Yelp Restaurants Recommendation Engine

---MSCA 31008 Data Mining Principles(Winter 2022)

Ryan Liao
Jason Lee
Norah Zhang
Milan Toolsidas
Minglun Pan

Agenda

Executive Summary → Data ETL → EDA/Feature Engineering → Model Mining → Model Evaluation

Executive Summary

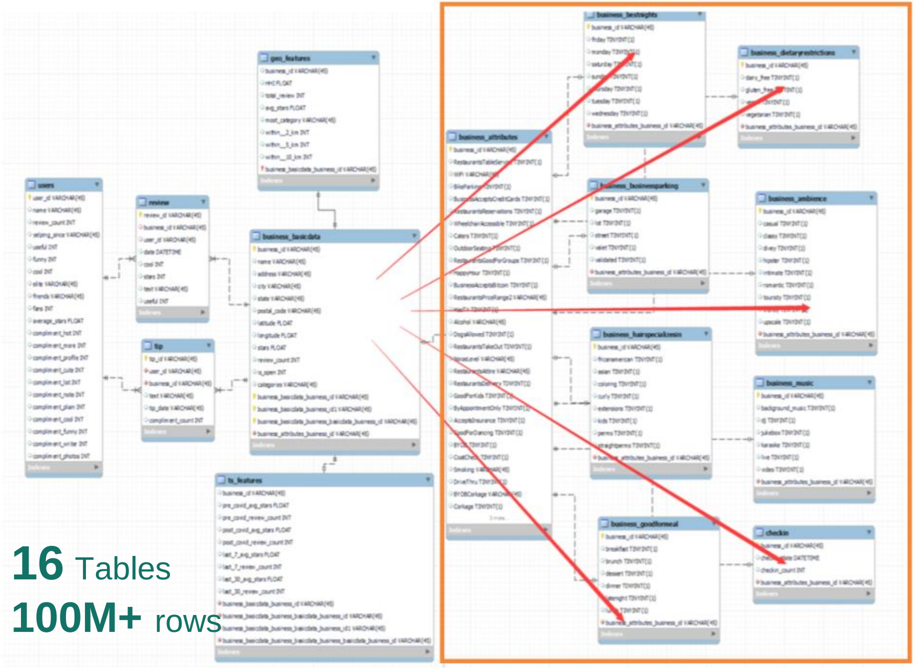
- To help users more quickly identify restaurants they would enjoy, our team created a recommendation engine to recommend restaurants based on Yelp's open dataset.
- In this project, we mainly focused on restaurants:
    - Extracted and transformed semi-structured JSON data into structured database
    - Loaded the data onto Research Computing Center, managed and connected to Python through Hive and Spark
    - Leveraged NLP, time series, and geographical analysis to engineer valuable features to add to our dataset
    - Reduced dimensionality through SVD, and generated restaurant recommendations using collaborative filtering, linear models, tree-based models and their ensembles prompted by Facebook.
- The ensemble model promoted by Facebook outperforms others, but the traditional Collaborative Filtering has its own advantage in interpretability.

Database Overview

Data Extraction: Source JSON files from Yelp

	Features	Rows	Complexity
Business	14	64K	**more than 30 nested variables**
Users	22	2.2M	**3.7 GB**
Review	9	8.9M	**7 GB**
Tip	5	938K	
Check in	2	62K	

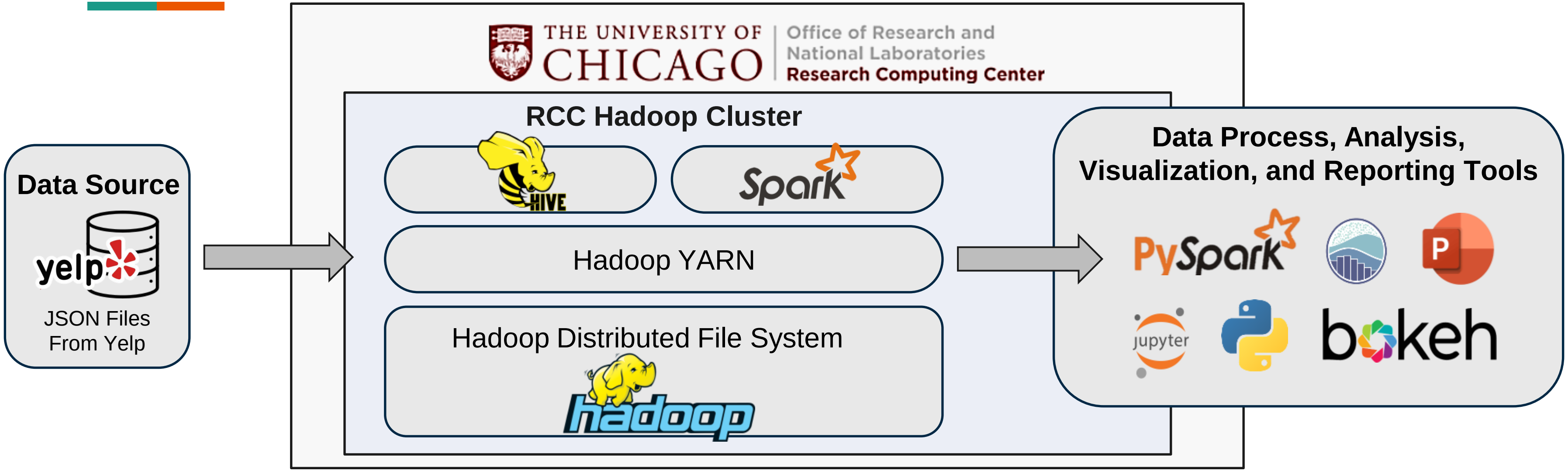
Business Attributes



16 Tables

100M+ rows

Database Architecture



- **Database Selection:** Considering ease of access and cost, we chose RCC over GCP
- **Data Transformation and Loading:**
  - Steps: Clean, normalize, feature select and upload
  - We created twelve tables focused on restaurants and uploaded in a database in Hadoop

EDA - NLP

To gain insight into how users could use different words to describe restaurants, here is a simple word cloud Venn diagram showcasing words used in the reviews.

- Blue words are only used in five-star reviews.
- Red words are only used in one-star reviews.
- Gray words in the middle are used in all reviews.

Good Review Words
Bad Review Words
Common Words

Feature Engineering - NLP

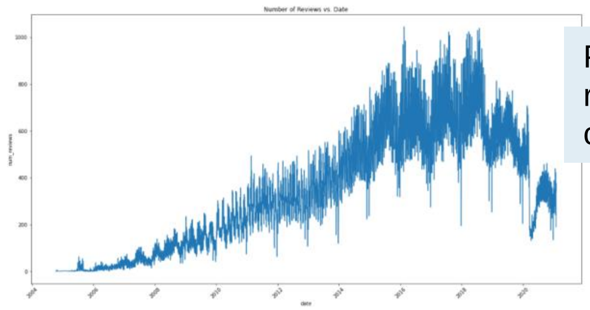
Goal: Extract insights from users' reviews to better understand user preferences

Methods:

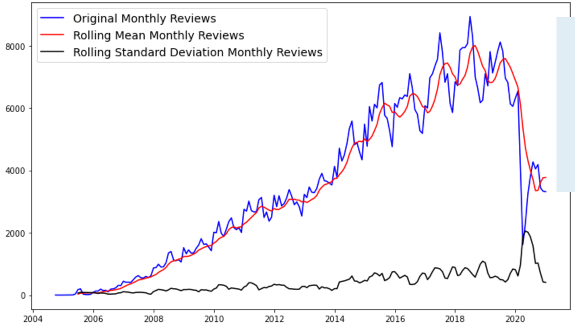
TF- IDF	BERT
• TF-IDF is a product of **Term Frequency** and **Inverse Document Frequency** • TD-IDF evaluates how relevant a word is to a document in a collection of documents	• BERT is **pre-trained bidirectionally trained language model** that extract high quality language features from text • BERT could be further **fine-tuned** for specific applications

- Generated over 20 GB of features from TF-IDF and BERT
- Quasi Map-Reduce process
- Combined the sparse matrix output into final model

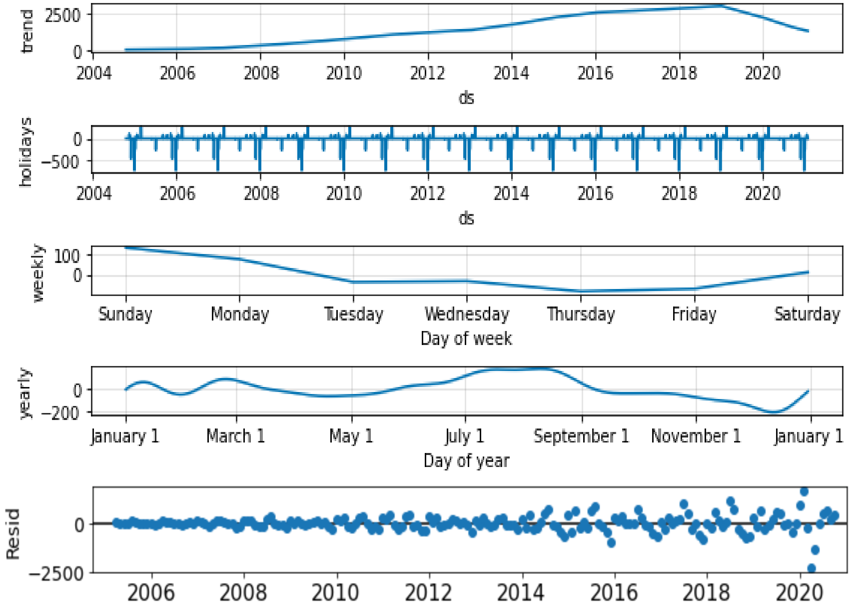
EDA - Time Series



Preliminary view of review counts show cyclic pattern



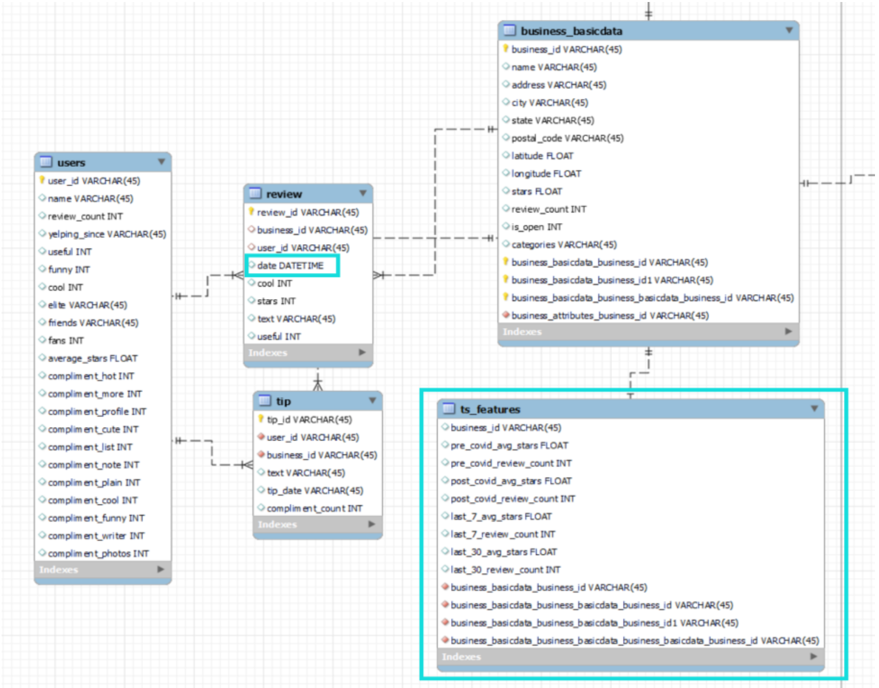
Sample of the largest category shows cyclic pattern in rolling means



The feature shows seasonality after decomposition:

- Notice disturbance due to COVID
- Number of reviews peak in summers and on Sundays

Feature Engineering - Time Series



Goal:

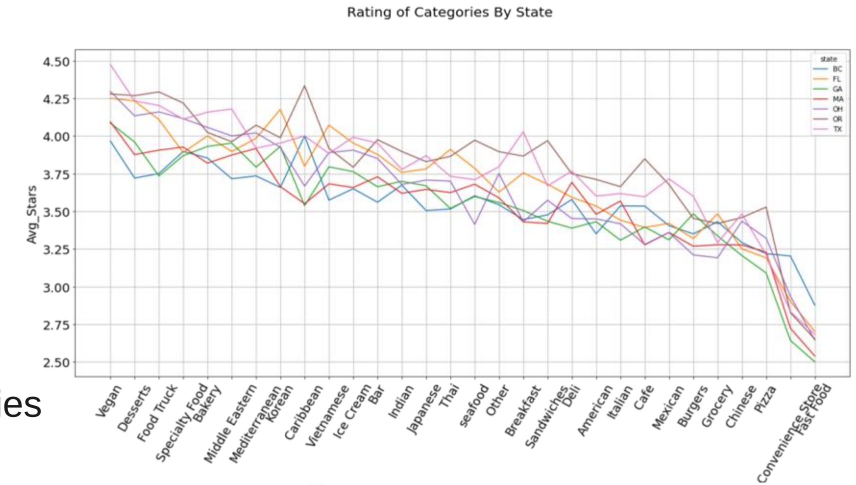
Extract seasonality and other features from time and turn them into single dimention variables

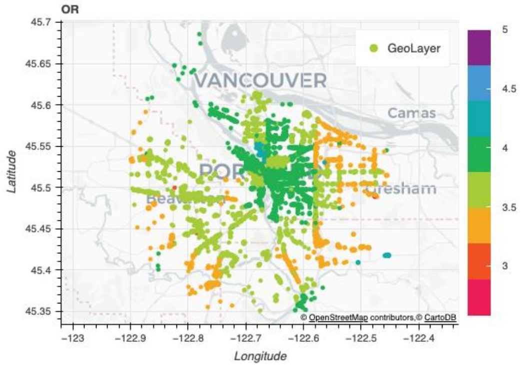
Method:

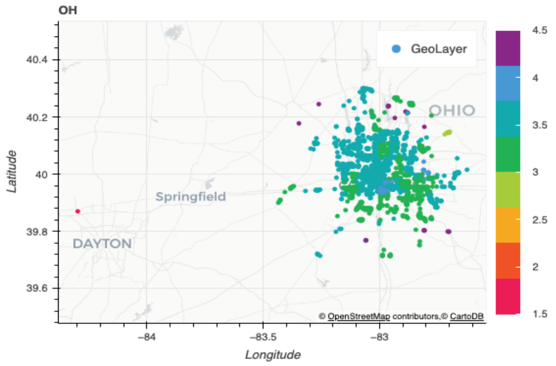
- Calculated features for average stars and number of reviews before and after COVID
- Created features for average stars and number of reviews in last 7 days and last 30 days

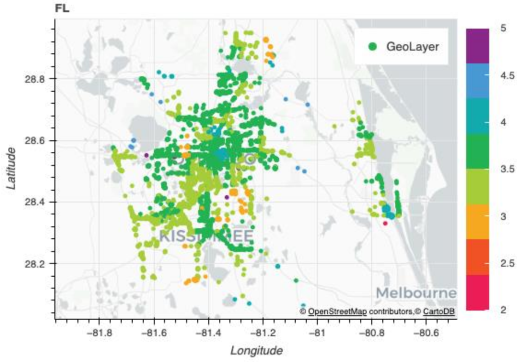
EDA – Geographic info

At state level:
- Ratings seem to have patterns among the states
- Higher ratings tend to be in in more centralized areas within the state
- Some states display preference of certain categories

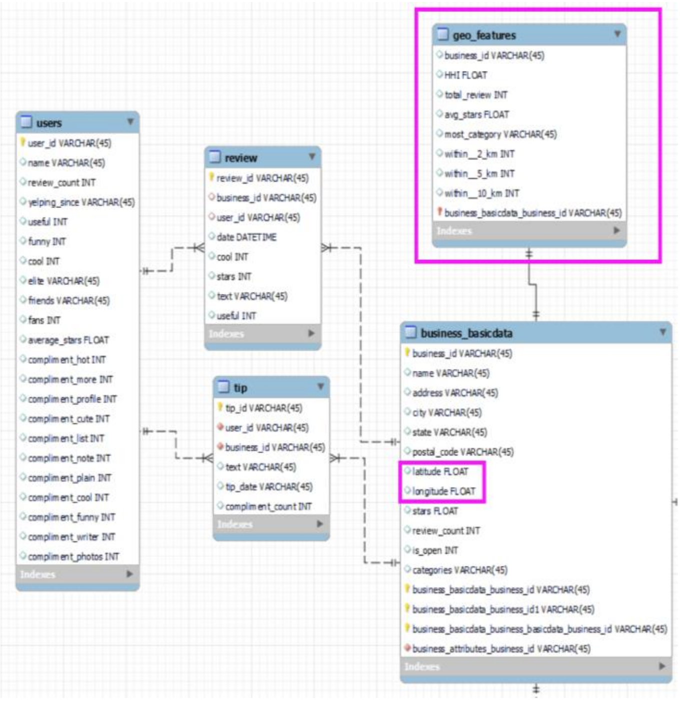








Feature Engineering - Geographic Info



Goal:

Convert (Latitude,Longitude) -> Multiple 1D data about the business

Method:

Calculate distance between each Business, then calculate aggregative informations

business_basicdata.business_id	longitude	latitude
6iYb2HFDywm3zjuRg0shjw	-105.0	40.0
tCbdrRPZA0oilYSmHG3J0w	-123.0	46.0
D4JtQNTl4X3KcbzacDJsMw	-123.0	49.0
jFYIsSb7r1QeESVUnXPHBw	-83.0	40.0
rYs_1pNB_RMtn5WQh55QDA	-105.0	
...	...	
Q78fYV6B6P6GmX07YVgi4g	-98.0	30.0
uXdQkuEtvLAzfc3MsO-sTQ	-84.0	34.0
bQX-kwVTyZgcdZGEPzce6Q	-81.0	28.0
GB75wPibj3ljNauaoCxyGA	-123.0	46.0
ngmLL5Y5OT-bYHKU0kKrYA	-81.0	29.0

63898 rows × 2 columns

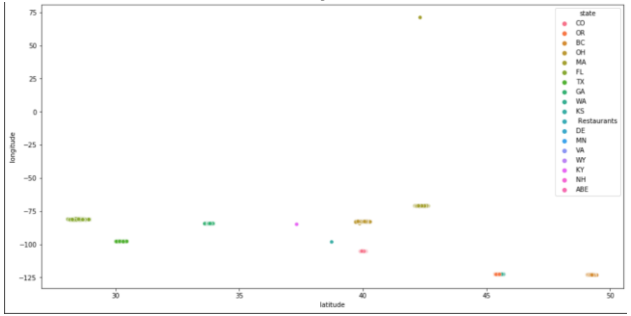
	ID1	ID2	...	ID63898
ID1	D(1,1)	D(1,2)	...	...
ID2	D(2,1)	D(2,2)	...	...
...	...	...	...	...
ID63898	...	...	...	...

	HHI	total_review	avg_stars
HPA_qyMEddpAEtFof02ixg	968.858131	4396	3.345588
hcRxdDg7DYryCxCol8ySQA	772.898129	117373	3.466014
jGennaZUr2MsJyRhijNBfA	1679.552336	31405	3.338235
iPD8BBvea6YIdQZPHzVrSQ	959.105851	246738	3.492674
Z2JC3Yrz82kyS86zEVJG5A	837.500000	2491	3.250000
...	...	...	...
qMjKO9FmxJgRLwaPLIHA	732.093391	7014	3.372180
VkNXTpW6AQ_N9rTihoCBVA	650.794894	10649	3.323009
RMcxvvVJI_R9ICL95doK2w	1221.500000	19192	3.897500
y4vrZcU0EIhX2BENvRtk_w	734.441925	11402	3.392857
1CHPSnEh4axD10HCqe8k-A	775.047259	5251	3.420290

63888 rows × 7 columns

Feature Engineering - Geographic Info

Geo data clustered by states



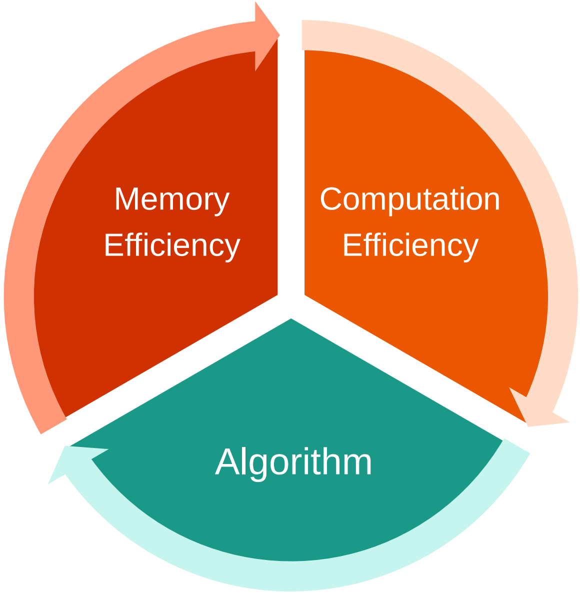
Geo data clustered by city

Challenge:
- Program time complexity is $O(N^2)$
- Memory issues (kernel dies)
- Running forever...

Solution:
- Calculate distance separatly within each states to reduced data-size by 85.6%
- Save and load intermediate results on hard drive to avoid memory issues
- Introduce Numpy vectorization to speed up calculation by 93.5%

Results:
- Reduced total run time by 99.1%
- Generated following aggregated features:
  - HHI (To measure category diversity)
  - Total review, average ratings within 2 km^2
  - Most frequent category within 2 km^2

Recommendation System Model - SVD



BACKGROUND

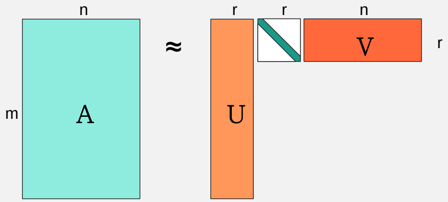
- Large Matrix
- Singular Matrix
- Missing Values

ASSUMPTION

- No Information is a kind of information
- Maximize the variance of projection

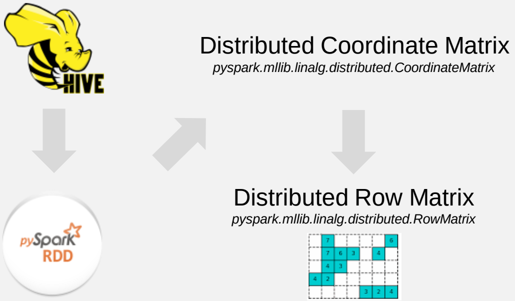
SOLUTION

ALGORITHM



Matrix Decomposition - SVD

DATA STRUCTURES



Recommendation System – Collaborative Filtering

ITEM-BASED COLLABORATIVE FILTERING

- Focuses on **relationship between pairs of items**
- Matches each users' rated items to similar items

Decomposed Business (Item) Matrix

feature1	feature2	feature...
...	...	...
...	...	...
...	...	...

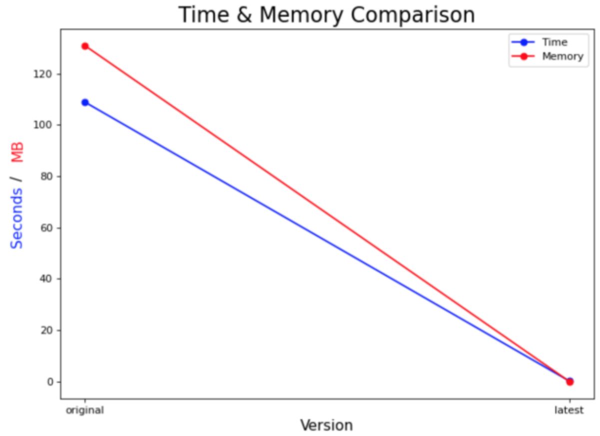
$$Similarity(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{||\vec{A}|| * ||\vec{B}||}$$

Recommendation System – Collaborative Filtering

Use items already rated by user that are **most similar** to missing item we want to generate rating for

$$rating(U, I_i) = \frac{\sum_j rating(U, I_j) * s_{ij}}{\sum_j s_{ij}}$$

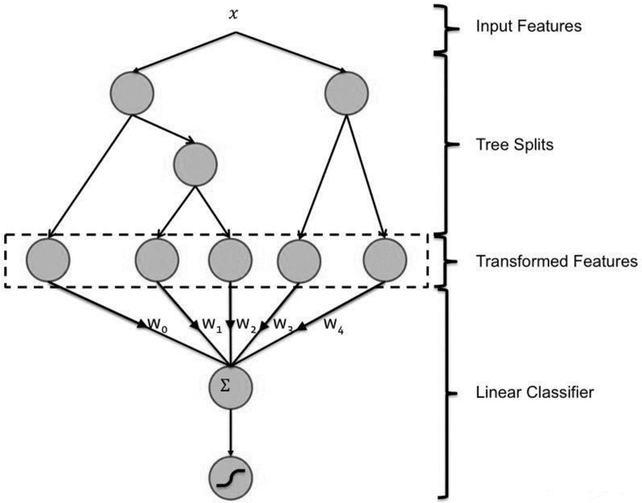
Generate a recommendation based on a **weighted sum of the ratings of other similar products**



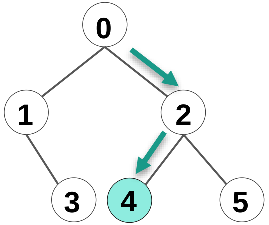
Executive Summary | Data ETL | Feature Engineering | Model | Model Evaluation

Recommendation System – XGB+LR

Structure



Step 1 **Embedding**



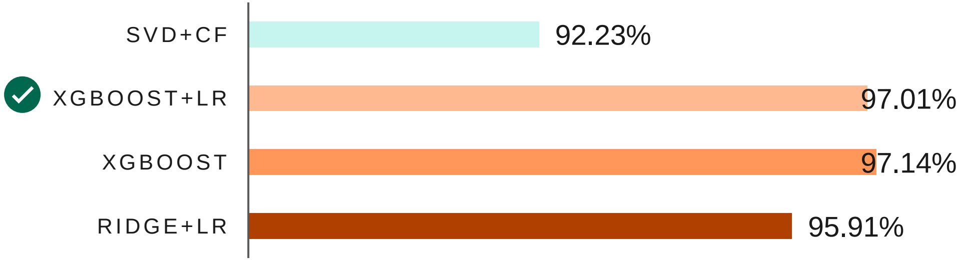
$$\#Features = \#Tree * 2^{maxDepth}$$

Step 2 **OneHot + LR**

Model Evaluation

RMSE

NDCG

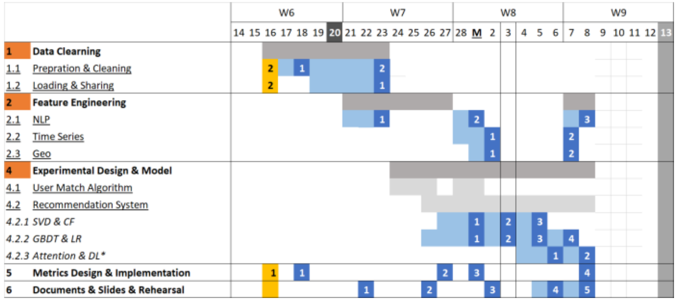


NDCG

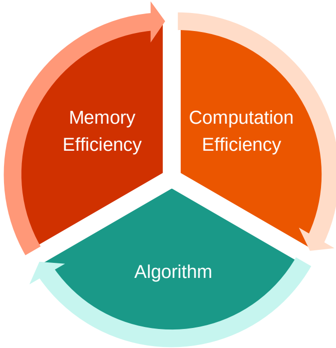
$$DCG_k = \Sigma_{i=1}^{k} \frac{rel(i)}{\log_2(i+1)}$$

i	rel(i)	log(i+1)	rel(i)/log(i+1)
1 = A	0.5	1	0.5
2 = B	0.9	1.59	0.57
3 = C	0.3	2	0.15
4 = D	0.6	2.32	0.26
5 = E	0.1	2.59	0.04

Key Takeaways



Agile Development



Three Principles

THE UNIVERSITY OF CHICAGO
Office of Research and National Laboratories
Research Computing Center

Tech Stacks

Future Work

- System
  - Improve the **stability** of daily recommendation system considering the memory and computation
  - Deal with the frequent down-time of Spark Service
  - Deep dive into the relationships between features, such as Network Analysis
- Model
  - Fine-tune the NLP Embedding models and XGBoost models
  - Implement **Gradient Descending Method** to find the proper matrix decomposition comparing to SVD

Thank you!
Questions?


yelp

Appendix

Resources

RCC Hadoop Hive Database Name: dmp_yelp_rs

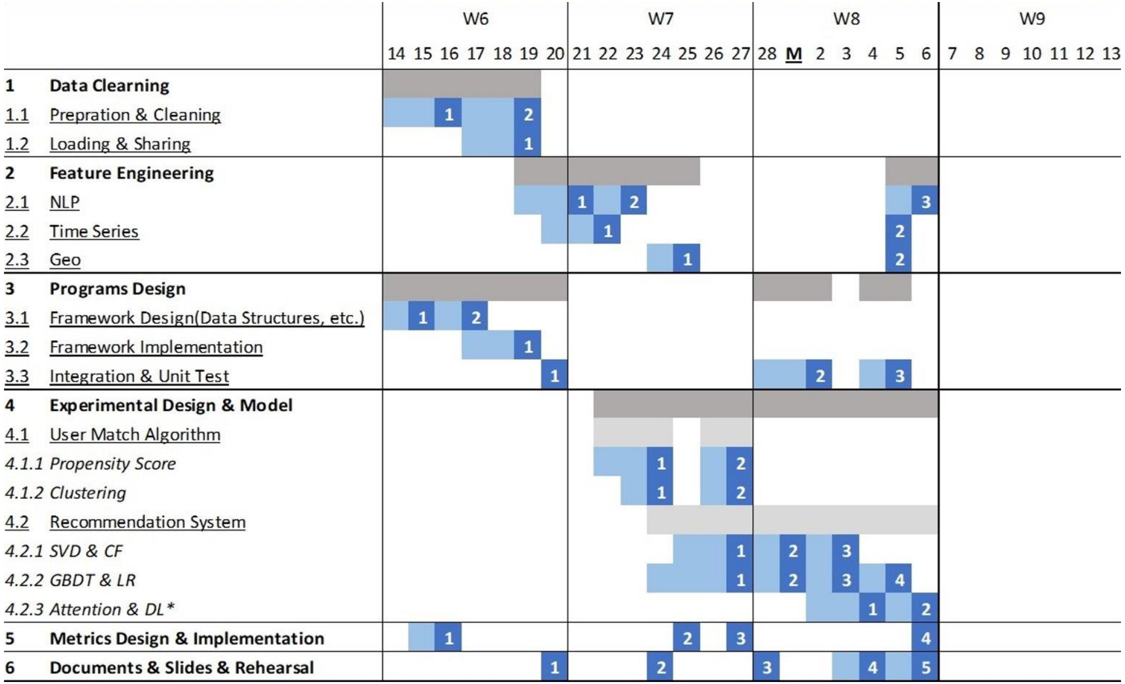
Link to GitHub Repository: https://github.com/MinglunPan/MSCA31008-Data-Mining-Principles

Link to Original Yelp Data: https://www.kaggle.com/yelp-dataset/yelp-dataset

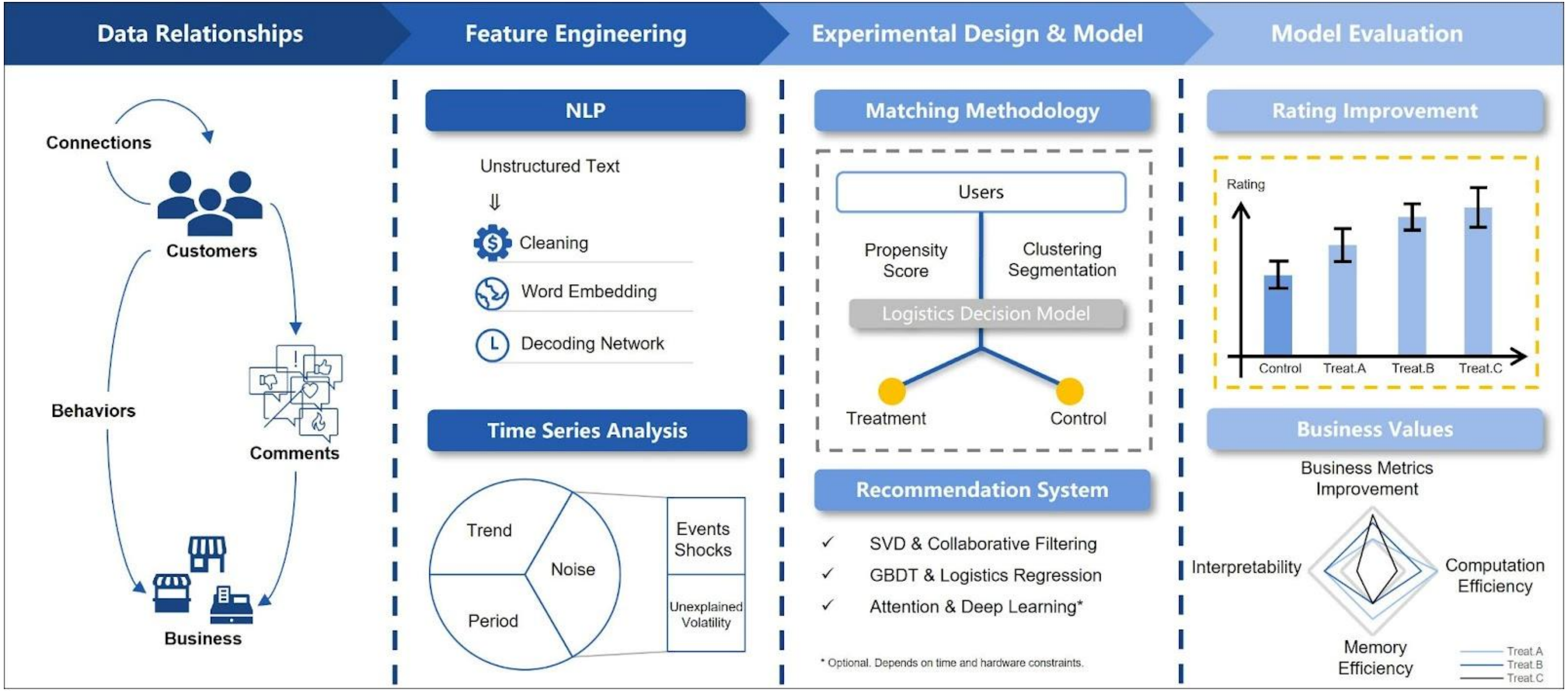
Team Introduction

Team Member	Role	Description
Jason Lee	Script	Note-taking for meetings
Milan Toolsidas	Quality Control	Checking if we meet the criteria of the assignments, formatting the PowerPoints and reports
Minglun Pan	Leader	Creating agendas for each meeting, making sure other roles' expectations are meet
Norah Zhang	Facilitator	Planning meetings, creating Zoom meetings when needed, making sure everyone's involved, timekeeping
Ryan Liao	Devil's Advocate	Providing constructive critical opinions, observing team dynamics

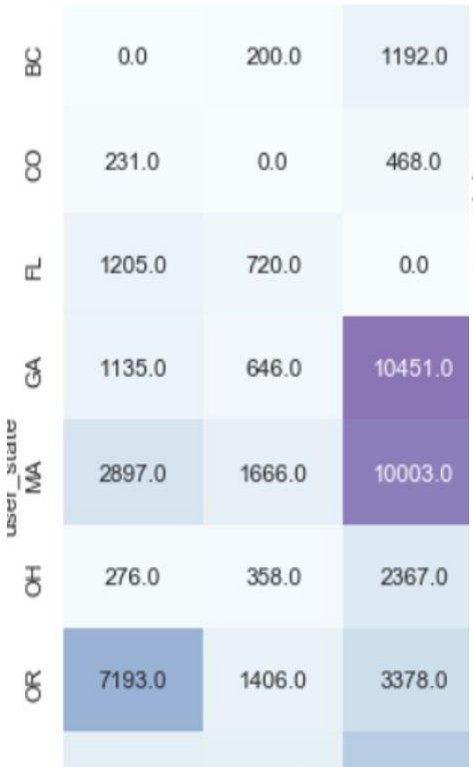
Project Workflow

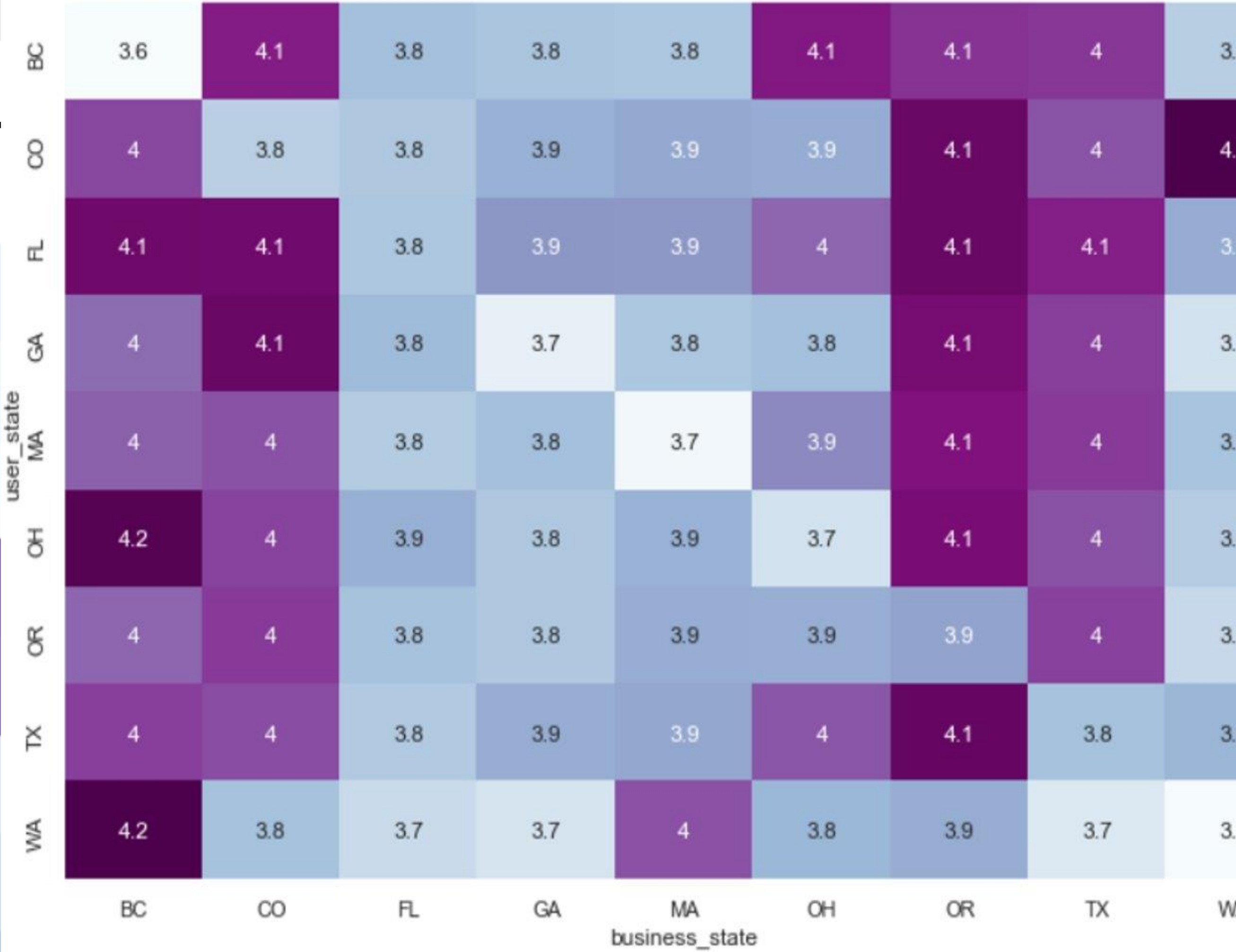


Project Workflow



EDA- Geograph





EDA - Time Series