NINKASI: Beer Recommender

By Alex Li,
Andrew Wu,
Nelson Chen,
Luke Chu

Outline

- Introduction
- Data scraping
- Exploratory data analysis
- Recommendation system
  - Content-based method
  - Collaborative filtering method
- Python Flask App
- Future improvements

Introduction

Introduction

- Build a recommender system for beer lovers!
- Recommender can find similarly beers based on reviews
- Recommender can also give suggestions based on other user explicit ratings
- Models tied together in a Flask App

Project Workflow

Data Scraping

Data Preprocessing & EDA

- NLP Processing
- EDA

Recommender System Algorithm

- Content-Based
- Collaborative Filter

Flask App

Web Scraping & Data Cleaning

Data Scraping

- Website: www.ratebeer.com
- About 280,000 reviews
- Limited data scope to top 25 beers per state

Scrapy Fields

Beer Information	Beer Review Information
Beer Name	User Name
Brewer Name	User Location
Weighted Average	Time of Review
Beer Image	User Rating
State Beer Produced	Aroma
Overall Score	Appearance
Beer Style	Taste
Alcohol by Volume	Palate
Estimated Calorie	Overall
IBU (Bitter Unit)	Review
Beer Description	

Data Scraping

Home > Breweries > United States: Illinois > Goose Island Beer Company (AB-InBev)

Goose Island Bourbon County Stout

* picture credits copyright may apply

overall **100**

100 style

Brewed by Goose Island Beer Company (AB-InBev)

Style: Imperial Stout - Top 50

Chicago, Illinois USA

Serve in Snifter

bottled common | on tap common | Broad Distribution

send corrections | shelftag | edit barcodes | update pic

RATINGS: 2809 WEIGHTED AVG: 4.26/5 IBU: 60 EST. CALORIES: 426 ABV: 14.2%

COMMERCIAL DESCRIPTION

"I really wanted to do something special for our 1000th batch at the original brewpub. Goose Island could have thrown a party. But we did something better. We brewed a beer. A really big batch of stout- so big the malt was coming out of the top of the mash tun. After fermentation we brought in some bourbon barrels to age the stout. One hundred and fifty days later, Bourbon County Stout was born-A liquid as dark and dense as a black hole with a thick foam the color of bourbon barrels. The nose is a mix of charred oak, vanilla,carmel and smoke. One sip has more flavor than your average case of beer. It overpowers anything in the room. People have even said that it's a great cigar beer, but I haven't yet tried a cigar that would stand up to it." Brewmaster Greg Hall;
IBU's 60-High Color - Midnight
Was 11% abv,
2007 and 2008 - 13% abv
2011 - 14.5% abv
2012 - 15% abv
2013 - 14.9% abv
2014 - 14.4% abv
2015 - 14.2% abv

Editor's Note: Baudoinia Fulton & Wood Series offering (And the Low Storage entry from FoBAB 2012) is simply Bourbon County Stout aged in the same barrels. It has been alted as it offers no different recipe or barrel type simply the presence of a distillers fungus on the barrels that adds no distinct characteristic to the beer from the fungus itself except for possible oxygen exposure changes. While Brewers Intent indicates that they consider it a new beer, no true distinction aside from this oxidation amount and possible aging time differences separates the beers. It's essentially a single barrel Bourbon County version something we have always treated as regular Bourbon County Stout in the past.

Most Recent | Top Raters | Highest Score | Rated By | Ticked By

4 AROMA 9/10 APPEARANCE 3/5 TASTE 8/10 PALATE 4/5 OVERALL 16/20

Sudz4Dayz (70) - Montreal, Quebec, CANADA - DEC 16, 2016

2016 bottle purchased at Fort Point Market in South Boston, MA. Pours jet black, smooth, cappuccino head with small frothy bubbles. Dissipates very quickly. Smells like burnt wood, vanilla, roastiness with bourbon sweetness and booze as well. Love it. Nice roasty flavor, smooth vanilla. Those come first and are quickly replaced by ample, everlasting dark chocolate flavor. Sweet bourbon barrel throughout and slight charred wood. Not the thickest mouthfeel but I wouldn't say light either. Medium. Overall not all too complex, balance is ok. The flavor is great and it definitely saves this one from being mediocre. In terms of overall quality it's definitely missing balance, a little complexity and a little on the mouthfeel. Delicious nonetheless. I enjoyed this one.

4.1 AROMA 8/10 APPEARANCE 3/5 TASTE 9/10 PALATE 4/5 OVERALL 17/20

chinchill (4448) - South Carolina, USA - DEC 15, 2016

12 oz bottle dated 29 August 2014 served in a Belgian snifter. Removing the cap indicated a secure seal, but this has very little carbonation. This is not ideal for the appearance, but the low carbonation seems about right for the smooth, full, high ABV body. Boozy and woody aroma with dark roasted grains and molasses. Sweet bourbon dominates the rich flavor. Moderately woody with faint vanilla and coffee in the finish. Mild bourbon barrels and dark bread in the aftertaste.

4.7 AROMA 9/10 APPEARANCE 5/5 TASTE 9/10 PALATE 5/5 OVERALL 19/20

Korcz (1142) - Warsaw, POLAND - DEC 11, 2016

Backlog, ocena przepisana z untappd, w ramach uzupełniania profilu na ratebeer. Genialne. Perfekcyjnie gładkie, oleiste, kremowe, eleganckie. Piękna wanilia, szlachetny alkohol, potężne nuty Bourbonu, odbeczkowy kokos i genialna jak na tę moc pijalność. Wybitne piwo, jeden z moich ulubionych stoutów ever. Jeśli nie ulubiony :)

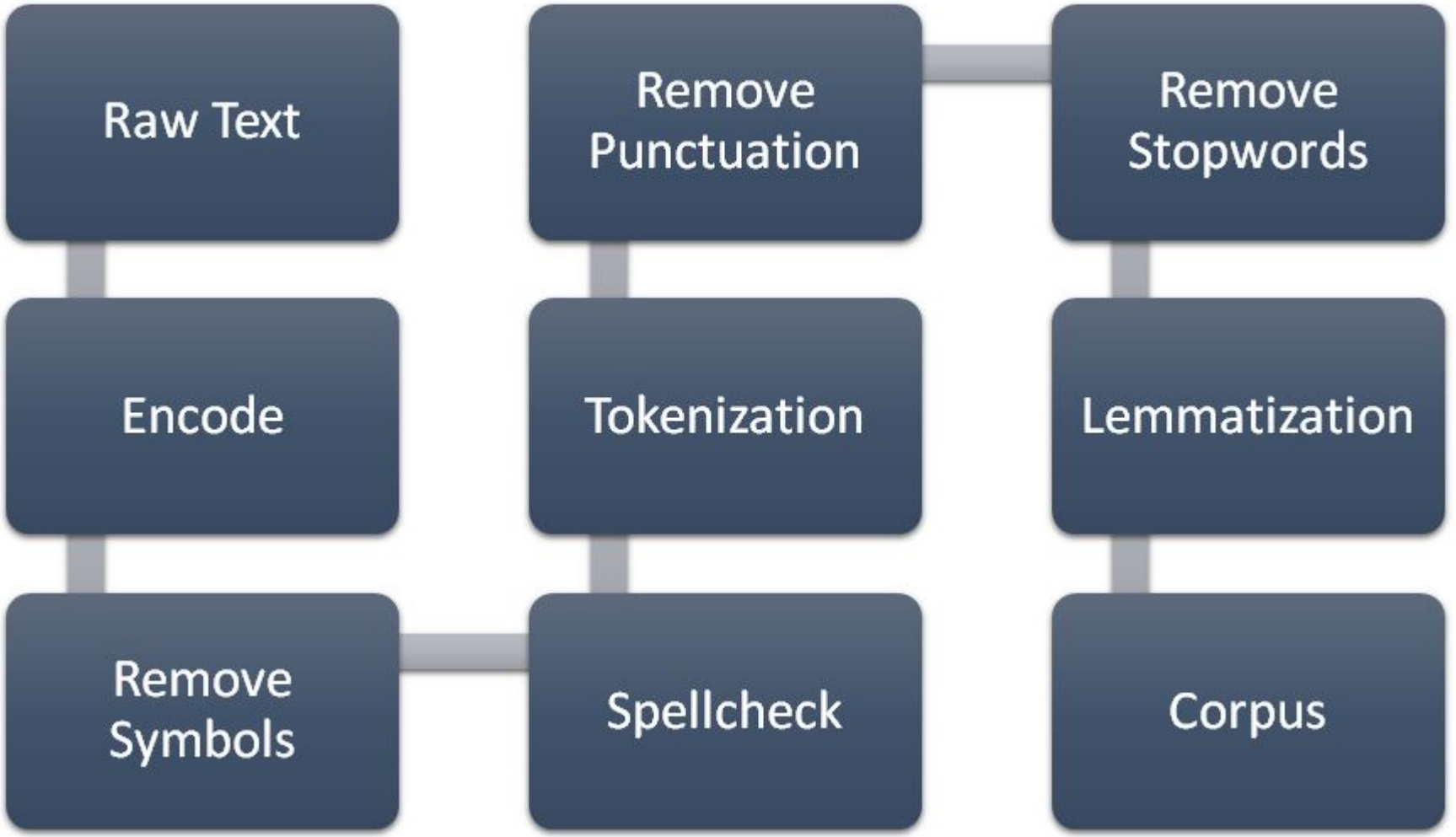
4.2 AROMA 9/10 APPEARANCE 5/5 TASTE 8/10 PALATE 5/5 OVERALL 15/20

Listigovers (967) - Toronto, Ontario, CANADA - DEC 10, 2016

Pitch black, very strong whisky scent. Taste is very sweet and there is also heat from the 14%, tar like texture, very roasted and bourbony flavour. A true sipper, incredibly heavy and warming beer.

Data Preprocessing

- ~1200 beers
- ~16000 users

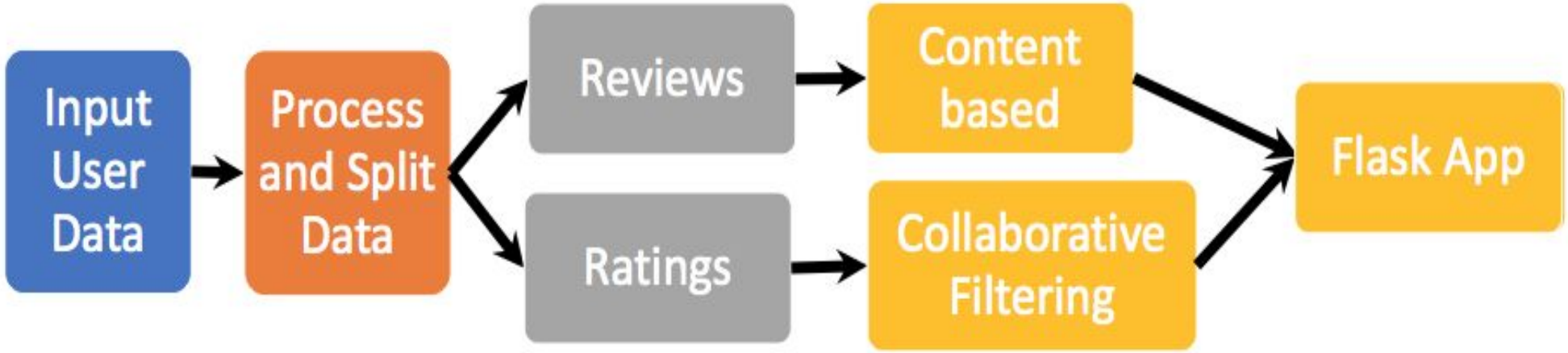


Exploratory Data Analysis

EDA: Beer Items

Recommender System

Recommender System



Content-Based Algorithm

- Recommendation based on user review.
- Two algorithms were implemented:
  - Term Frequency-Inverse Document Frequency (TF-IDF) to produce the document-term matrix
  - Latent Semantic Analysis (LSA) does dimension reduction on the document-term matrix

Term Frequency-Inverse Document Frequency (TF-IDF)

- Calculate the “importance” of every word to a review in a corpus.
- Produce a document-term matrix
- Term Frequency is the number of times a word occurs in a document.

$$\mathrm{tf}(t,d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}}$$

- Inverse Document Frequency measures how much information each word provides, or how rare is the word across all documents.

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

- The TF-IDF is defined as

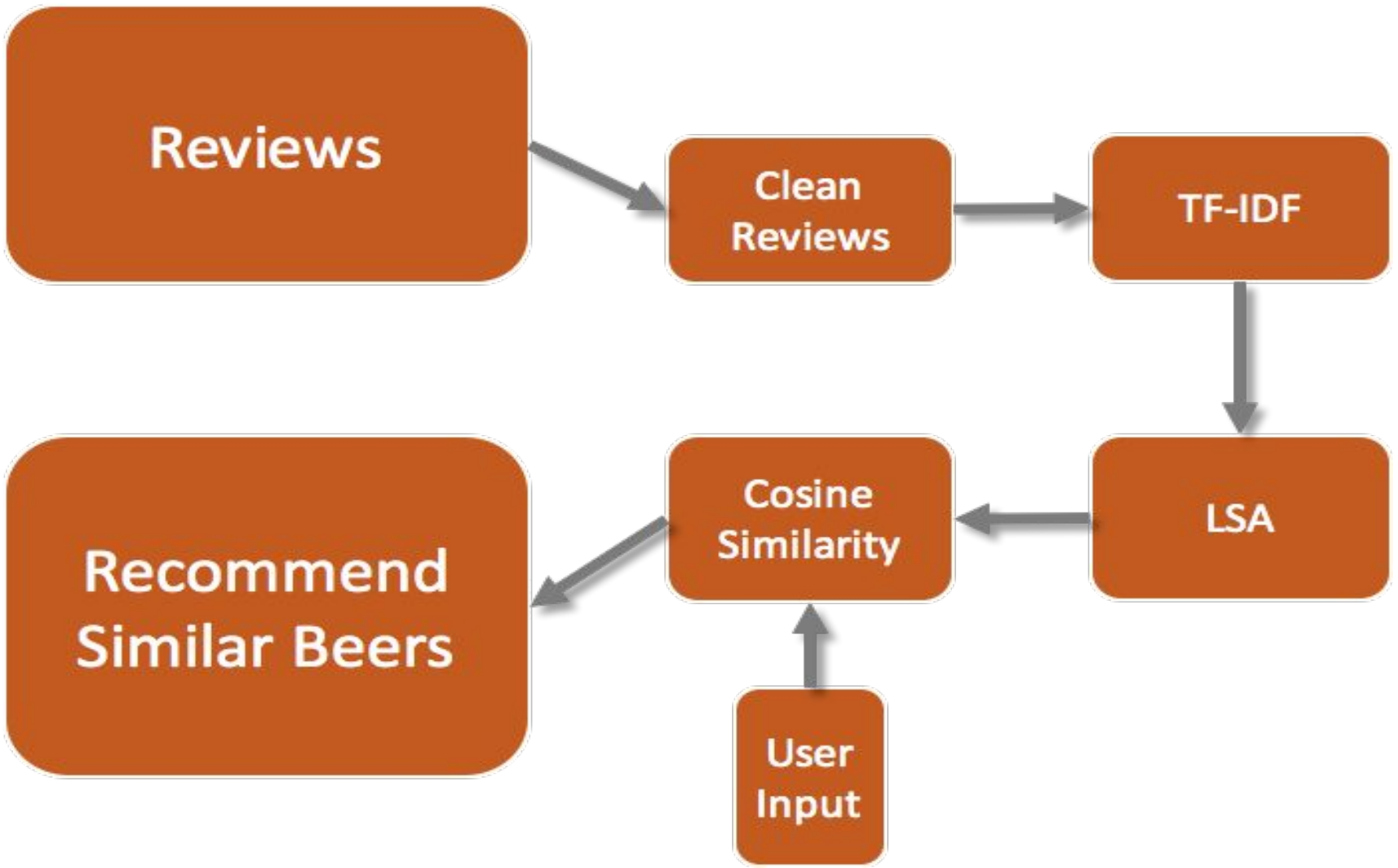
$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D)$$

Latent Semantic Analysis (LSA)

- Similar to PCA, LSA does dimension reduction by performing SVD on the document-term matrix

$$
\begin{array}{cccccc}
 & X & & U & \Sigma & V^T \\
 & (\mathbf{d}_j) & & & & (\hat{\mathbf{d}}_j) \\
 & \downarrow & & & & \downarrow \\
(\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix} & = & (\hat{\mathbf{t}}_i^T) \rightarrow & \begin{bmatrix} \begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} & \cdots & \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_l \end{bmatrix} \cdot \begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix}
\end{array}
$$

Content-Based Recommender System

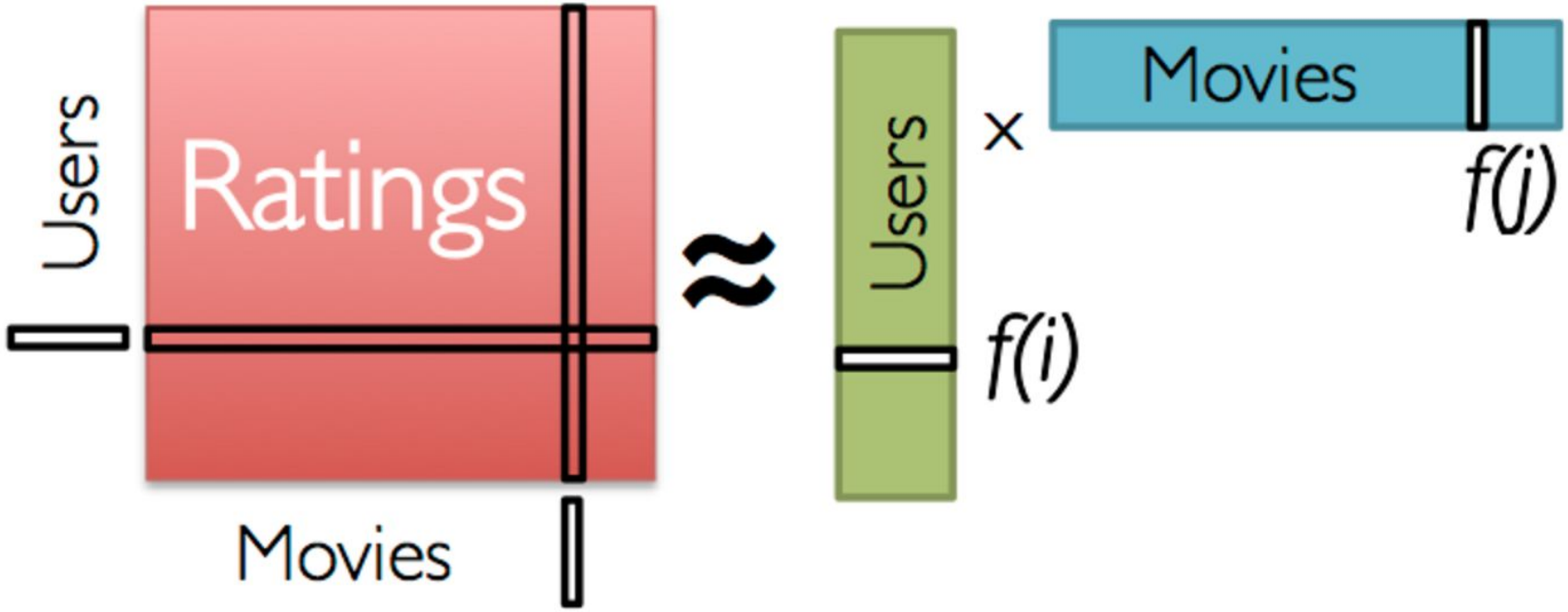


Collaborative Filtering

- Recommendation based on beer rating (explicit information)
- Two separate models were used
  - Singular Value Decomposition++ (SVD++)
  - Restricted Boltzmann Machine (RBM)

SVD For Collaborative Filtering

- Latent Matrix Factorization of user-item matrix to latent features
- Used Spark to implement as baseline: RMSE = 2.15



SVD++ (Implicit Feedback Version)

- Developed by Netflix Challenge winners
- Further decompose ratings to global average, user/item biases, implicit feedback, and latent features
- Remove bias of each user and item to center data
- Users that have less ratings are penalized more (given rating closer to average)

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} y_j\right)$$

Predicted rating

Global average

User bias

Item bias

Item Latent Feature

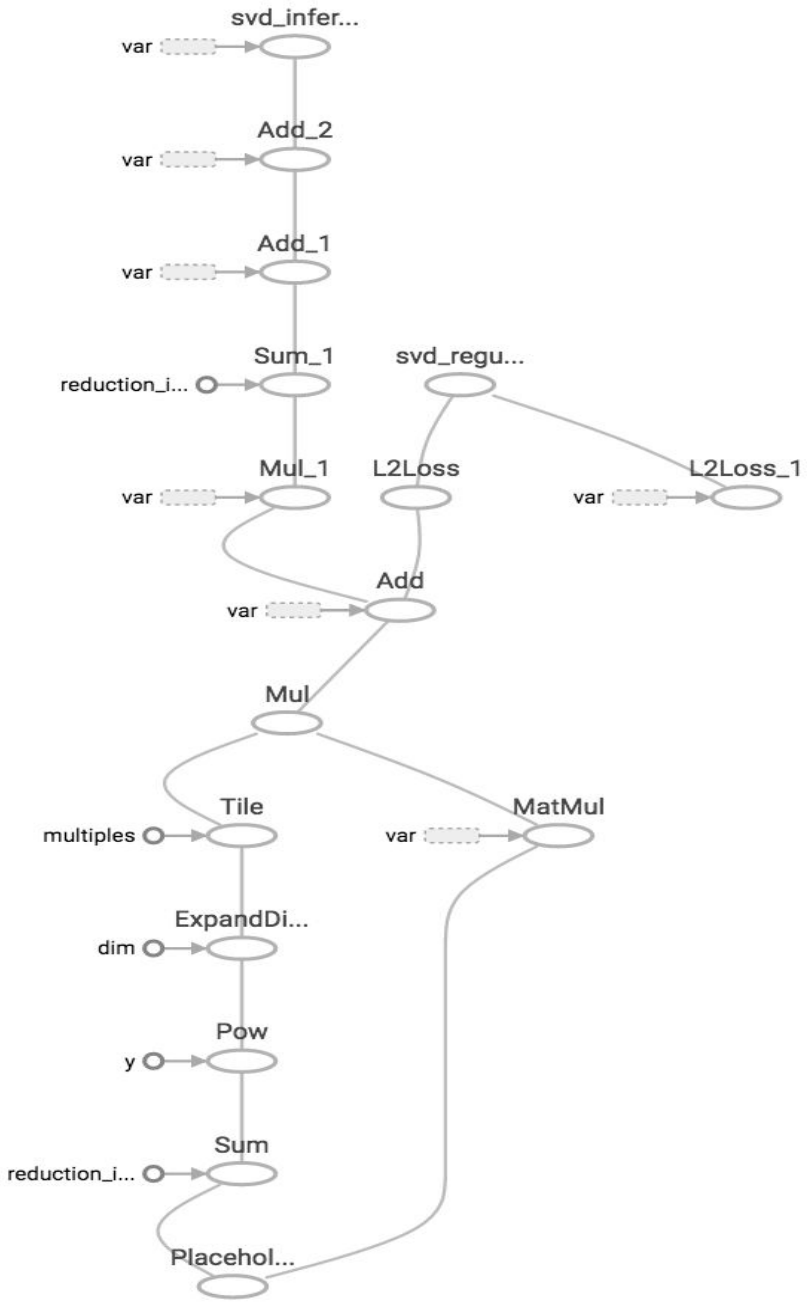
User latent feature

Implicit feedback

Implicit feedback Parameters

SVD++ Implementation

- No fast publicly available package
- Realized SVD++ can be highly optimized using ***TensorFlow***
- Found vanilla SVD algorithm built in Tensorflow
- Augmented code to include biases, implicit feedback, k-fold cross validation, and early stopping
- Accelerated by C++ backend, GPU computation
- **Results: RMSE = 1.69**

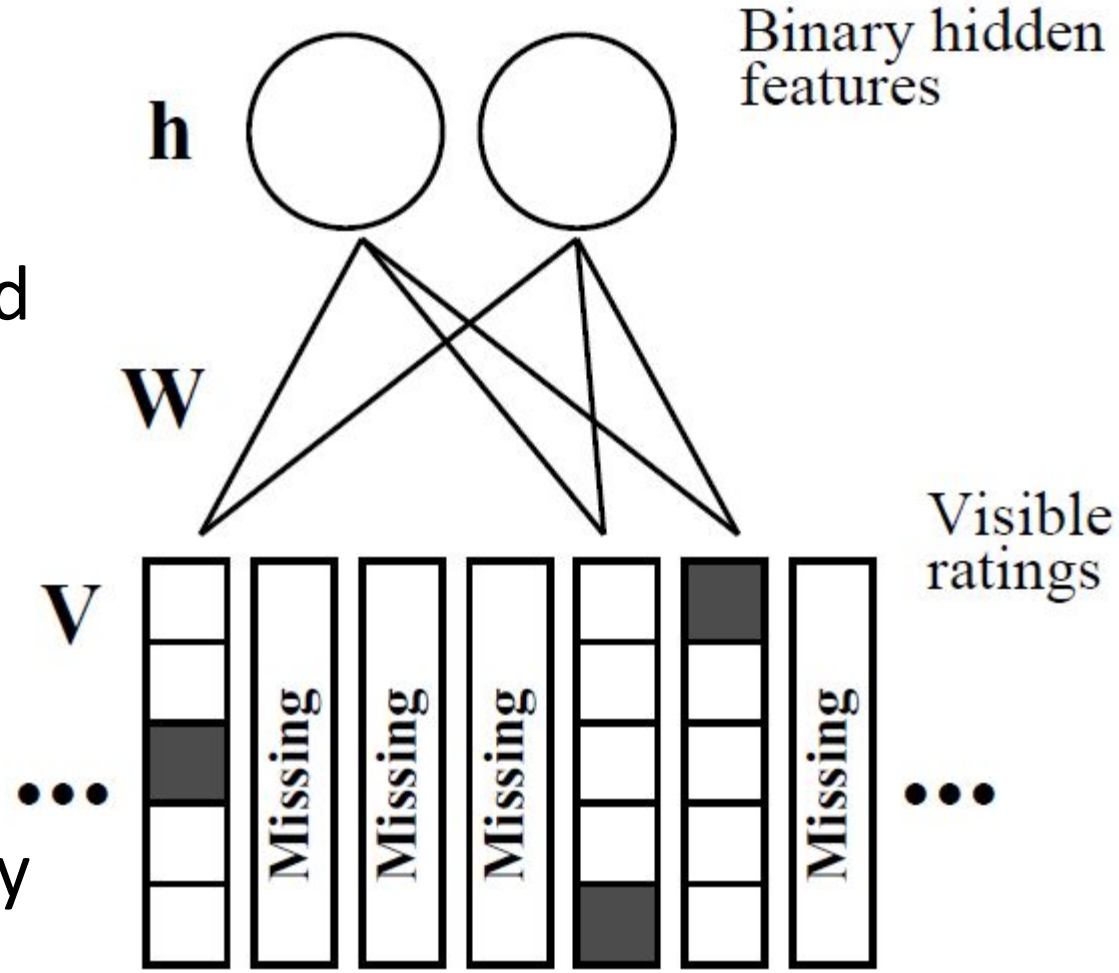


Restricted Boltzmann Machine (RBM)

- RBM is an unsupervised, two-layer neural network
- Inspired by the Boltzmann Distribution from thermodynamics and fluid dynamics to minimize the energy function
- Performs CF by reconstructing the user-item matrix
- Hidden units can be thought of as binary latent factors

Restricted Boltzmann Machine

- To train a RBM:
  - Initialize the visible layer
  - Hidden layer is calculated
  - Missing values are imputed
  - Weights are updated based with gradient descent
- The RBM is trained for every user, but the weights and bias are shared across all users



Ensemble CF models

- Averaged SVD++ and RBM predictions to get final predictions

Prediction Method for New Users

- Neighborhood approach
- Compute cosine similarity between new user and all users
- Impute missing ratings of new user by a weighted average proportional to similarity metric
- Rank recommendations and output top 10

Flask App Demonstration

Lessons Learned & Future Steps

Lessons Learned

- Text is fundamentally messy to work with
- When an algorithm is not available, implement it yourself
- Hacking other people's code can be time consuming (RBM), document your code for readability
- Don't try to build Flask app from scratch in two days

Future Steps

- Add in more features (i.e style, palate, appearance, aroma, taste, and time)
- Tune hyperparameters to get optimal single models
- Ensemble smarter (use minimizer or stacking)
- Develop App aesthetics and functionalities further