# Recipe Recommendation Engine through Collaborative and Content Filtering

Jason Summer

2020

# Agenda

01 Executive Summary

02 Opportunity Overview

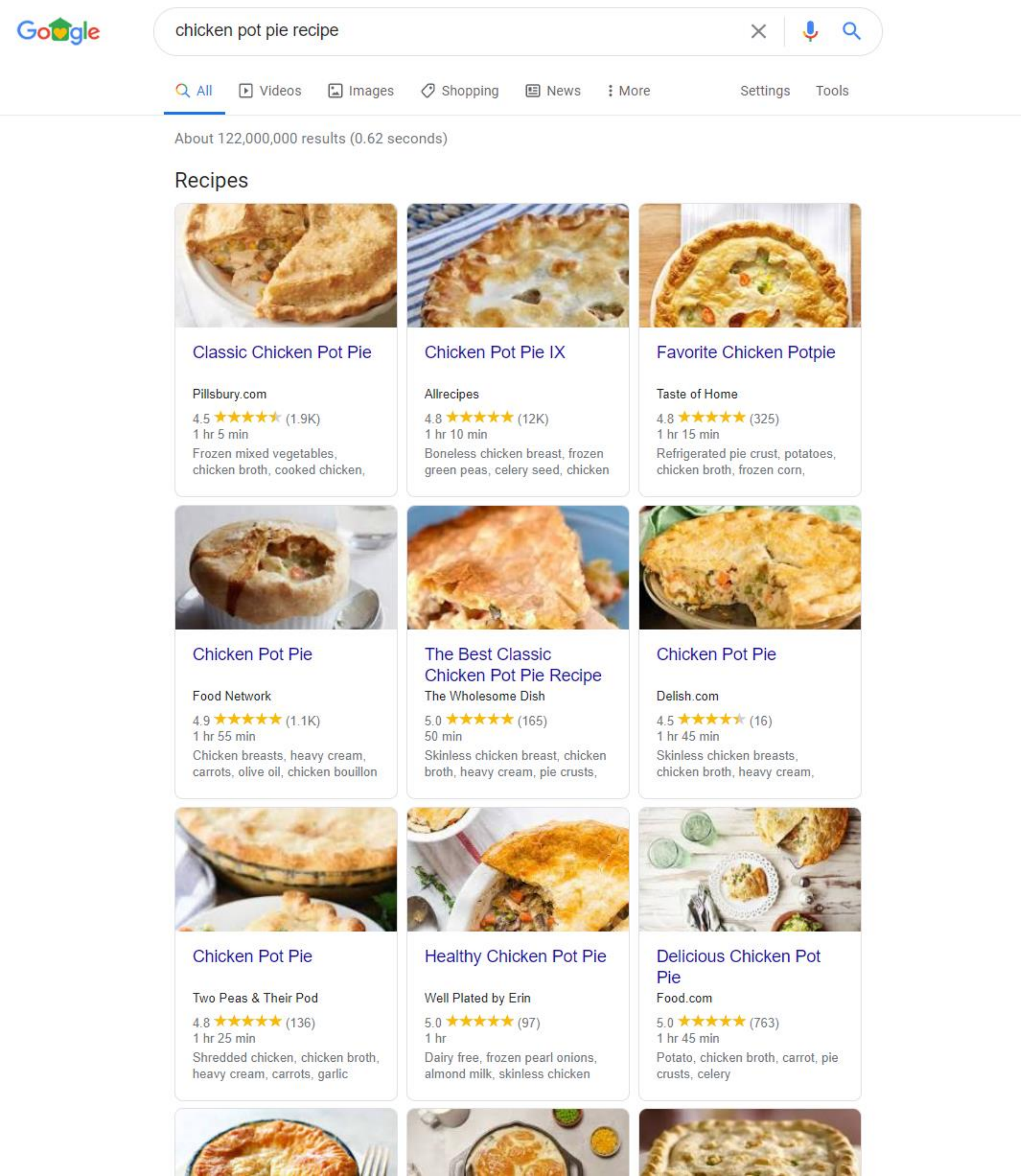
03 Data Introduction and Analysis

04 Collaborative Filtering

05 Content Filtering

06 Feature Engineering

06 Key Takeaways




# Executive Summary

## Drive loyalty and user growth of recipe platforms through intelligent recommendations

In the saturated market of recipe aggregation, organizations seeking to grow loyalty and expand their user base should consider machine learning techniques to simplify the customer experience. There is no shortage of quality recipes digitally available. However, simplifying the process to find these recipes and ensuring consistent quality of recipes is key.

The following analysis details how collaborative filtering merged with content filtering can be used to drive loyalty of an existing customer base. Furthermore, it details the effect of a saturated market on usership and how providing optimal first recommendations is crucial for growth.

# Agenda

# Opportunity Overview

Digital recipe aggregators should begin transforming their large networks into measured recommendations

**Large Recipe Library**

Ensure recipe offerings are diverse and well labeled

**Grocery List Generation**

Integrate ingredients into interactive shopping list

**Collaborative Filtering**

Recommend recipes based on similar users

**Content Filtering**

Relate recommendations to users' previous recipes

**User Submitted Ratings**

Incentivize frequent rating submissions and reviews

**Large User Base**

Host a large network of engaged customers

Websites and apps offering recipe libraries today should focus on taking the next step towards data-driven recipe suggestions. These organizations should seek to leverage large user networks for collaborative filtering and recipe descriptions for content filtering. Removing the need for a customer to browse recipes and instead simply trust a suggestion and create a shopping list could result in a key differentiation.

# Agenda

# Data Landscape

To create an explanatory proof of concept, data from Food.com was sourced via Kaggle.com. The data consist of 200K+ recipes and 1M+ recipe reviews covering 18 years of user interactions and uploads

### Recipe Identifier
Identifier to associate to recipe file for more information

### User Identifier
Unique identifier of individual that submitted rating of recipe

### Recipe Rating 0-5
Star rating from 0 to 5 indicating user's review of corresponding recipe

### Submission Date
Day stamp of when the rating was submitted


### Recipe Identifier
Unique identifier used to track corresponding ratings

### Recipe Name
User submitted name of the recipe upon creation

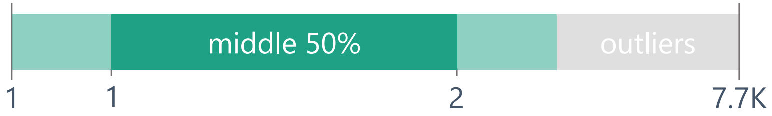
### Tags
Keyword tags scraped from food.com recipe pages

### Ingredients
Full listing of ingredients used in recipe

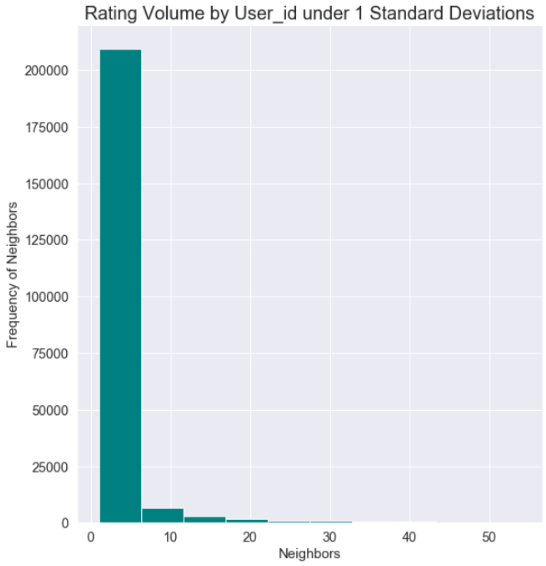
# Rating Submissions

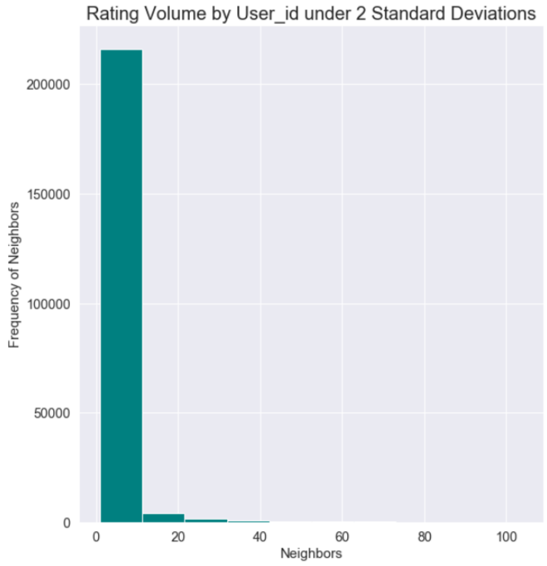
227K unique users rated 231K unique recipes, totaling over 1 million reviews. However, users tend to submit 1-2 ratings and recipes tend to receive 1-4 ratings, indicating very low engagement.

## Ratings by User

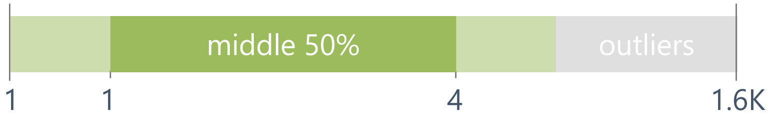



Dataset does not feature any cold start users. Generally users submit 1-2 ratings indicating high possible attrition and low loyalty.

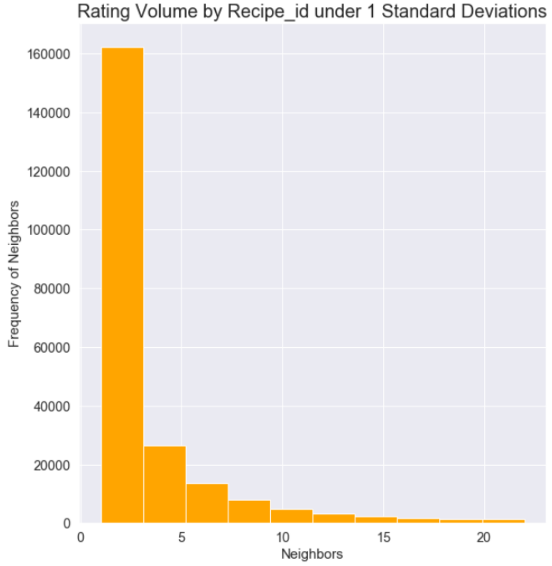


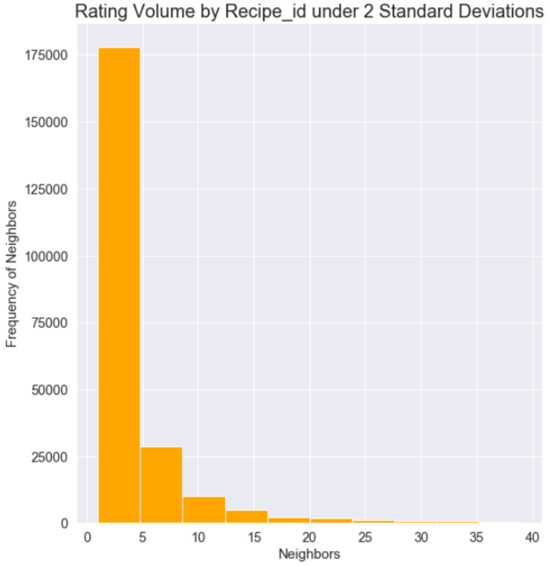




## Ratings by Recipe




Ratings by recipe shows less of a long tail but still generally low for most recipes.



# Rating Scores

While users generally only submit 1-2 ratings, these ratings seem to generally be between 4 and 5. Evaluating the distribution of common rating submissions shows an obvious tendency to submit positive ratings.

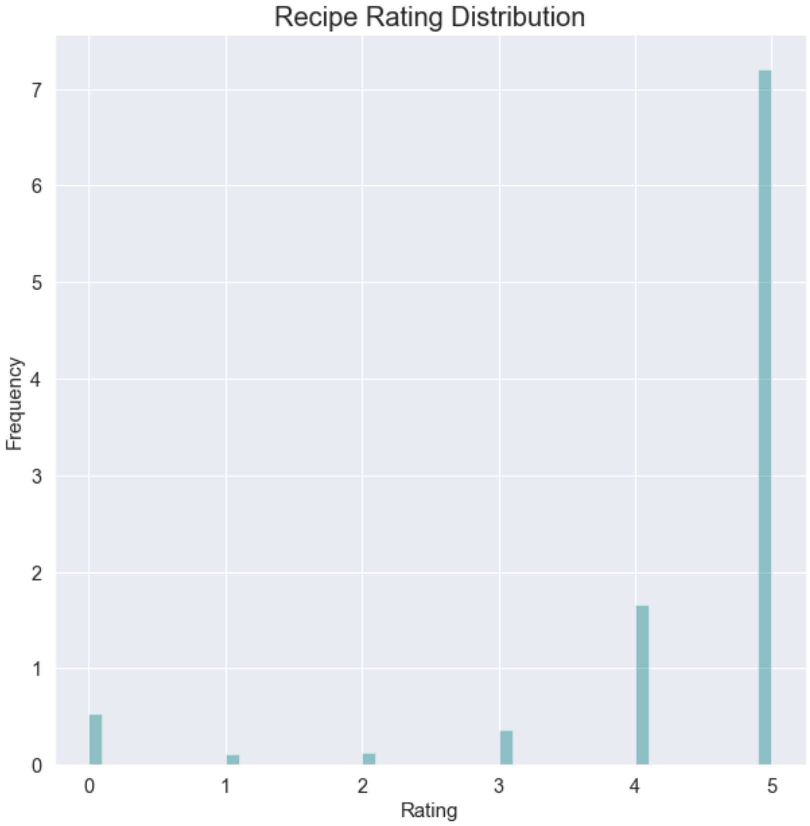
**89%** of ratings are 4 or higher

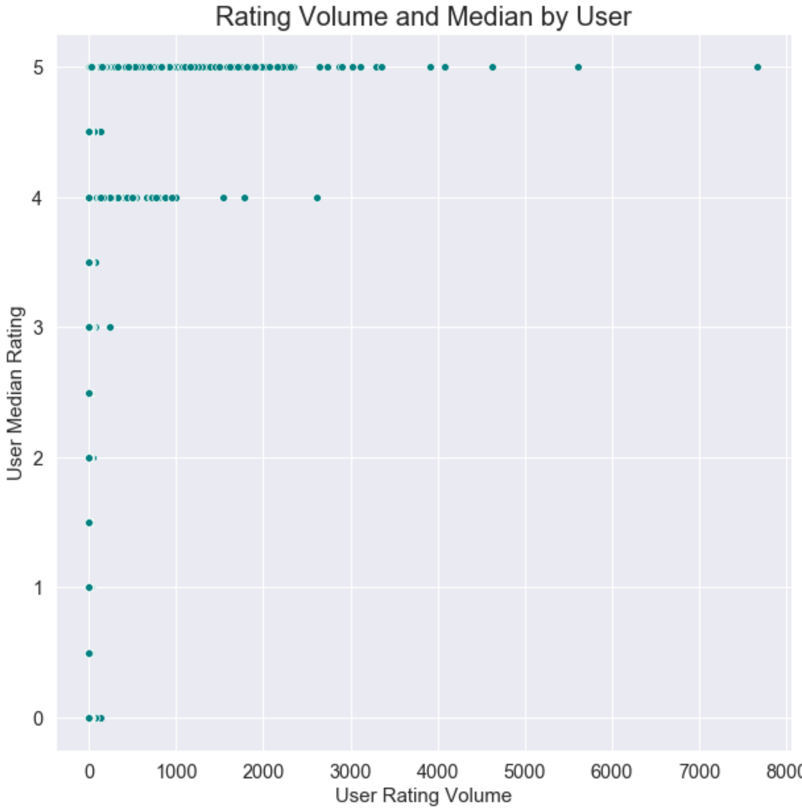
A user's tendency to submit high ratings increases as their rating submissions increase.

This increasing tendency to submit high ratings coupled with generally low engagement highlights key user behavior and opportunity.

A single "poorly selected" recipe can lead to swift user attrition highlighting the need for smart recommendations for new and low usage customers.

There is also an opportunity to drive engagement and loyalty of existing user base through personalized recommendations.




Ratings of 0 were updated to 0.1 for modeling purposes

# Agenda

01 Executive Summary

02 Opportunity Overview

03 Data Introduction and Analysis

04 Collaborative Filtering

05 Content Filtering

06 Feature Engineering

06 Key Takeaways

# Algorithm Selection

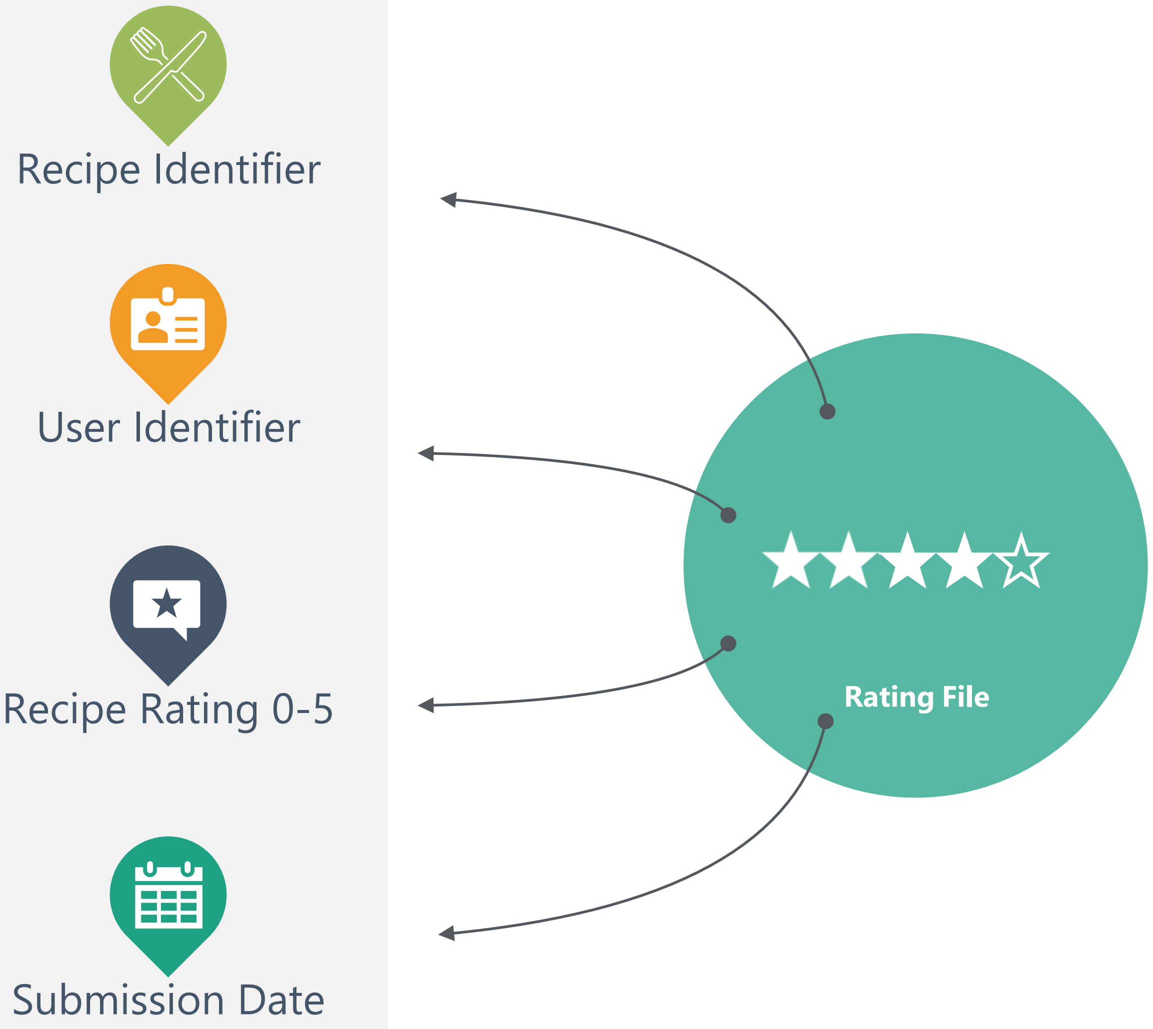
11 algorithms were compared for collaborative filtering and 2 were selected for further tuning and evaluation

The Surprise library for building and testing recommender systems was selected for this collaborative filtering effort. The specific library was selected due to its similarity to scikit-learn modules and read-to-use prediction algorithms.

The Surprise package offers 11 out of the box algorithms including baseline algorithms, neighborhood-based methods, and matrix factorization approaches. The recipe identifiers, user identifiers and ratings were sampled and used to compare the 11 algorithms.

| Algorithm | Mean RMSE | Training Time |
|---|---|---|
| SVD ++ | 1.0533 | 2.197224 |
| KNN Baseline | 1.054044 | 0.734534 |
| SVD | 1.054203 | 0.990171 |
| Baseline Only | 1.054441 | 0.079275 |

The top 4 algorithms (shown above) were very similar in their overall root mean squared error. After more precise training time evaluations were conducted, SVD++ was eliminated due to memory limitations. SVD and Baseline Only were selected for further tuning.




Ratings of 0 were updated to 0.1 for modeling purposes

# Algorithm Tuning

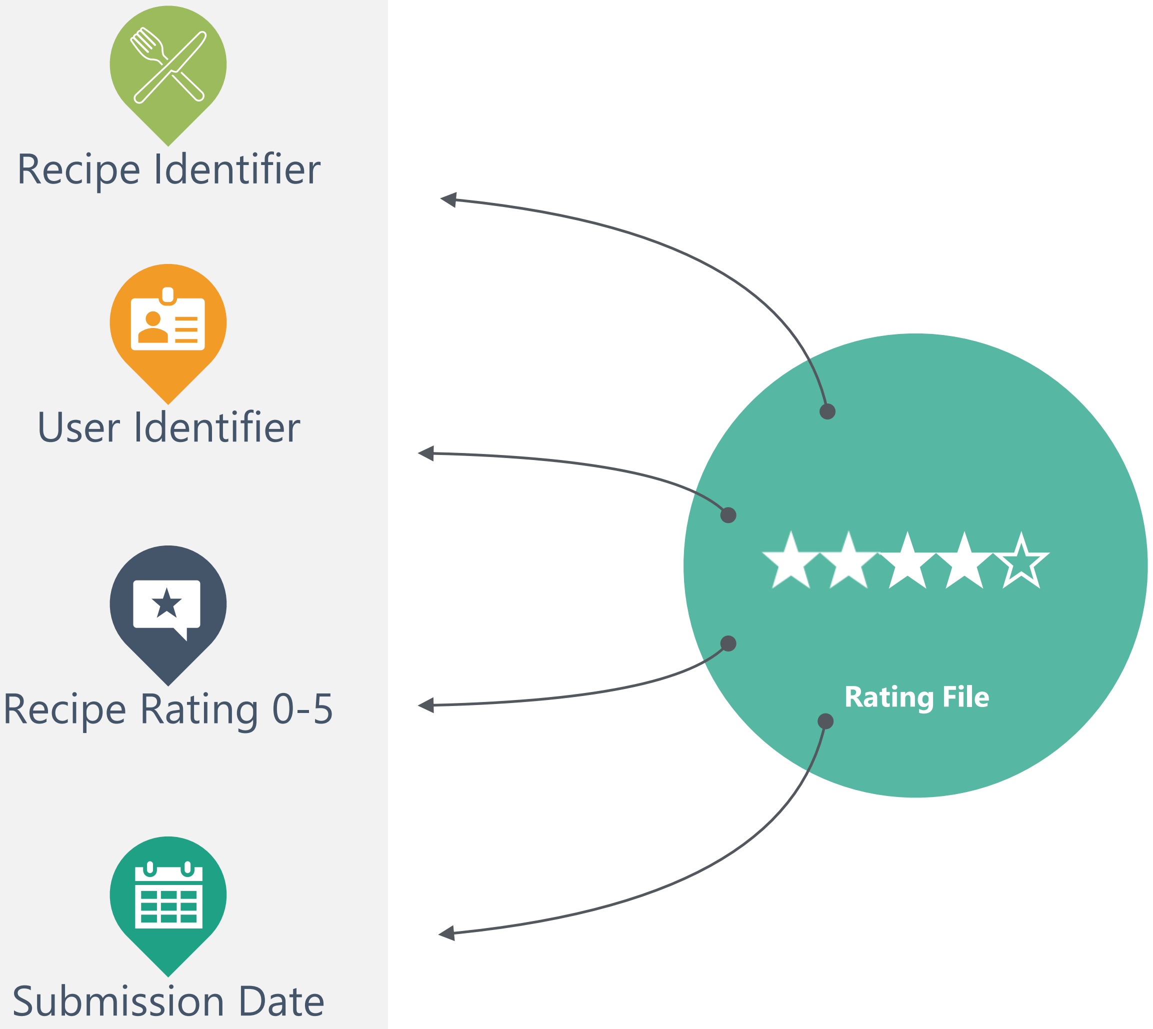
Using Surprise library's built in cross validation grid search module, the SVD and Baseline Only were tuned using a sample of the ratings data.

Hyperparameter tuning of the SVD and Baseline Only algorithms showed minimal improvement. However, optimal values were selected for each.

SVD uses stochastic gradient descent and attempts to characterize the data using fewer dimensions than the raw data. It provides 4 key parameters for tuning.

| Hyperparameter | Parameter Description | Value |
|---|---|---|
| n_factors | number of latent features used to identify similarities in preferences | 25 |
| n_epochs | number of iterations of the stochastic gradient descent used to minimize regularized squared error | 30 |
| ir_all | learning rate for all parameters | 0.003 |
| reg_all | regularization term for all parameters | 0.1 |

The Baseline algorithm offers two alternative methods to estimate and reduce the regularized squared error of each user and recipe combination. These methods were compared, and the stochastic gradient method performed slightly better than the alternating least squares method.



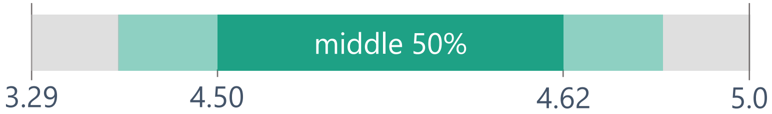

Ratings of 0 were updated to 0.1 for modeling purposes

# Prediction Performance

The ratings data were sorted ascending by submission date to allow the model to learn emerging trends.
Each model was trained on the initial 70% of ratings and evaluated on the remaining 30%.

## SVD Predictions

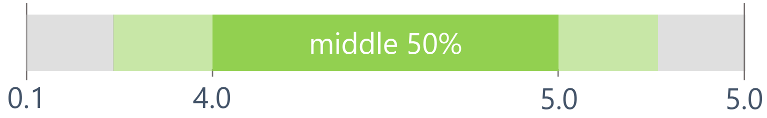
RMSE:
1.0307




## Baseline Predictions

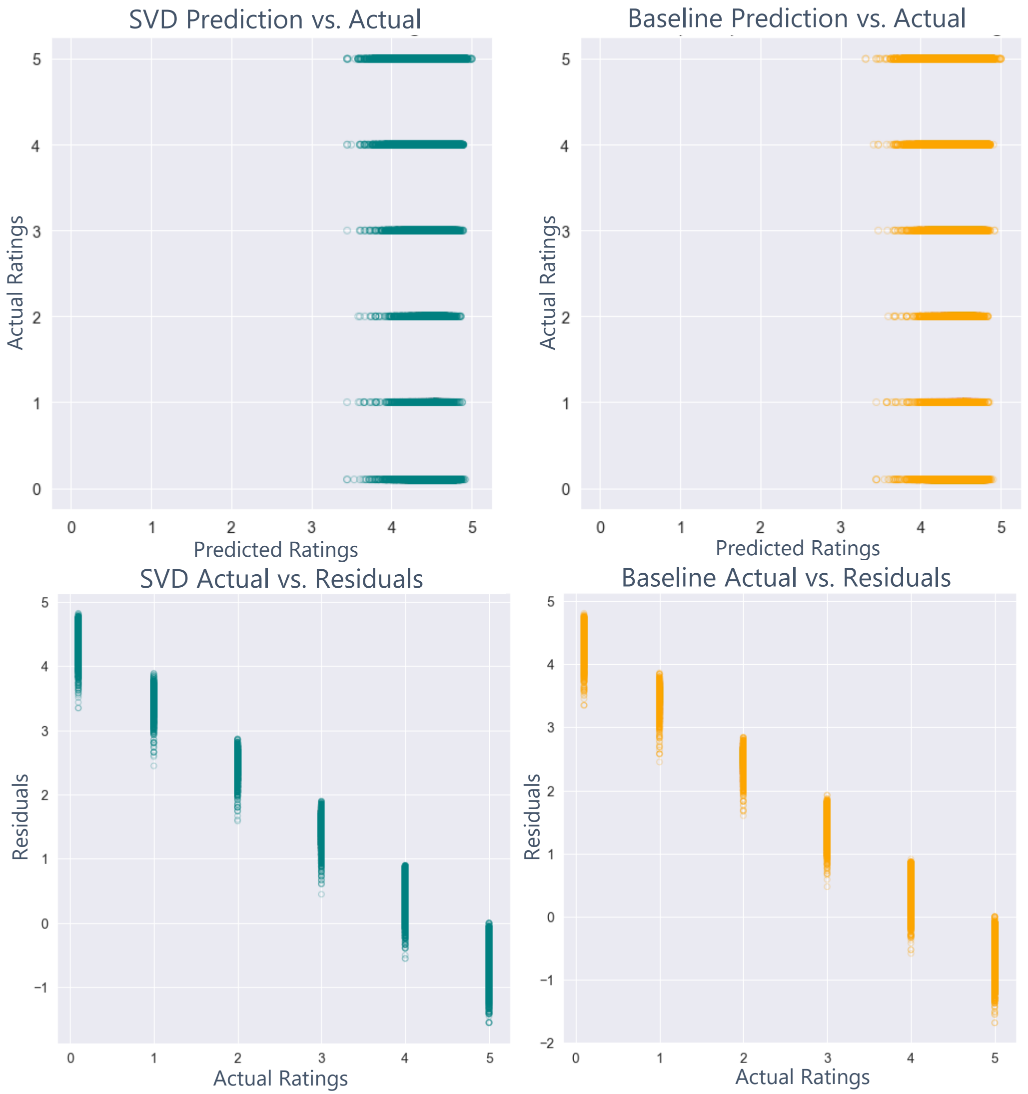
RMSE:
1.0306


## Actual Ratings




Both algorithms perform comparably well with a general error around 1. More interesting however was the apparent effect of imbalanced ratings. Predicted ratings are consistently above 3. As suspected, residuals decrease as actual ratings increase, further highlighting the tendency towards high ratings.

As a comparison, the minority ratings were bootstrapped to create a balanced dataset and the modeling was performed again. The RMSE and variance increased by 1.5x and 29x, respectively. The wider breadth of ratings using the bootstrapped samples confirms the change in behavior due to imbalanced ratings. See Appendix for outputs.




Ratings of 0 were updated to 0.1 for modeling purposes

# Agenda

01 Executive Summary

02 Opportunity Overview

03 Data Introduction and Analysis

04 Collaborative Filtering

05 Content Filtering

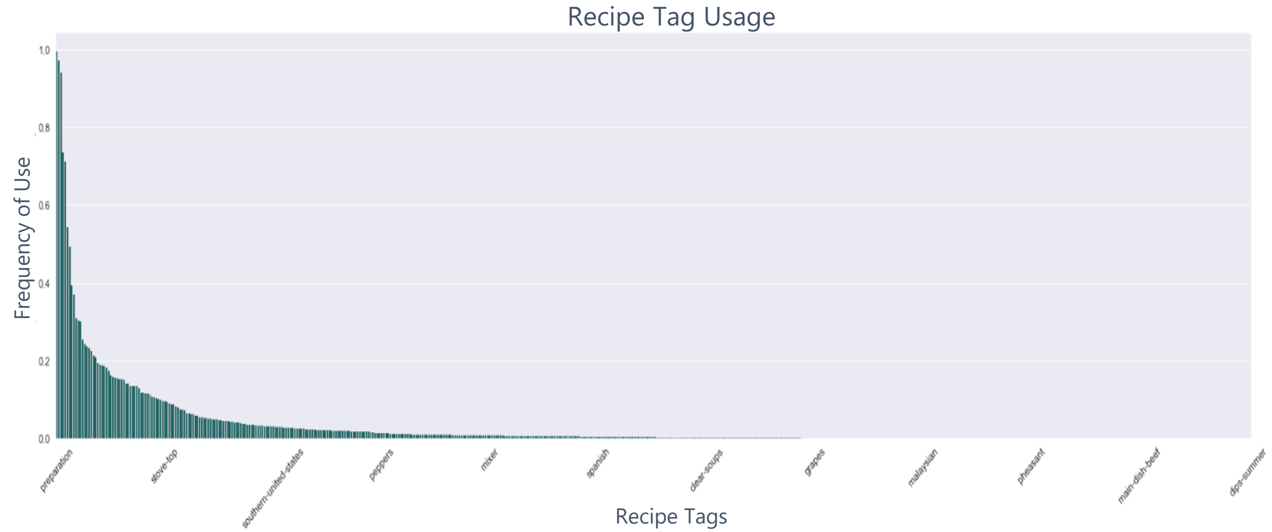
06 Feature Engineering

06 Key Takeaways

# Recipe Similarity

Scikit-Learn TF-IDF and Count Vectorization of recipe tags (i.e. keywords) were used to measure recipe similarities.

**552** tags were identified across all recipes




Recipe tags offered a unique comprehensive way to differentiate themes. Tags were not repeated within a single recipe but there were obvious tags serving as noise and an apparent long tail of highly unique tags.

TF-IDF (term frequency-inverse document frequency) offered a unique approach to down weight overly common words. Since words were not repeated within a single recipe's tags, simple binary count vectorization offered a quality baseline as comparison. Tag vectors were then used to calculate cosine similarity values for every pair of recipes, filtering to the most rated recipes.

**input**

my thai sesame noodles

**Count Vectorization**

easy asian beef noodles

- ❑ asian
- ❑ course
- ❑ 15-minutes-or-less
- ❑ cuisine
- ❑ preparation
- ❑ main-ingredient
- ❑ pasta
- ❑ pasta-rice-and-grains
- ❑ beginner-cook
- ❑ easy
- ❑ time-to-make

**TF-IDF Vectorization**

thai noodles with spicy peanut sauce

- ❑ side-dishes
- ❑ asian
- ❑ course
- ❑ cuisine
- ❑ preparation
- ❑ main-ingredient
- ❑ pasta
- ❑ thai
- ❑ pasta-rice-and-grains
- ❑ easy
- ❑ time-to-make

# Agenda

01 Executive Summary

02 Opportunity Overview

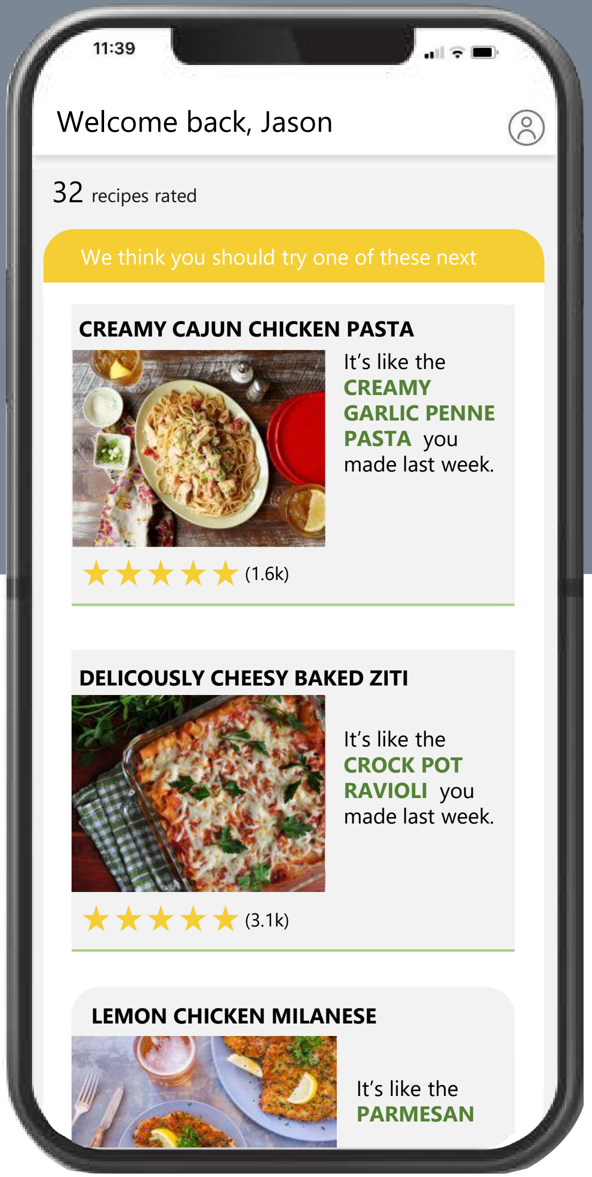
03 Data Introduction and Analysis

04 Collaborative Filtering

05 Content Filtering

06 Feature Engineering

06 Key Takeaways




# Feature Development

Results from collaborative and content filtering machine learning were functionalized to provide simplified, personalized recommendations in a seamless fashion.

- Use collaborative filtering to identify best possible prediction ratings for each user
- Leverage context filtering to relate to similar recipes from a user's history
- Intelligently measure recipe popularity and track new user rate of return
- Generate and adjust automatic grocery list creation from recommendations

## Top 10 Recipes For You

Recommend best possible recipes based on similar users

## Your Similar Recipes

Contextualize recipe recommendations based on previous experiences

## Popular Starter Recipe

Ensure new users receive the best possible recommendations

## Add to Grocery List

Integrate recommendations into interactive grocery list

# Agenda

01 Executive Summary

02 Opportunity Overview

03 Data Introduction and Analysis

04 Collaborative Filtering

05 Content Filtering

06 Feature Engineering

06 Key Takeaways

# Key Takeaways

A large network of diverse ratings, recipes, and repeat users, may be ripe for collaborative and content filtering recommendations

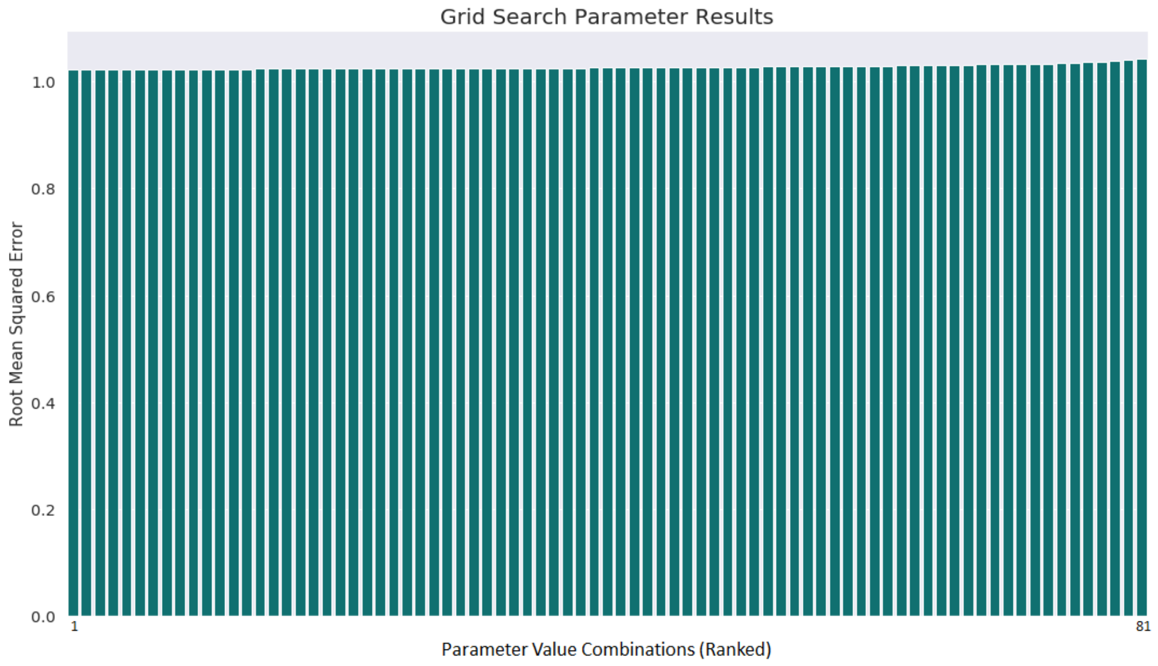
Simplified recommendations from collaborative filtering and association to a user's history through content filtering could provide a differentiating loyalty driver for large scale recipe aggregators. However, an organization should fully understand the distributive nature of their recipe ratings and user frequency before pursuing a collaborative filtering engine.

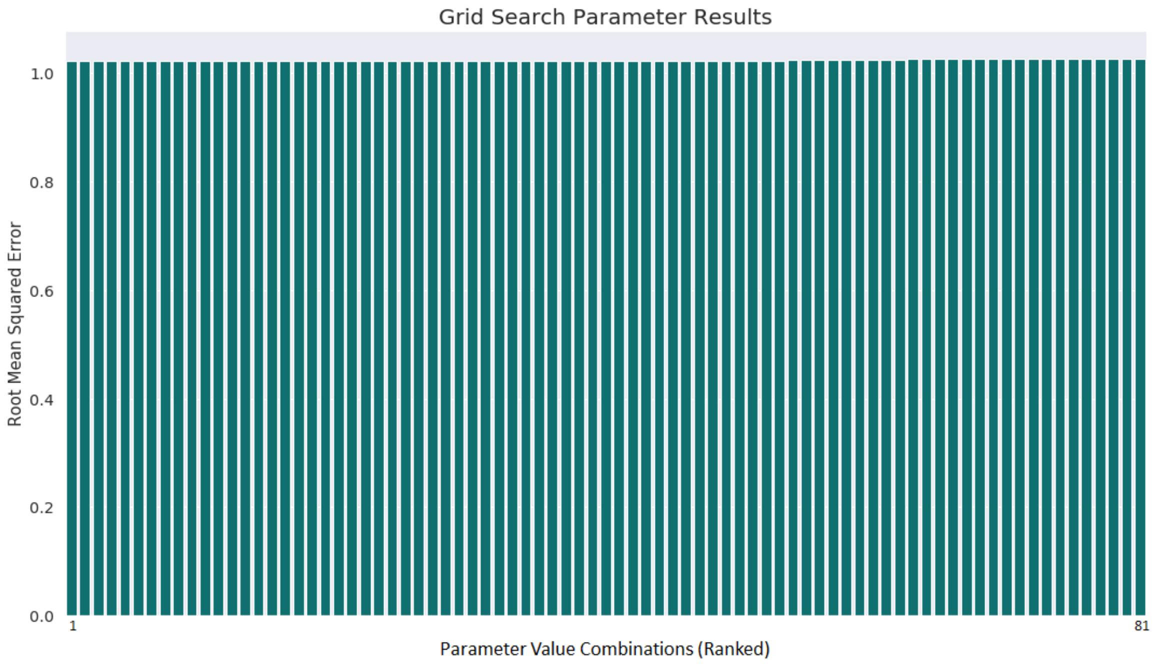
Additionally, organizations should implement a separate strategy based on popularity filters to capture and retain new customers.

The market for digital recipe aggregation is saturated; however, there is room for a highly simplified, highly intelligent recipe suggestion engine.

# Appendix

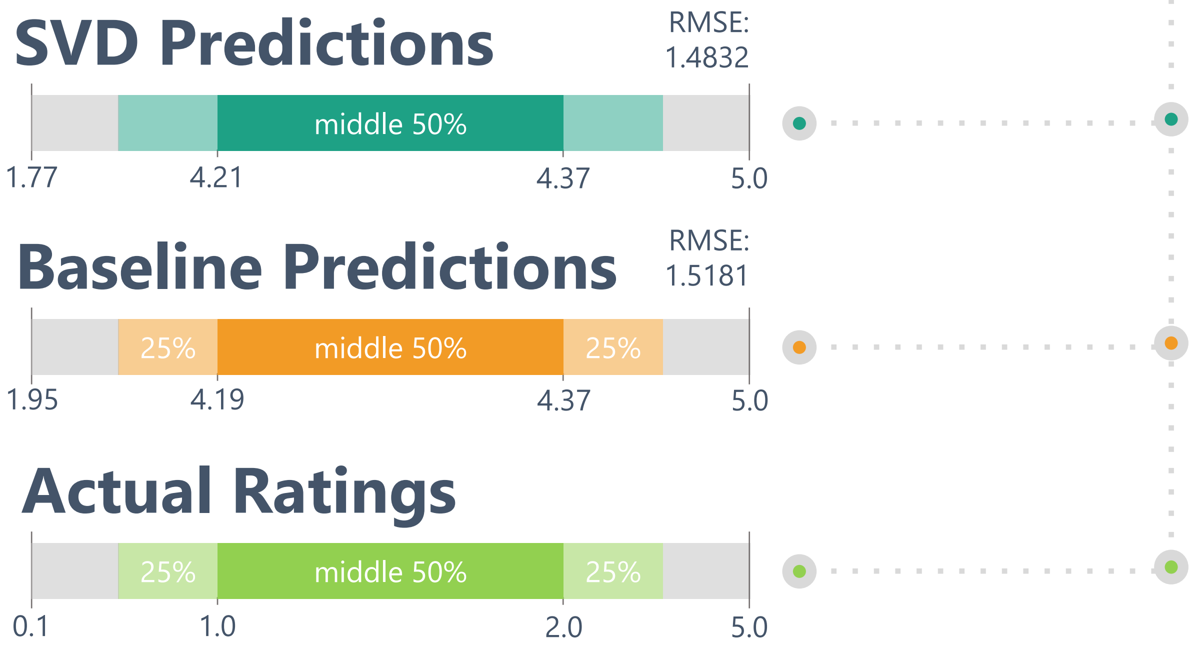
# SVD Grid Search Outputs

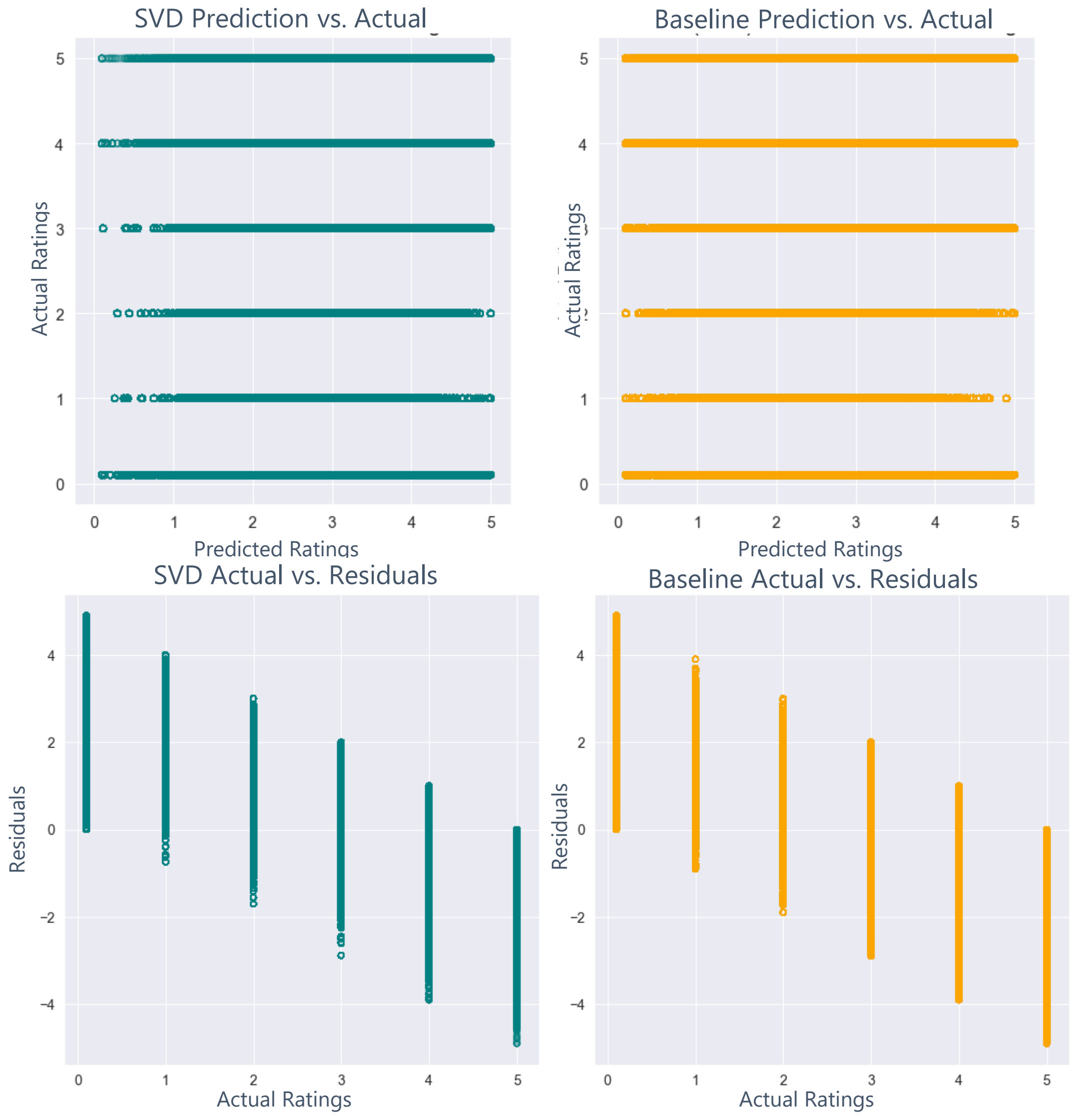


# Prediction Performance (Balanced Ratings)

The ratings data were sorted ascending by submission date to allow the model to learn emerging trends.
Each model was trained on the initial 70% of ratings and evaluated on the remaining 30%.

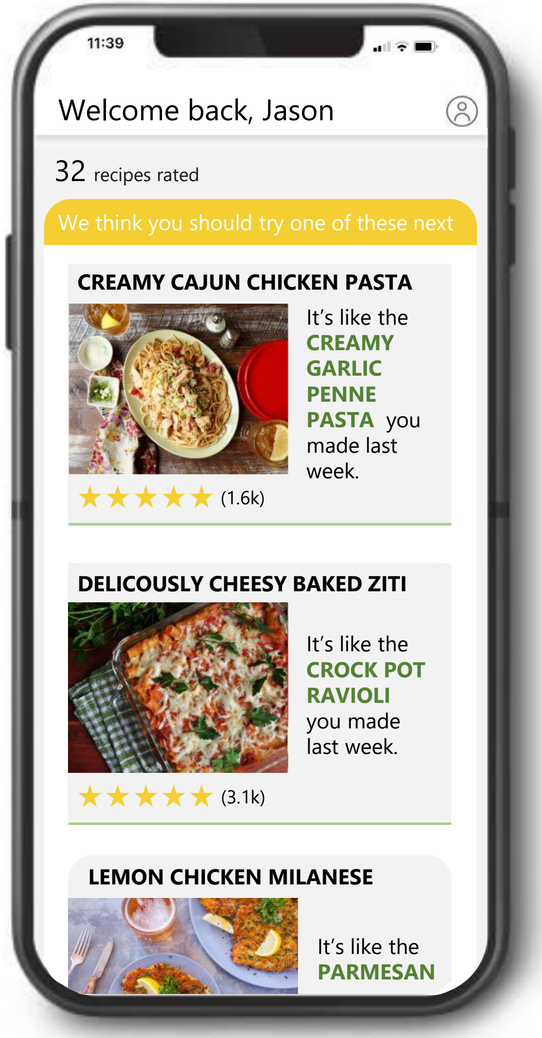



Ratings of 0 were updated to 0.1 for modeling purposes

# Feature Mockups

## Existing User




## New User