YELP RECOMMENDATION SYSTEM

Presented by: Jason Lee, Melody Feng, Yue Liu, Steve Shi

MEET THE TEAM

Jason Lee

Melody Feng

Steve Shi

Yue Liu

AGENDA

01 **Executive Summary**

02 **Business Problem**

03 **Data**

- Data Profile
- Data Infrastructure
- Data Preparation

04 **Exploratory Data Analysis**

- Insight 1
- Insight 2
- Insight 3
- Insight 4
- Insight 5

05 **Recommendation System**

- ALS
- NLP
- Regression

06 **Project Execution**

EXECUTIVE SUMMARY

The **number of reviews dropped significantly** on Yelp during **Covid**. As the pandemic eases, we expect users would come back on Yelp to look for good restaurants to dine in at or attractions to go to. Yelp could **adjust** its **recommendation system** to **provide better suggestions** and search results to **retain users** and pursue its mission of connecting people with great local businesses.

In this project, we focused on analyzing restaurants in Ohio state, and...

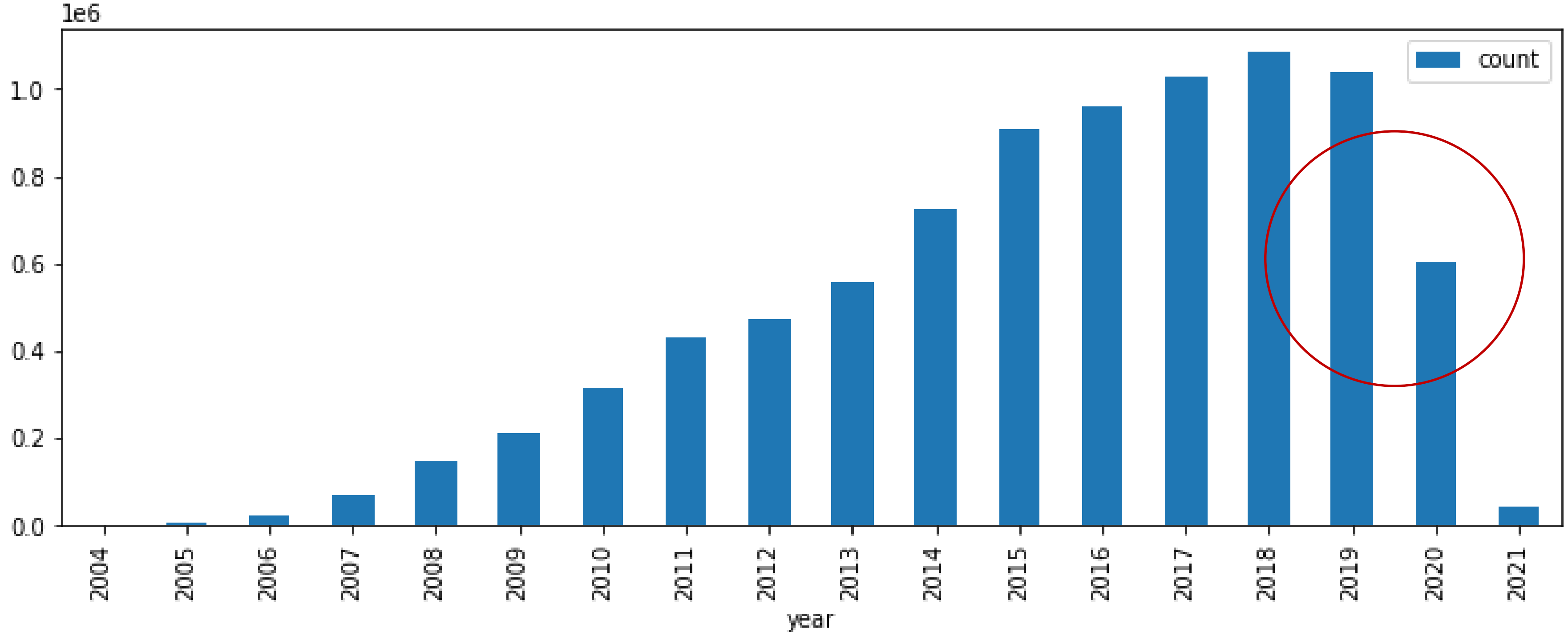
- Analyzed the Yelp dataset provided by Yelp.
- Stored our JSON raw data on **Google Cloud Platform Cloud Storage**.
- Utilized **Google BigQuery** as our data warehouse.
- Ran **PySpark** on Google **Dataproc** to clean the data and develop models.
- Trained the **Alternating Least Squares** model as a base recommendation model.
- Ran **NLP** on reviews to divide restaurants into several topics.
- Fit and trained **regression models** to predict how users rate restaurants and recommend the highest scored restaurants.

Our **base model** had an **RSME** score of **1.49** and **R2** of **85.9%**, and our **final model** had an **RSME** score of **0.89** and **R2** of **95.2**%. We are confident that we could make better suggestions to more active users with our final model. In the future, we could better optimize our model by **training on more data**, implementing **time series analysis**, and using **more robust NLP** models to understand our users better.

BUSINESS PROBLEM

During Covid, **the number of reviews dropped significantly on Yelp**. As pandemic eases, with the CDC dropping mask mandatories and vaccination checks, we expect a surge in people looking for good restaurants to dine in.

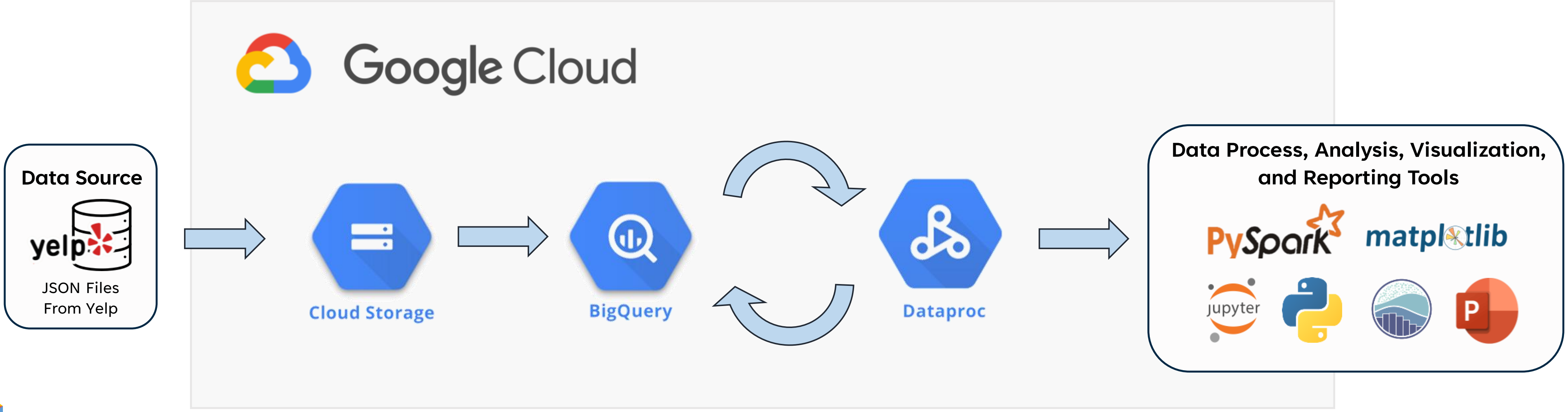
Yelp could adjust its recommendation system to provide better dining suggestions and search results for its users. In this project, we will focus on analyzing restaurants in Ohio. We aim to **boost user experience** by **recommending restaurants based on** their **past reviews**. Our recommendation engine analyzing past user reviews and deliver **more personalized recommendation for users.**



DATA PROFILE

	SOURCE	DESCRIPTION	DATA SIZE	FORMAT
Business	Yelp	Contain business information ‘attributes’ nested 22 variables ‘hours’ nested 7 variables	124MB	JSON
Review	Yelp	User reviews Large text variable	6.4GB	JSON
Tips	Yelp	User tips	230MB	JSON
User	Yelp	Contain user information ‘friends’ and ‘follow’ can have huge lists	3.68GB	JSON
Covid	Kaggle	Contain restaurant's covid features	30MB	JSON
Total			11.40GB	

DATA INFRASTRUCTURE



Data Lake

GCP Storage

Data Warehouse

Tables: Business, Reviews, User, and more

Data Science Platforms

Used Dataproc to run PySpark for data cleaning and analysis
Matplotlib, Pandas, Seaborn were used for visualization

DATA PREPARATION/CLEANING

Import data

- Import data into GCP buckets

Create Cluster

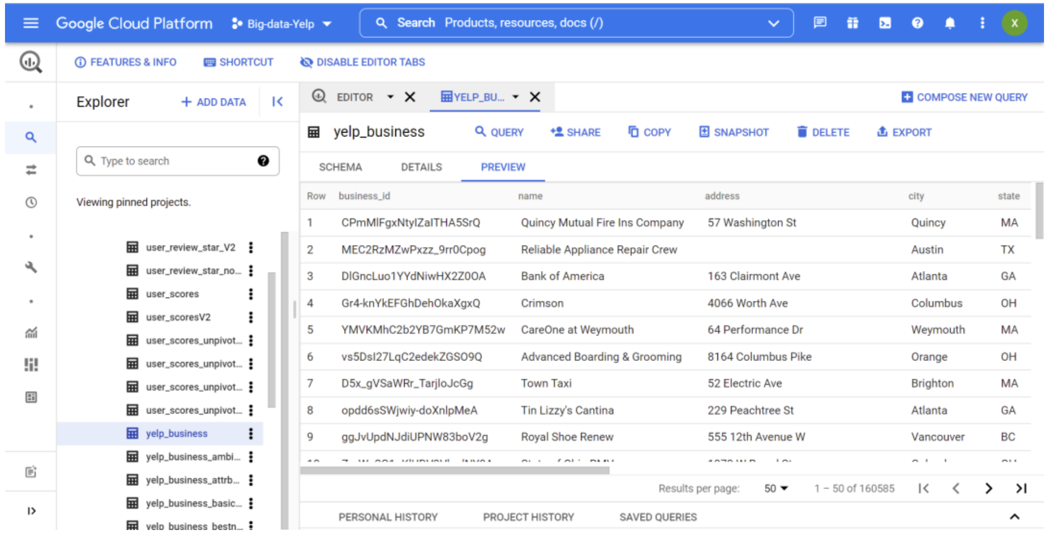
- Create cluster with 4 worker nodes
- Install necessary packages: pyspark.ml, pyspark.mllib,etc

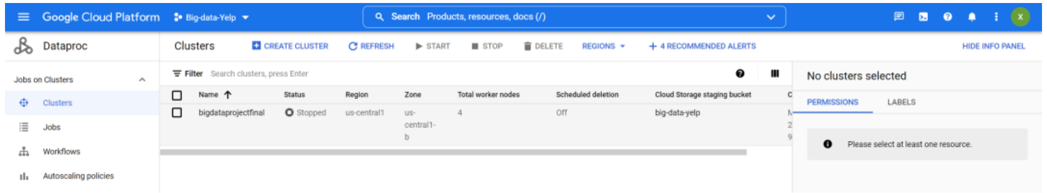
Big Query

- Import data into Big Query environment
- Separate nested columns from business table to multiple individual tables.

Data cleaning

- Drop/fill null values
- Convert data type

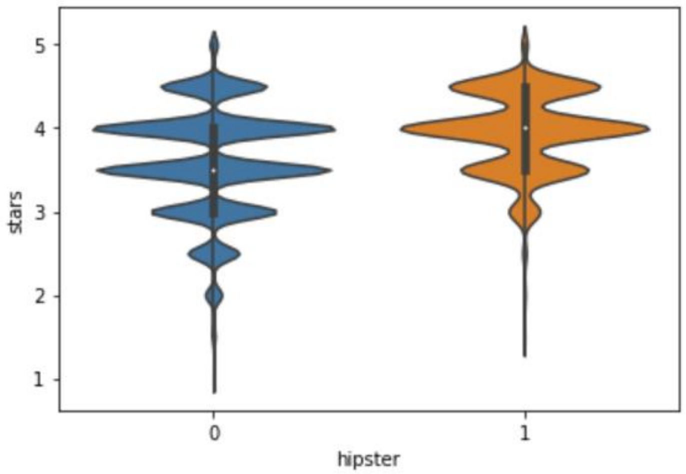


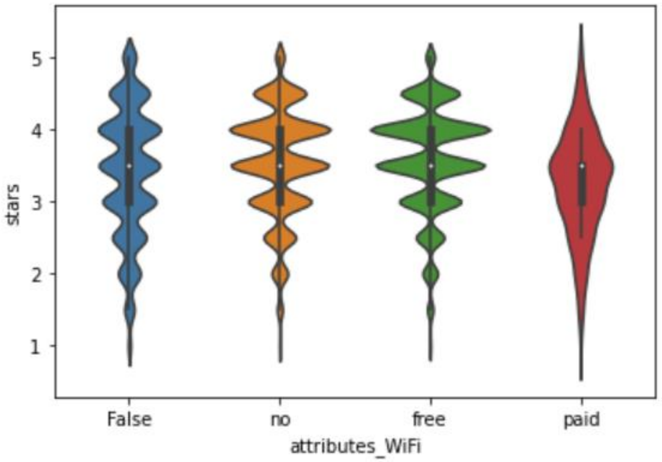


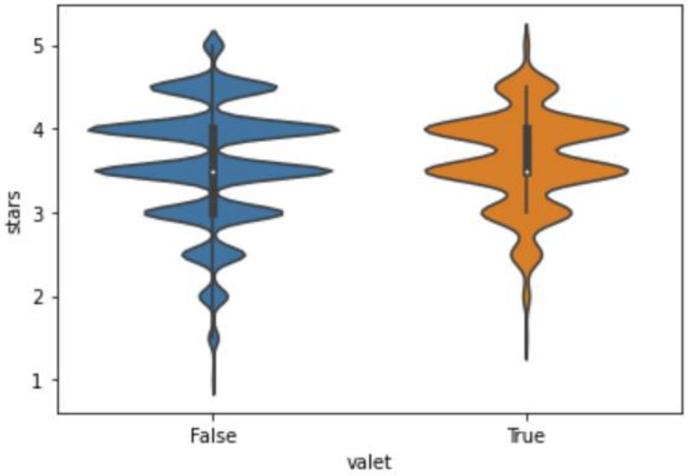
EXPLORATORY DATA ANALYSIS

Insight 1

- Several business attributes, such as **ambience** and **parking,** result in different **average rating**.
- We incorporate those features into our models.



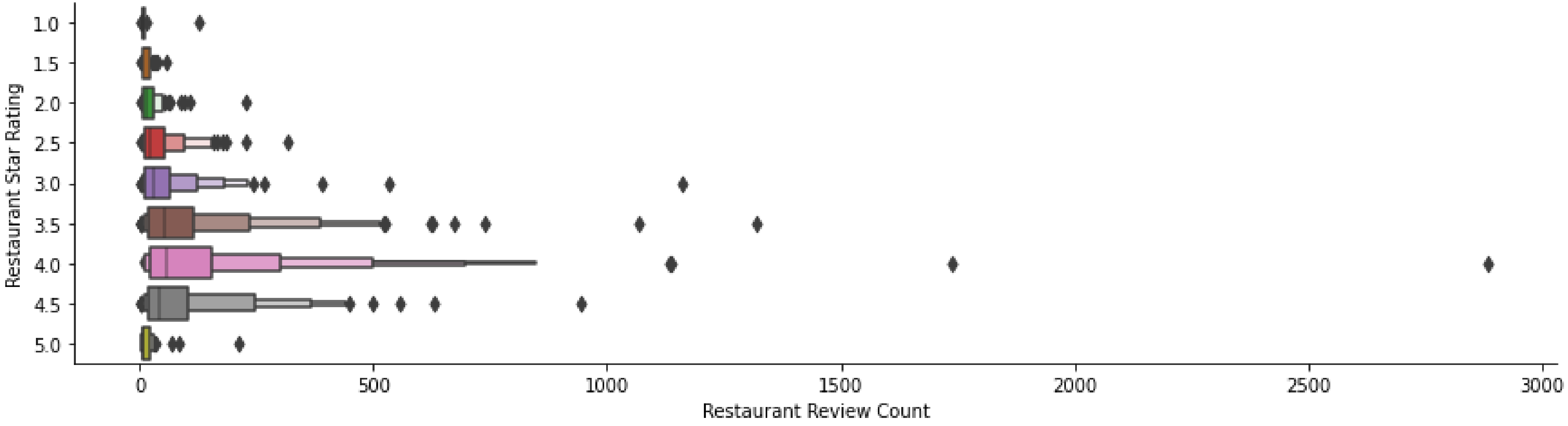




EXPLORATORY DATA ANALYSIS

Insight 2

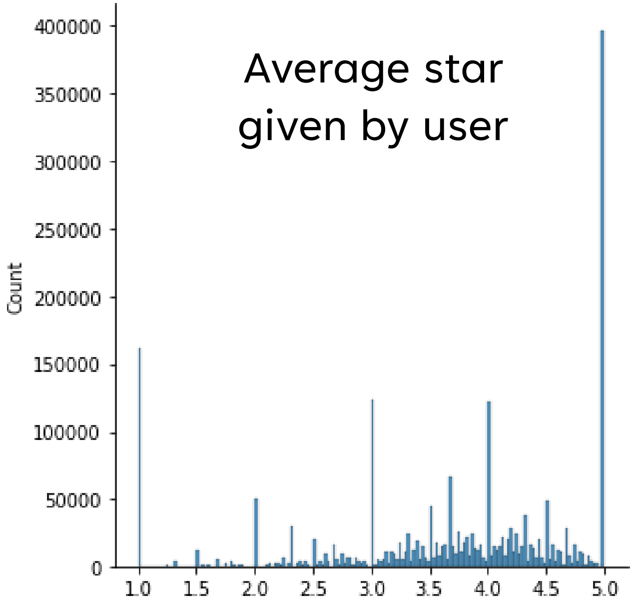
- **High review counts doesn't necessary mean that the restaurant is good.** In fact, the restaurants with the most reviews tend to have an average star rating.
- Five stars rated restaurants with high review counts should weight more than those with low review counts. We should reward those with high review counts in our model.

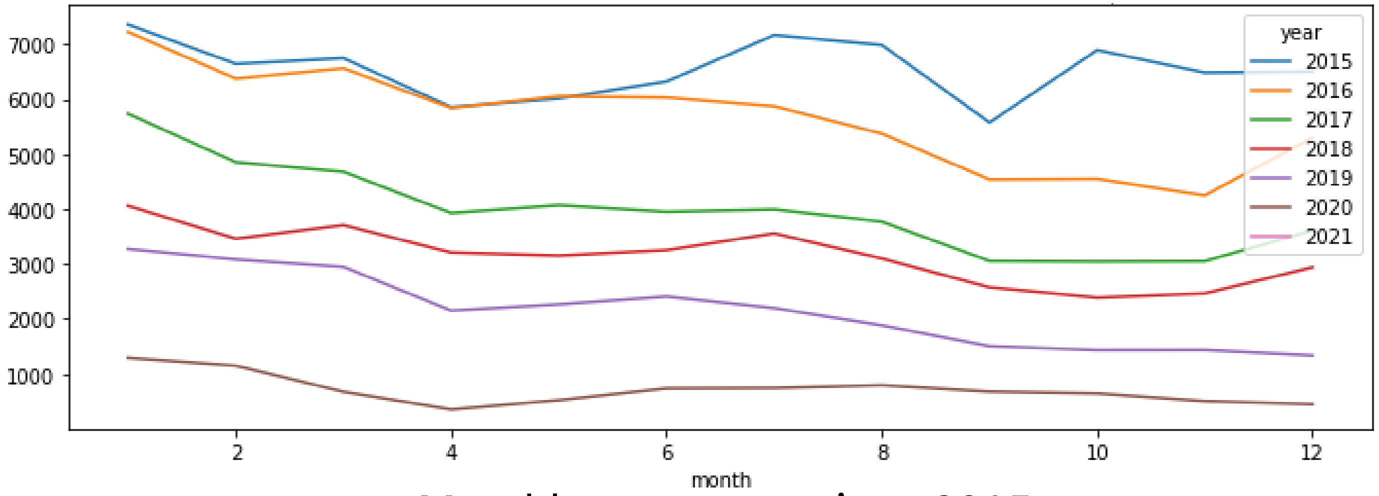


EXPLORATORY DATA ANALYSIS

Insight 3

- A lot of users like to give **extreme scores** if they **like/dislike** a restaurant.
- The Venn diagram shows what words user use to give extreme review scores.
- The **number of new users decrease** every year since **2015**





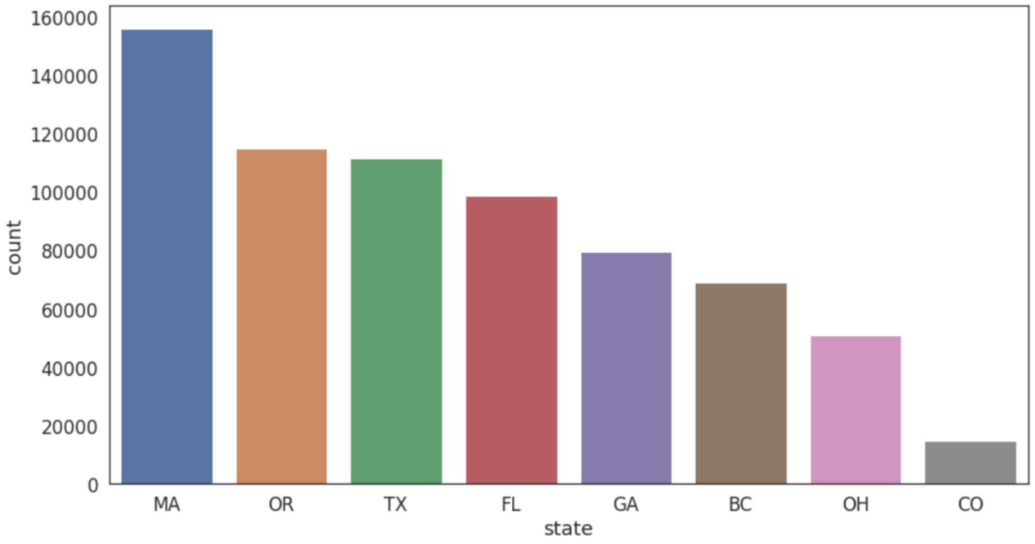
Monthly new users since 2015

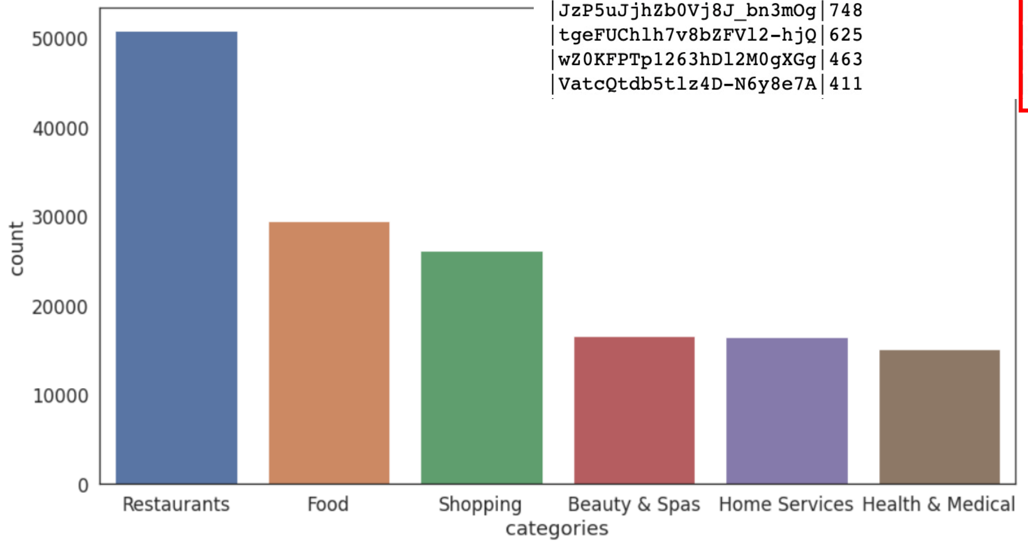
EXPLORATORY DATA ANALYSIS

Insight 4

- The business table contain multiple types of business. We decide to focus on **restaurants** for it is the **majority of all business types**
- We build our recommendation on scale of State because a lot of **users travel across cities** for restaurants

user_id	count(review_id)	count(city)
C1kTSvNdJH_S2bBhitr6ZA	917	20
R1FVpAyl_BtxHBWdau2VLg	890	29
JzP5uJjhZb0Vj8J_bn3mOg	748	27
tgeFUChlh7v8bZFVl2-hjQ	625	26
wZ0KFPTp1263hDl2M0gXGg	463	15
VatcQtdb5tlz4D-N6y8e7A	411	19

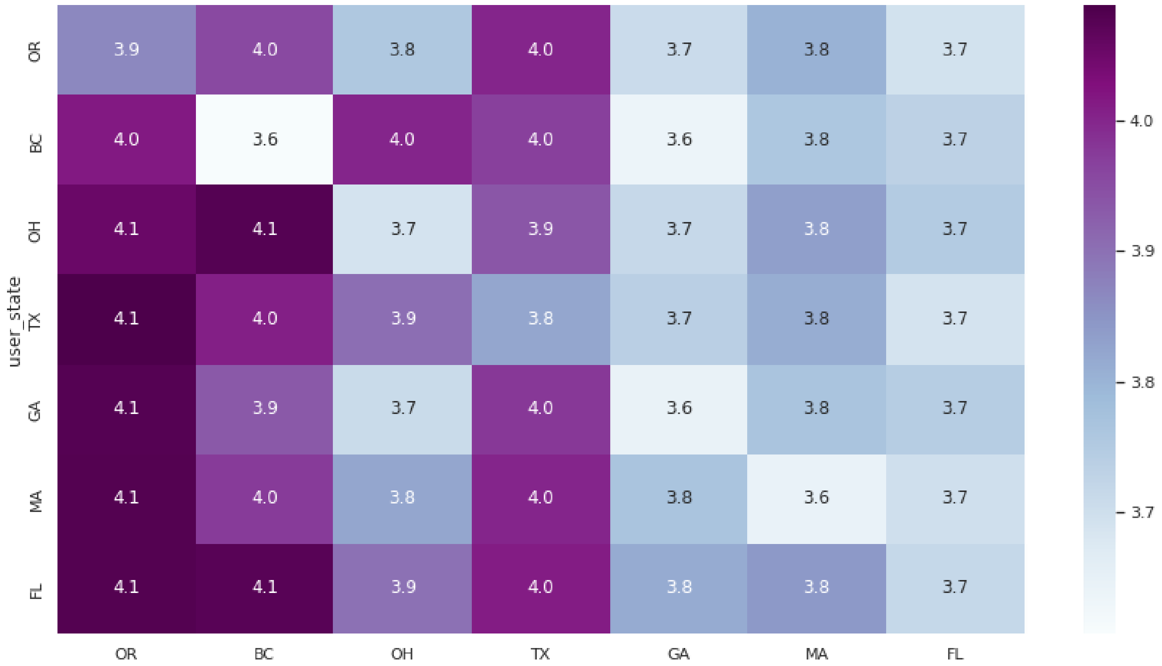


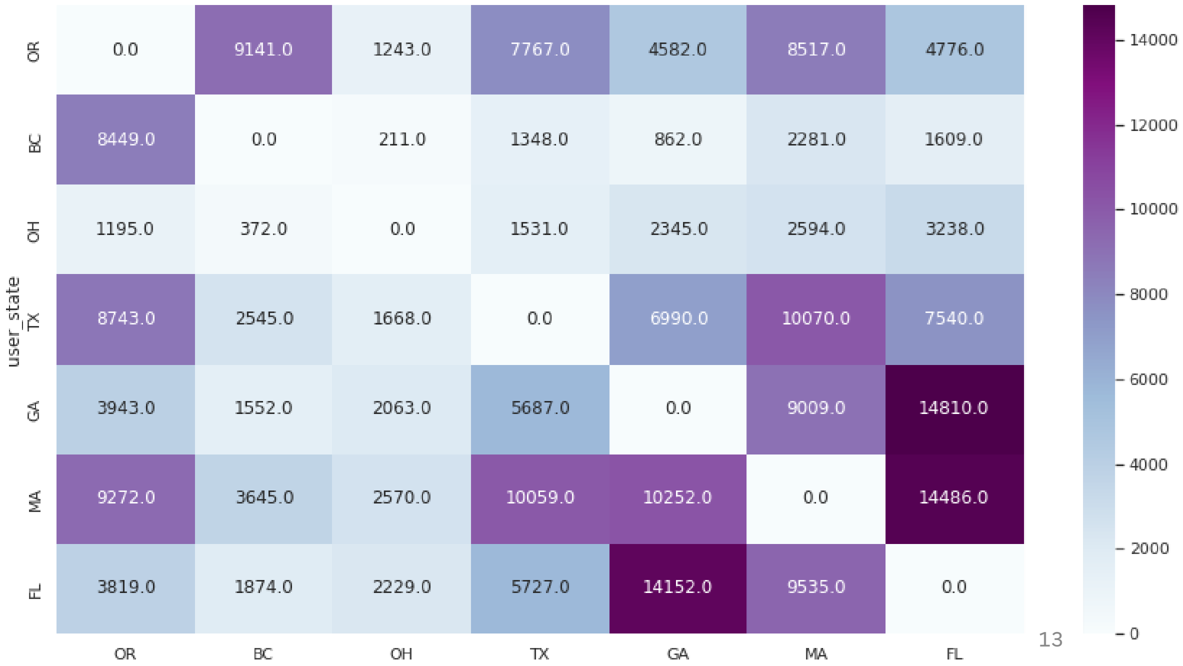


EXPLORATORY DATA ANALYSIS

Insight 5

- The first graph illustrates the **average star user** in each state gave to restaurants in different states. For example, users in Ohio only gave average star 3.7 to restaurants in Ohio, but gave 4.1 stars to restaurants in British Columbia
- The second graph shows the **number of reviews user in each state** gave to **restaurants in different states**. We can see users in Oregon gave a lot of reviews to restaurants at BC





RECOMMENDATION ENGINE - ALS

Feature Engineering

- The ALS model in Spark ML needs **numeric inputs** for ratingCol, itemCol, and userCol
- Used windows dense_rank to assign distinct integer index to business id and user id

Split and Train

- **Split dataset** into training and test sets (0.8, 0.2)
- Used **cross-validations**
- Joined with business table on business_id to show name and city of businesses

Recommend

- Generated business recommendations for each user
- Generated user recommendations for each business
- Generated business/user recommendations for a specified set of users/businesses

RECOMMENDATION ENGINE - ALS

A) Data Preparation

business_id	name	city	stars	user_id	business_id1	user_id1
qa4SegtG2bWMBhJgW...	Katalina's	Columbus	5.0	--1_pDM1pQ26cqhLx...	3724	1
32AcG_zpsPzMgo0aW...	Stack City Burger...	Columbus	4.0	--2PnhMMH7EYoY3wy...	257	2
8lS-sVYxXqVbhV8vj...	Hong Kong House	Columbus	4.0	--2PnhMMH7EYoY3wy...	656	2
AEzIqFtXrJITE4toG...	Mark Pi's Express	Columbus	3.0	--2PnhMMH7EYoY3wy...	758	2
IHCD--427ou0ODW6J...	Brazenhead	Dublin	4.0	--2PnhMMH7EYoY3wy...	1281	2

B) Final dataframe

stars	business_id	user_id
5.0	3724	1
4.0	257	2
4.0	656	2
3.0	758	2
4.0	1281	2

C) After running ALS, join with business table to view business name and city

name	city	business_id1	stars	business_id	user_id	prediction
The Royce	Columbus	1	1.0	1	33385	1.0111362
KFC	Hilliard	3	1.0	3	5365	1.873271
KFC	Hilliard	3	1.0	3	23723	2.2786682
KFC	Hilliard	3	1.0	3	75980	0.8214189
ZenCha Tea Cafe	Bexley	4	1.0	4	14959	3.5914705
Happy Wok	Pickerington	5	1.0	5	38133	3.576452
Morone's Italian Villa	Columbus	6	1.0	6	24690	1.5763088
McDonald's	Reynoldsburg	8	1.0	8	25834	0.4725808
McDonald's	Reynoldsburg	8	1.0	8	41119	0.23104912
Genji Japanese Steakhouse	Dublin	10	1.0	10	11862	2.5413952

RECOMMENDATION ENGINE - ALS

ALS is simple and scales well to very large datasets. Our model has a **RMSE** score of **1.49** and **R^2** of **85.9%**

A) Business recommendations for each user

user_id	recommendations
31	[{2217, 2.5845842...
34	[{2217, 3.1959221...
53	[{1337, 5.5929847...
65	[{3988, 5.333324}...
78	[{3231, 6.13207},...

B) User recommendations for each business

business_id	recommendations
28	[{24175, 5.60053}...
31	[{28777, 5.587188...
34	[{83083, 5.780057...
53	[{73340, 5.782814...
65	[{81733, 6.457925...

C) Users/businesses recommendations for a specified set of business/user

user_id	recommendations
1	[{2791, 5.7482758}, {2217, 5.485434}, {1029, 5.234603}, {627, 5.197041}, {2379, 5.079887}, {1115, 5.051845}, {129, 5.0370674}, {3370, 4.9783773}, {3549, 4.976562}, {2790, 4.96323}]
3	[{3988, 7.005713}, {2217, 6.525013}, {1953, 6.3541136}, {2859, 6.2487063}, {2977, 6.1182995}, {2036, 6.0590153}, {3231, 6.049763}, {1423, 6.0278826}, {4332, 6.0137014}, {3019, 5.9585557}]
2	[{2791, 6.118296}, {2217, 5.767094}, {129, 5.6899962}, {2036, 5.4956713}, {4187, 5.4802237}, {3538, 5.4736433}, {1239, 5.445331}, {2546, 5.35516}, {3793, 5.341137}, {1236, 5.3402057}]

business_id	recommendations
257	[{57986, 5.6981764}, {47788, 5.5353694}, {76830, 5.46328}, {38950, 5.4226103}, {36548, 5.411033}, {14080, 5.407138}, {15761, 5.4038715}, {1760, 5.3847013}, {80674, 5.3521786}, {7406, 5.3471394}]
3724	[{67688, 5.986271}, {16075, 5.819498}, {64902, 5.6489186}, {51840, 5.6404643}, {47784, 5.6356544}, {39797, 5.6146955}, {78118, 5.610285}, {22799, 5.586524}, {70127, 5.585662}, {80522, 5.577901}]
656	[{56327, 6.264087}, {42987, 5.8925776}, {44003, 5.5490704}, {23774, 5.527795}, {27256, 5.507446}, {14040, 5.492872}, {14330, 5.4794335}, {35408, 5.470227}, {78167, 5.463355}, {3520, 5.4199753}]

RECOMMENDATION ENGINE

Problem	Solution
ALS model's input only consisted of '**stars**' rating	**Regression** model can take in more predictors/variables that we **feature engineered** (Business Ambient, Parking Options, Popularity)
ALS model **did not** utilize **NLP**	**Regression** model can utilize topics from reviews generated by **NLP** as a predictor/variable
ALS model's star **rating scale** is **different** (max is more than 5 stars) from the rating users used (1-5 stars), resulting in a **high RMSE**	**Regression** model uses the **same scale of 1-5,** increasing **ease of understanding and lower RMSE**
ALS model has a **higher RMSE** than expected	Can run **multiple personalized regression models** for each user to **obtain better RMSE**

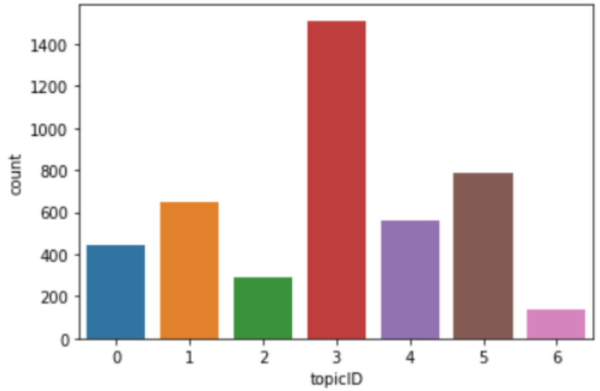
RECOMMENDATION ENGINE REGRESSION - NLP

SparkML pipeline

- Used a **SparkML** pipeline to build **NLP** models
- Assembled text into document, tokenized reviews, removed stop words, stemmed words

Topic Generation

- Applied **CountVectorizer** on tokens
- Applied **LDA** on vectorized tokens
- Extracted **7 topics** from LDA



Feature Engineering

- Generated **topic distribution** with LDA of each topic for each restaurant, **assigned the maximum topic to each restaurant**
- Averaged each topic's stars of restaurants by user id to see if there is a difference between topics

user_id	topic_0_star	topic_1_star	topic_2_star	topic_3_star	topic_4_star	topic_5_star	topic_6_star
wQT4QSglmm1c--0iT...	3.8	4.8	0.0	3.5	4.6	3.9583333333333335	4.333333333333333
2V6aMCtato51cIYBG...	0.0	0.0	0.0	0.0	0.0	1.6	0.0
kG3mjYoXQ9CGeIn M...	5.0	1.5	0.0	3.0	5.0	5.0	0.0

RECOMMENDATION ENGINE - REGRESSION

Feature Engineering

- Joined business, review, and user tables
- **Selected top categories** and added parking, ambience and 7 topics derived from NLP
- Feature-engineered column - **Popularity:** Normalized stars and reviews

$$(business\ stars - average\ business\ stars) * \sqrt{review\ count}$$

Split and Train

- Selected top users to train and recommend restaurants
- **Split dataset** into training and test sets (0.8, 0.2)
- Fitted **linear model**, **decision tree**, **random forest**, and **XGBoost**
- Ran grid search on **random forest**

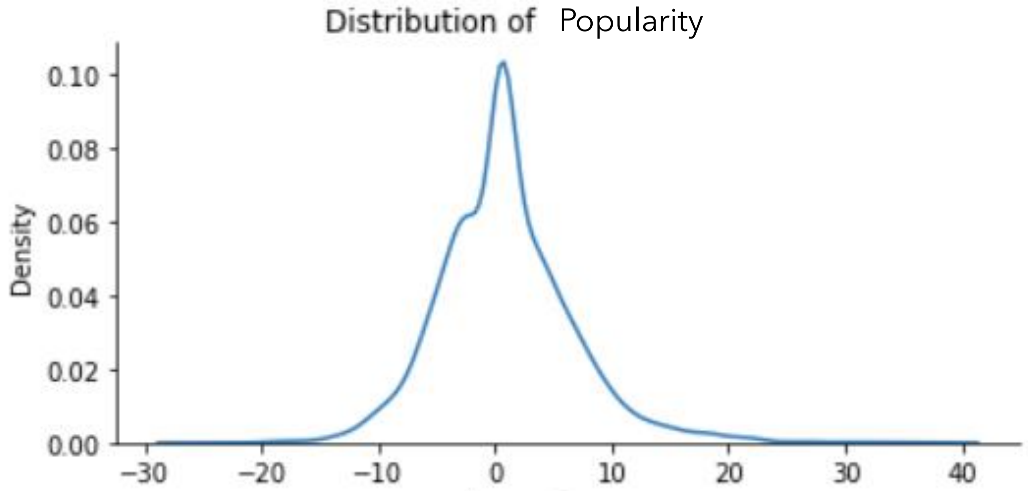
Recommend

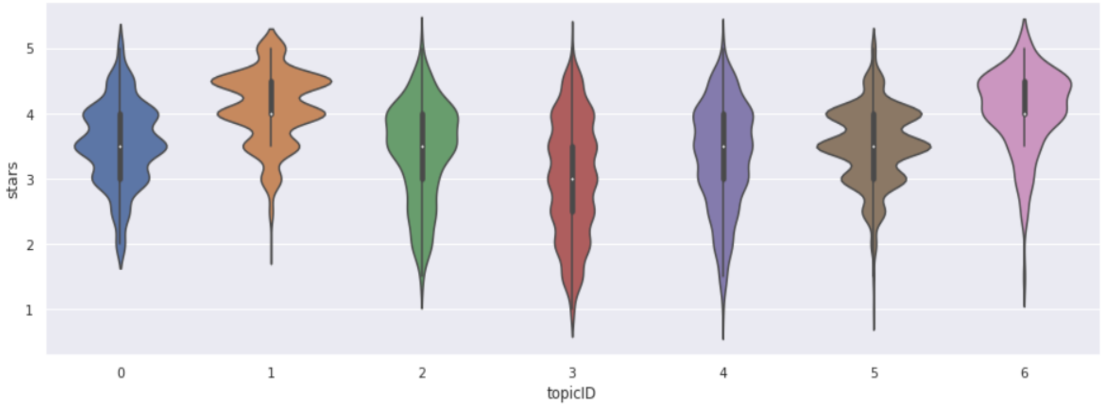
- Generated top business recommendations for top user based on predicted star rating

RECOMMENDATION ENGINE - REGRESSION

A) Data Cleaning

Attributes	Features
Basic Data	business_id, name, is_open
Categories	Nightlife, Bars, Fast_Food, American_Traditional, Sandwiches, Pizza, American_New, Burgers, Breakfast_Brunch, Mexican, Salad, Coffee_Tea, Chinese, Italian, Chicken_Wings
Parking	garage, lot, street, valet, validated
Ambient	casual, classy, divey, hipster, intimate, romantic, touristy, trendy, upscale
NLP	is_topic_0, is_topic_1, is_topic_2, is_topic_3, is_topic_4, is_topic_6, is_topic_5
Stars and Reviews	Popularity





RECOMMENDATION ENGINE - REGRESSION

B) Models

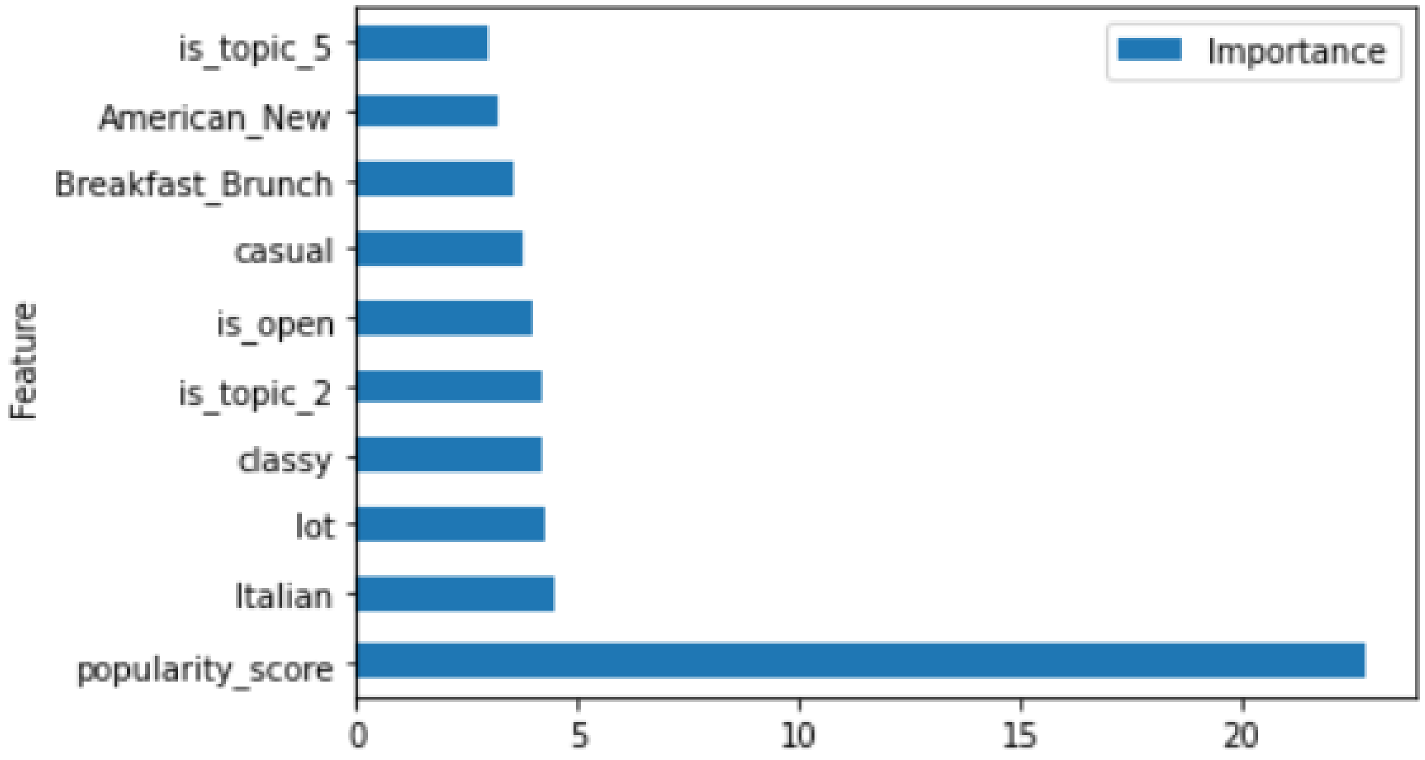
	Linear Regression	Decision Tree	Random Forest	XGBoost	Grid Search on Random Forest
RMSE on train	0.933318	0.877509	0.85789	0.50573	0.6625
RMSE on test	0.937383	0.974007	0.893568	1.17312	0.92972
R2 on train	0.947039	0.953183	0.955253	0.984607	0.973518
R2 on test	0.946286	0.942007	0.95119	0.912384	0.945058

C) Predictions drop duplicate

```
+--------------------+--------------------+--------------------+-----+------------------+
|         business_id|             user_id|                name|stars|        prediction|
+--------------------+--------------------+--------------------+-----+------------------+
|D6vNP2CBjP3Lg7Xid...|C1kTSvNdJH_S2bBhi...|Columbus Fish Market|  5.0|4.9700932224741745|
|B_W4Nq3-iFWV2ato5...|C1kTSvNdJH_S2bBhi...|The Refectory Res...|  5.0|  4.95669191919192|
|ewFMsE_X1PcS09yuO...|C1kTSvNdJH_S2bBhi...|J. Gilbert's Wood...|  5.0| 4.936651860157553|
|oRqgWTs4YBjEWCoz0...|C1kTSvNdJH_S2bBhi...|Gallo's Kitchen +...|  5.0| 4.932596473635128|
|yKyKvEqumEes4FOQY...|C1kTSvNdJH_S2bBhi...|  The Top Steakhouse|  5.0|4.8913504919596456|
|WQSziTOUaS36KC1es...|C1kTSvNdJH_S2bBhi...|        J Alexander's|  5.0| 4.858799435035911|
|4ergn03AcRW2kzgvZ...|C1kTSvNdJH_S2bBhi...|      The Old Mohawk|  4.0| 4.79865234982882 2|
```

- We will **recommend** the **top 10** restaurants that has the **highest predicted stars**
- The **predicted stars** predict **how many stars will the user give**. Higher score means that **user is more likely** to **like the restaurant**

REGRESSION – FEATURE IMPORTANCE



PROJECT EXECUTION TIMELINES

Week 1 – Week 2	Week 3 – Week 4	Week 5 – Week 6	Week 7 – Week 8	Week 8 – Week 9
• Creating GCP **VM clusters** • Uploading Datasets • Data **Cleaning** & **Preparing**	• **Exploratory Data Analysis**	• Feature engineering • Recommendation Engines - **ALS**	• **NLP** • Recommendation Engines - **Regression**	• **Evaluation** • Future suggestions

LESSONS LEARNED & RECOMMENDATIONS

Lessons Learned	Future Improvements
• **Data cleaning** takes up **60%** of the entire process • **EDA** and **Feature engineering** is an **essential** step to understand the dataset. Creating/mutating columns can have a huge impact on the model scores • Tuning **hyperparameter** could **improve performance** and **avoid overfitting/underfitting** • **GCP** provides a **synchronized working environment** • Cloud storage such as **GCP BigQuery** scales out **horizontally** to handle big data • **GCP Dataproc** clusters allow for processing of big data using **Apache Spark** by performing **parallel computing**	• **Data size:** Increase Data size, especially for ALS. We only used restaurants within Ohio • **Unused Data:** Geospatial location such as longitude and latitude to recommend locations within a certain distance of the user • **Sentimental Analysis :** Sentimental analysis on the individual reviews could help better understand users' preference (currently normalized stars and review counts to understand overall sentiment of reviews) • **Time Series Analysis:** Utilize time series analysis to recommend businesses according to time of day or season **Overall, adding data or implementing more machine learning methods could help improve results**

CONCLUSION

Business Problem:

The number of reviews dropped significantly since Covid

EDA:

Five insights
Found relevant features for regression model

ALS Base Model:

RMSE - 1.49

R2 - 85.9%

Regression Best Model:

RMSE - 0.89

R2 - 95.2%

Best Model:

Random Forest with grid search

THANK YOU