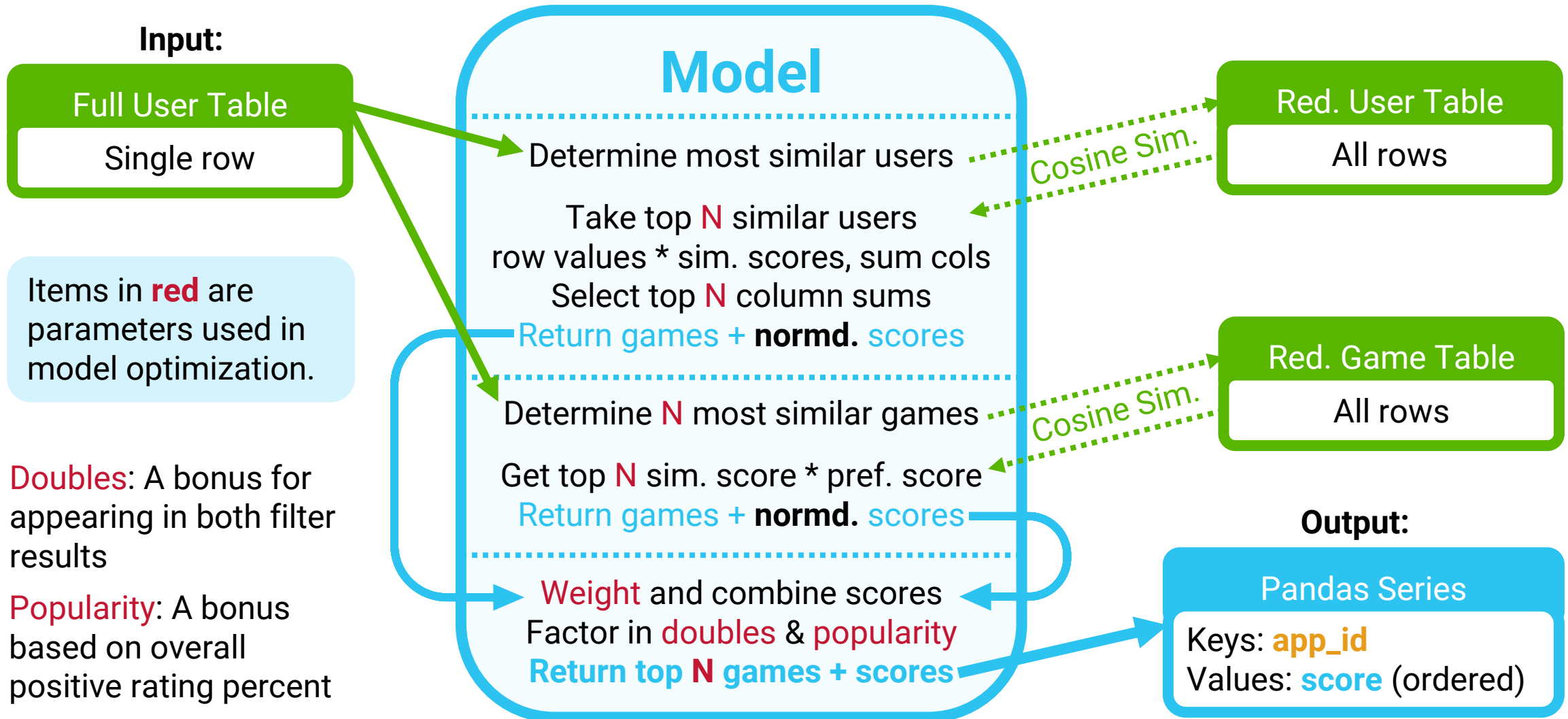
Condensed Sparse Row Format

	game 1 col index	game 2 col index	game 3 col index	... (100k)
user index 1	1	0	0.2	...
user index 2	0.2	0.2	0	...
user index 3	0	-1	0.2	...
... (1.9mil)

- “Reduced” users table omits users with <10 games (arbitrary).
- Values indicate levels of preference:
 - Recently played: 0.2
 - Positive review: 1
 - Negative review: -1

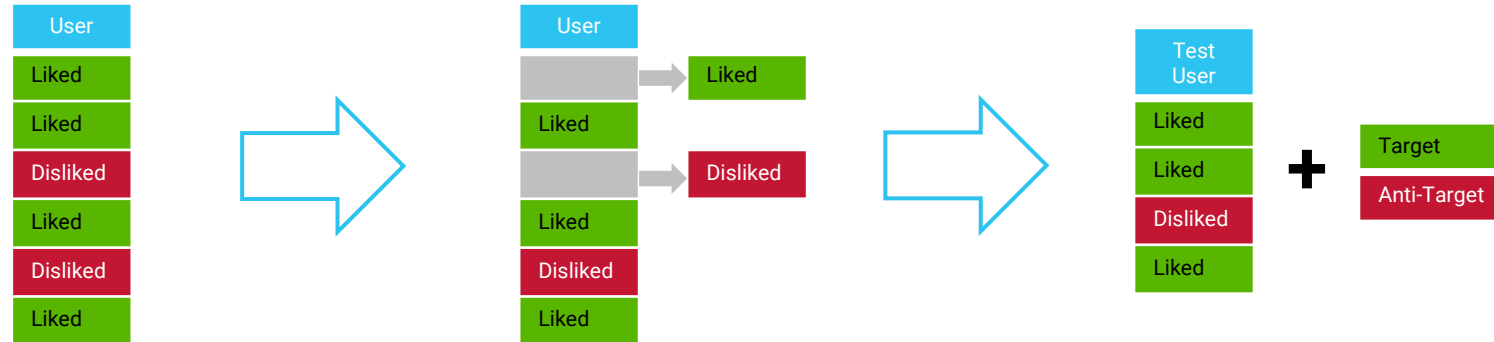


Modeling Flow



Testing & Evaluation

1. **Create test users** by removing one **liked** game and one **disliked** game from random existing user profiles.



2. **Generate recommendations** for the test user.
3. **Compare the results** to the removed games.
 - If we recommended the **liked** game, score **+1**
 - If we recommended the **disliked** game, score **-1**

Results

Best parameters*:

'similar_user_limit': 250

'collab_filter_limit': 103

'content_filter_limit': 32

'double_bonus': 1.79

'popular_bias': 1.84

'ratio': 0.71

'recs': 20

* via BayesianOptimization()

Best model performance:

Test users: 100

Good recs: 30

Bad recs: 17

Total score: 13

Next Steps

- Continuously increase dataset
- Include more features (developer, publisher, date released, etc)
- Control for new vs old users (only have “recently” played game info)
- Develop a higher-resolution evaluation function (utilize ranking, etc)
- Implement PostgreSQL database
- Move to the cloud

Thank you!

Springboard Data Science Career Track Final Capstone Project

Joshua Ogden-Davis

April 2024

GitHub: [joshtod](#)

Email: ogden.davis@gmail.com

