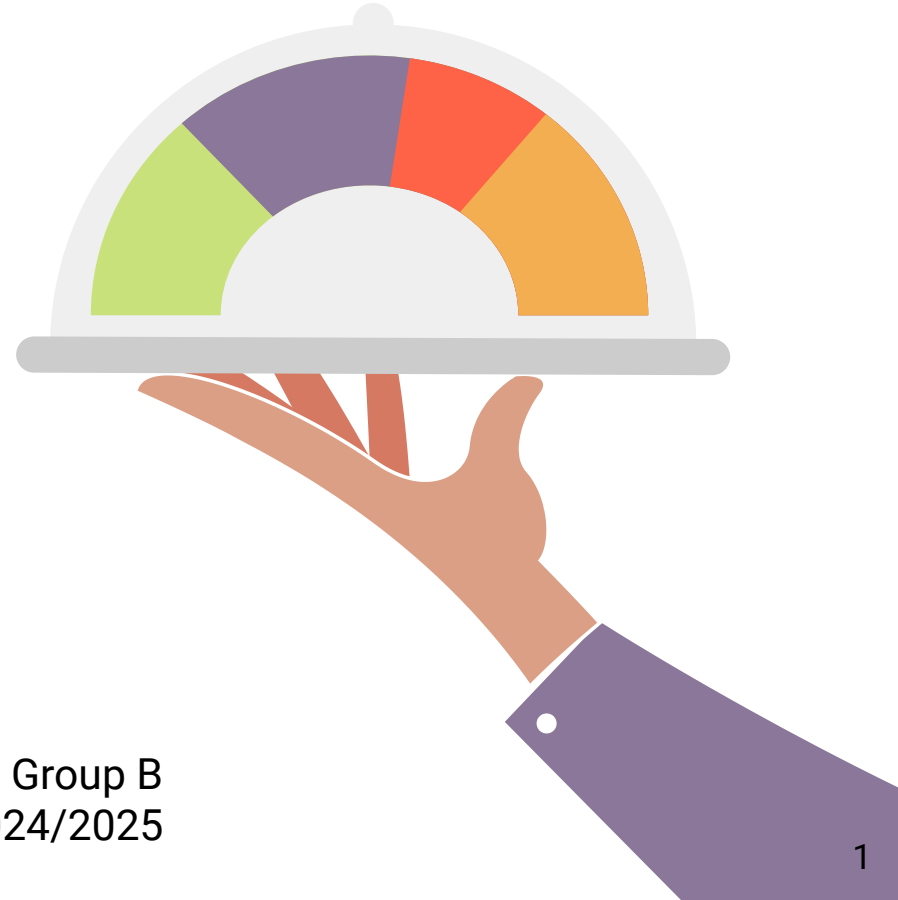
Restaurant Recommendation, Analysis and Exploration



U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Analysis of Complex Data - Group B
2024/2025

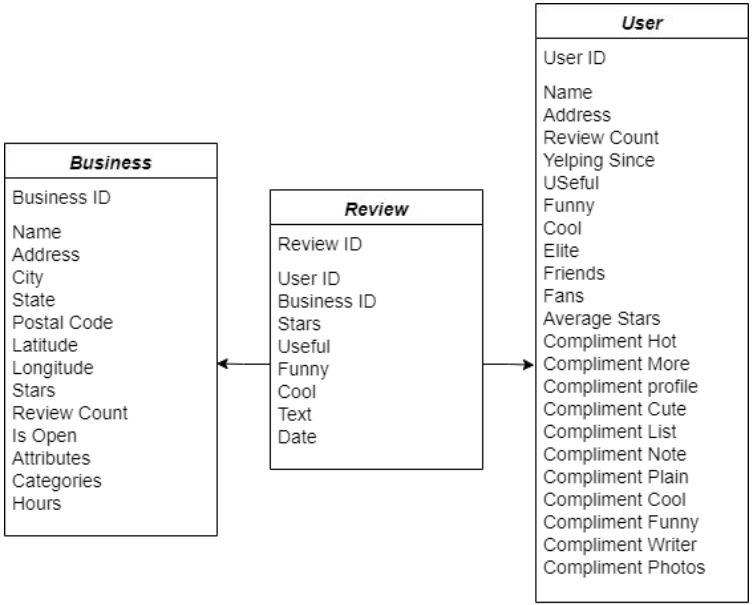
*To address the teacher's questions during the presentation, additional observations have been incorporated into this report. These changes are marked as: '**Additional observation**: XXX' throughout the slides.*

Contents

Dataset

Preview

- Yelp dataset
- Focusing on the Review, User and Business tables



Filter

- City with more businesses and reviews: Philadelphia
- Business with more reviews: Restaurants
- Only considering active businesses and restaurants and users with more than 5 reviews

Aggregate

- aggfunc='mean': when there are multiple reviews by the same user for the same restaurant, the function mean is used to aggregate the stars values, so, if a user has reviewed a restaurant multiple times, only the average rating is shown

	Original Dataset	**Our dataset**
# Users	1.987.929	178.325
# Business	150.346	3.525

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Part I: Content-based Recommender System with NLP

Baseline

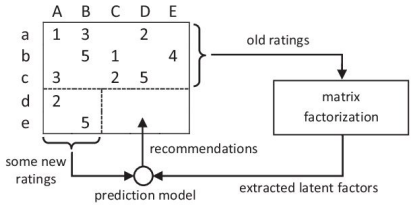
- As a baseline approach to establish the first set of results, several methods were tried for the recommender system.

- The methods included Collaborative Filtering, both user-based and item-based, each with two different similarity measures - cosine similarity and MSD.

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} \text{sim}(u,v) \cdot r_{vi}}{\sum_{v \in N_i^k(u)} \text{sim}(u,v)}$$

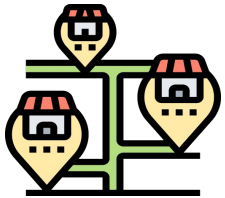
- Additionally, a model-based approach in the form of Matrix Factorization was also included for variety.

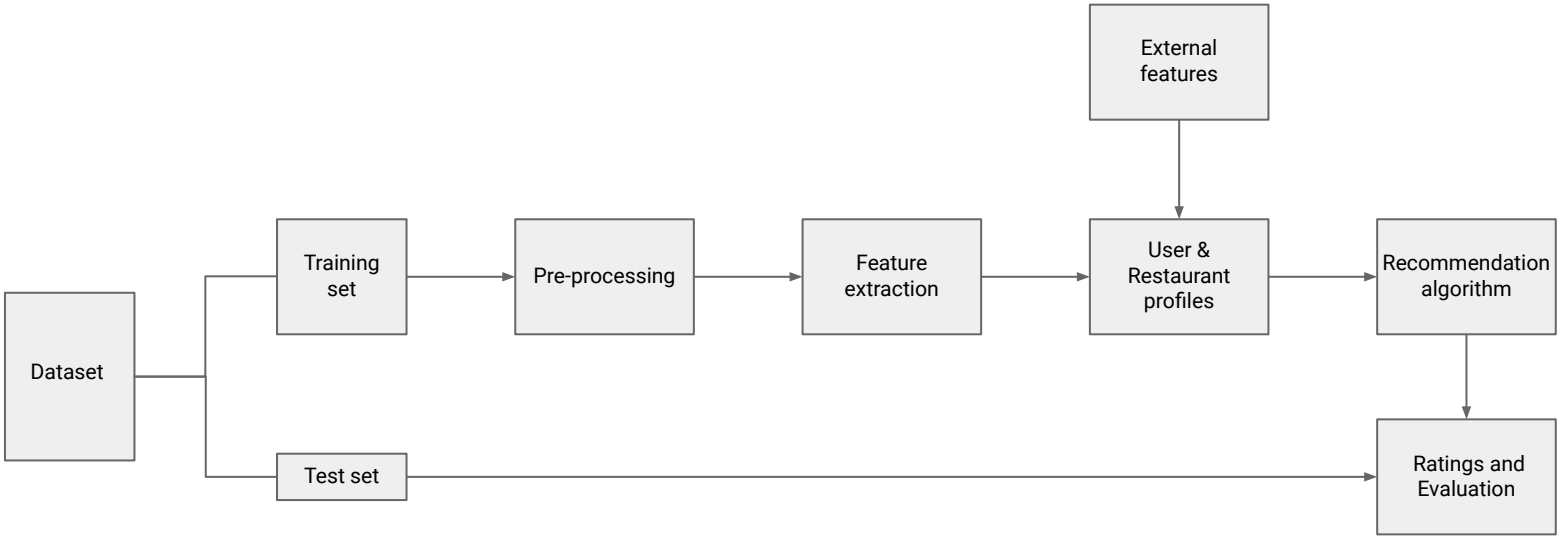
$$M_k = U_k \times \Sigma_k \times V_k^T$$



Hybrid Approach

Content-based





Pre-processing

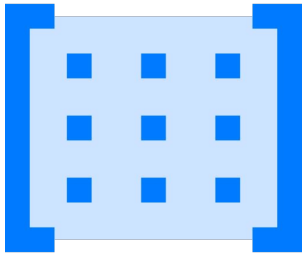
To reduce the vocabulary size we applied lemmatization and stemming techniques. These methods allowed us to standardize words with different grammatical forms or variations, effectively consolidating them into a single form. We were able to minimize redundancy and ensure a more concise and uniform vocabulary. We also removed stopwords and punctuation. We chose not to convert everything to lowercase due to instances like “EXCELLENT,” as we can see bellow, where the capital letters clearly emphasized a positive sentiment, and so we wanted to retain this intensity in our analysis. Preserving the original casing allowed us to capture and leverage these nuances, ensuring that important contextual information wasn't lost.

	Original	With Lemmatization	With Stemming
0	Wife and I both had the lobster rolls. They d...	Wife I lobster rolls. They skimp meat deliciou...	wife i lobster rolls. they skimp meat deliciou...
1	I just visited Ed's pizza and I specifically t...	I visited Ed's pizza I specifically told woman...	i visit ed' pizza i specif told woman counter ...
2	Had the pho here. Definitely tasty and afforda...	Had pho here. Definitely tasty affordable opin...	had pho here. definit tasti afford opinion. i'...
3	This place is unbelievably good. We popped in ...	This place unbelievably good. We popped yester...	thi place unbeliev good. we pop yesterday earl...
4	Flambo was EXCELLENT from the service to the f...	Flambo EXCELLENT service food everything betwe...	flambo excel servic food everyth between. i es...
5	Best bagels in Fishtown/ Northern Liberties, h...	Best bagel Fishtown/ Northern Liberties, hand ...	best bagel fishtown/ northern liberties, hand ...
6	100% the best doughnuts my taste buds will eve...	100% best doughnut taste bud ever honor tastin...	100% best doughnut tast bud ever honor tasting...
7	Amazing food !!!!! Also the service was perfec...	Amazing food !!!!! Also service perfect!!! The...	amaz food !!!!! also servic perfect!!! the ser...
8	Aside from the fact that they are super nice t...	Aside fact super nice bread, danish coffee exc...	asid fact super nice bread, danish coffe excel...
9	I must say, if there was 5 stars for rudeness,...	I must say, 5 star rudeness, whole staff would...	i must say, 5 star rudeness, whole staff would...

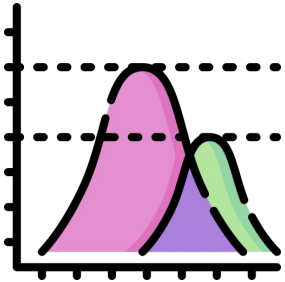
Feature Extraction

We experimented various techniques, including Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), and Doc2Vec, to capture meaningful patterns and relationships within the text data. These approaches allowed us to model topics and semantic similarities effectively. We decided against using the bag-of-words method, as its high dimensionality would have made it computationally expensive and impractical for our dataset size further down in our pipeline. For LSA and LDA we fixed the number of topics at 8, and for Doc2Vec we chose 100 dimensions.

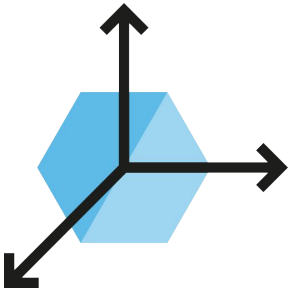
Bag-of-words



Latent Semantic Analysis



Latent Dirichlet Allocation



Doc2Vec

External Features

User

Given that we still had information about the users beyond the reviews they wrote, we wanted to incorporate these variables into the dataset to enrich our model. This additional user data included features such as:

Review Count

This feature represents the number of reviews each user wrote.
This variable serves as an indicator of user activity and engagement level on the platform. It also helps differentiate between casual users, who might only post occasionally, and frequent users.

Average of Ratings

The goal of this variable is to give us the average of all the ratings of an user. This offers valuable context for interpreting their reviews and preferences. It allows us to identify whether users are generally more critical — some users may rarely give 5 stars, even when they enjoy a place, while others may tend to rate more positively overall.

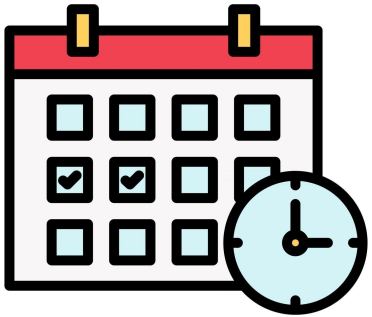
External Features
User

Years on Yelp

This feature was created based on the user's sign-up date, converting it into the number of years the user has been active on the platform. This variable provides insight into the user's longevity and loyalty to Yelp. Users with more years on the platform may have accumulated more experience and credibility, making their reviews potentially more valuable. It also helps differentiate between new users.

External Features
Business

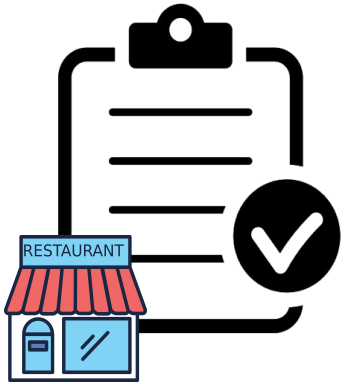
We also had detailed information about the restaurants beyond the reviews they received, so we wanted to incorporate these variables into the dataset. We aimed to provide a more comprehensive understanding of each restaurant's reputation, longevity, and appeal to different users demographics. This additional restaurant data included:



Schedule

Initially we had a hours feature that consisted in dictionaries where the keys represented the days of the week, and the values indicated the restaurant's opening and closing hours. To simplify and standardize this variable, we transformed it into seven individual features, each representing a day of the week with its corresponding category based on the opening and closing times. By converting the schedule into defined categories, the model can better assess and compare businesses according to their availability during specific time slots, thus improving the precision of our recommendations based on user preferences and timing.

External Features Business



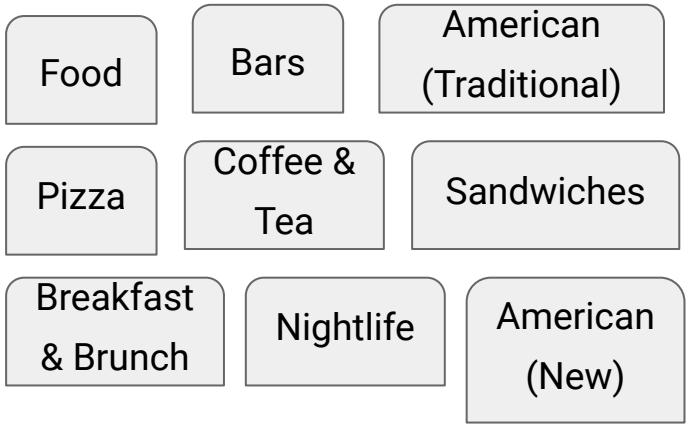
Review Count

This feature represents the number of reviews each business has received. This variable serves as an indicator of the business's popularity and visibility on the platform. It also helps differentiate between less-known establishments, which may receive reviews infrequently, and highly popular businesses that attract a larger number of reviews.

Average Rating

The goal of this variable is to provide the average of all the ratings a business has received. This offers valuable context for interpreting the restaurant's overall reputation and customer satisfaction levels.

External Features Business



Top 9 Categories

The *categories* variable originally contained multiple categories for each restaurant. We analyzed the most frequently occurring categories and identified the top 9, which stood out significantly as they appeared in many more restaurants than the others. To make this feature more manageable and useful for our model, we created 9 separate variables, one for each of these top categories. Each variable is binary, with a value of 1 if the restaurant belongs to that category and 0 otherwise. This transformation allowed us to capture the most relevant and common categories while simplifying the analysis.

External Features Business

Top 9 Attributes

RestaurantsTakeOut	RestaurantsGoodForGroups	RestaurantsDelivery	BikeParking	HasTV
BusinessAcceptsCreditCards	GoodForKids	RestaurantsAttire_casual	BusinessParking_street	

We applied a similar approach to the attributes variable, where each value was originally a dictionary with keys representing attributes and boolean values indicating their presence. We transformed each key into a separate variable. However, some keys had values that were also dictionaries, as shown below. In these cases, were created variables by combining the main key with each of its sub-keys. Once all attributes were properly distributed into variables, we analyzed the distribution of each one to assess the proportion of true and false values. Based on this analysis were identified the attributes with the highest frequency of true values, which allowed us to understand the most relevant and common features.

```
{'BusinessParking': "{'garage': False, 'street': True, 'validated': False, 'lot': False, 'valet': False}"
```

User & Restaurant Profiling

The initial feature extraction based on NLP techniques was performed for each document (review text). Each review would then be represented by a vector. In order to obtain the profile of a user, the vectors that represent the reviews done by the user were average across all dimensions which hopefully translates into a fair vectorized representation of the user's preferences. The same thing was done to each restaurant to obtain the restaurant profiles but this time with the comments that were done towards the restaurant. This was done to users and restaurants with at least 5 reviews to ensure some variety. Finally, the external features described before were added to the respective user/restaurant and this way, every user, as well as every restaurant, was characterized by a single vector.

User profiles matrix

Restaurant profiles matrix

Rating Prediction

User-based approach

With the two matrices we can now move to the rating prediction phase. For this we have 3 different methods, the first one being user-based.

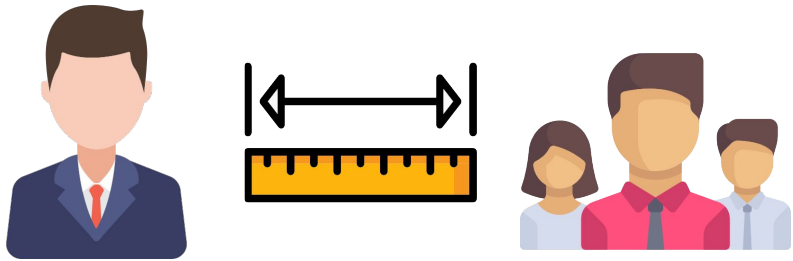
1. A user and a restaurant are provided;

2. Other users that reviewed the restaurant are retrieved as well as their given ratings;

3. The user profiles are retrieved from the User Profiles matrix;

4. The cosine similarity between the given user and every user retrieved is calculated;

5. The predicted rating is calculated as a weighted average using the similarities as weighting on the ratings given by each retrieved user.



Rating Prediction

Item-based approach

The second method is similar to the first, but applied to the restaurants.

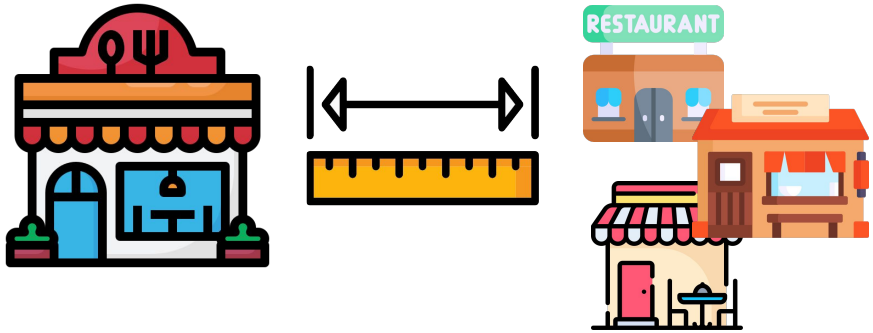
1. A user and a restaurant are provided;

2. Other restaurants reviewed by the user are retrieved as well as the given ratings;

3. The restaurant profiles are retrieved from the Restaurant Profiles matrix;

4. The cosine similarity between the given restaurant and every restaurant retrieved is calculated;

5. The predicted rating is calculated as a weighted average using the similarities as weighting on the ratings given to each retrieved restaurant.



Rating Prediction

User-item approach

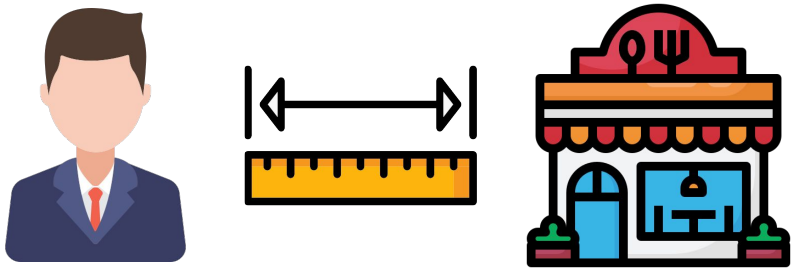
The last one compares users and restaurants directly. It's important to note that for this method the external features were removed for two reasons: 1) the vectors didn't have the same number of dimensions - restaurants had much more external features than users; 2) most importantly, the features didn't mean the same for users and restaurants, they are not in the same vector space as the topics from LSA/LDA or dimensions from Doc2Vec are.

1. A user and a restaurant are provided;

2. The vector representing the user and the one representing the restaurant are retrieved from the respective matrices;

3. The cosine similarity between the two vectors is computed;

4. The predicted rating is a linear transformation of the similarity that maps it from the original range of [-1;1] to [1;5] as the other ratings.

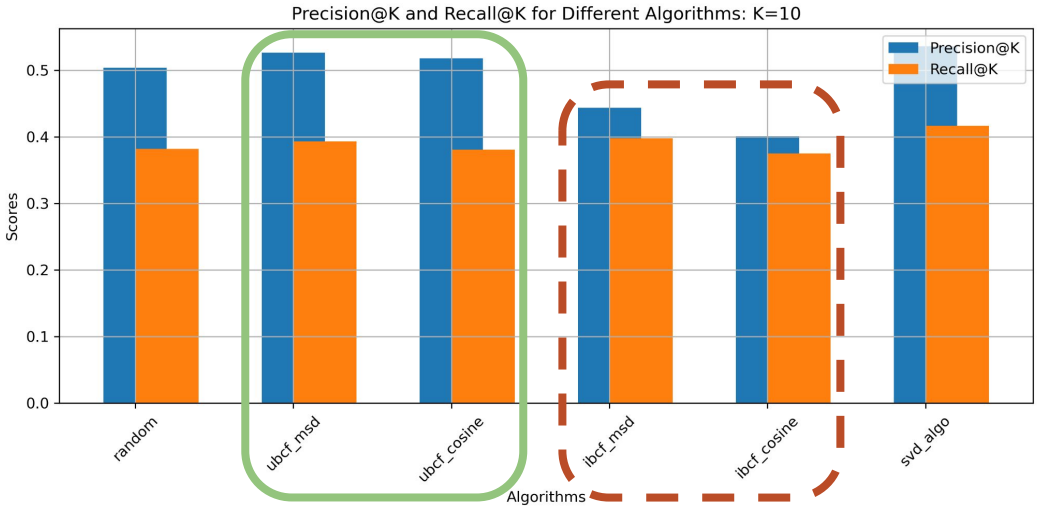


Evaluation

Baseline

		RMSE	Precision@3	Recall@3	Precision@5	Recall@5	Precision@10	Recall@10
Baseline	Normal Predictor	1,47	0,51	0,35	0,50	0,37	0,51	0,39
	KNN User-based	1,06	0,53	0,34	0,52	0,37	0,53	0,38
	KNN Item-based	1,09	0,41	0,32	0,41	0,35	0,41	0,38
	SVD	**1,00**	**0,55**	**0,38**	**0,54**	**0,40**	**0,55**	**0,42**

- ➔ Random = Normal Predictor
- ➔ The best overall method seems to be SVD.
- ➔ User-based CF (in green line) has a slight edge over Item-based CF (red line), and is sliglty better overall, specially on precision metrics.



Evaluation

Pre-processing	Features	Algorithm	RMSE	Precision@3	Recall@3	Precision@5	Recall@5	Precision@10	Recall@10
Lemmatization	Doc2Vec	IBH	1,29	0,21	0,06	0,37	0,21	0,35	0,18
Lemmatization	Doc2Vec	UBH	0,97	0,73	0,28	0,75	0,48	0,65	0,49
Lemmatization	Doc2Vec	UIBH	1,04	0,58	0,22	0,63	0,40	0,60	0,42
Lemmatization	LDA	IBH	1,28	0,15	0,05	0,32	0,18	0,35	0,18
Lemmatization	LDA	UBH	0,96	0,75	0,29	0,76	0,49	0,65	0,49
Lemmatization	LDA	UIBH	1,10	0,65	0,25	0,65	0,41	0,65	0,52
Lemmatization	LSA	IBH	1,28	0,15	0,05	0,31	0,17	0,30	0,16
Lemmatization	LSA	UBH	0,97	0,73	0,29	0,75	0,48	0,65	0,49
Lemmatization	LSA	UIBH	1,28	0,62	0,24	0,68	0,44	0,60	0,42
Stemming	Doc2Vec	IBH	1,25	0,21	0,06	0,36	0,21	0,35	0,18
Stemming	Doc2Vec	UBH	0,96	0,75	0,29	0,75	0,48	0,65	0,49
Stemming	Doc2Vec	UIBH	1,04	0,58	0,21	0,67	0,42	0,55	0,40
Stemming	LDA	IBH	1,27	0,15	0,05	0,32	0,18	0,35	0,18
Stemming	LDA	UBH	0,96	0,75	0,30	0,75	0,48	0,65	0,49
Stemming	LDA	UIBH	1,09	0,60	0,22	0,65	0,41	0,60	0,50
Stemming	LSA	IBH	1,30	0,15	0,05	0,31	0,17	0,30	0,16
Stemming	LSA	UBH	0,97	0,74	0,29	0,75	0,48	0,65	0,49
Stemming	LSA	UIBH	1,28	0,65	0,25	0,67	0,43	0,60	0,42

Evaluation

Hybrid approach

		RMSE	Precision@3	Recall@3	Precision@5	Recall@5	Precision@10	Recall@10
Pre-processing	Lemmatization	1,13	0,51	0,19	0,58	0,36	0,53	0,37
	Stemming	1,12	0,53	0,18	0,59	0,38	0,52	0,37
Feature extraction	LDA	1,11	0,51	0,19	0,58	0,35	0,54	0,39
	LSA	1,17	0,50	0,19	0,58	0,36	0,52	0,36
	Doc2Vec	1,09	0,54	0,21	0,59	0,38	0,53	0,36
Recommendation algorithm	**UBH**	**0,96**	**0,75**	**0,29**	**0,75**	**0,48**	**0,65**	**0,49**
	IBH	1,28	0,17	0,05	0,33	0,18	0,33	0,18
	UIBH	1,14	0,62	0,23	0,66	0,42	0,60	0,45

*Averages

Additional observation:

- ➔ For the hybrid approach we computed 18 different combinations of pre-processing methods, feature extraction methods and recommendation algorithms. (2*3*3=18)
- ➔ All these combinations are summarized in the table above by averaging the results per method, as in a “groupby” operation.
- ➔ Results in the ‘Stemming’ line for pre-processing, are the average results for all the combinations of feature extraction and recommendation algorithms that used Stemming as a pre-processing method.
- ➔ The full table is shown below on the previous slide.

Evaluation

Hybrid approach

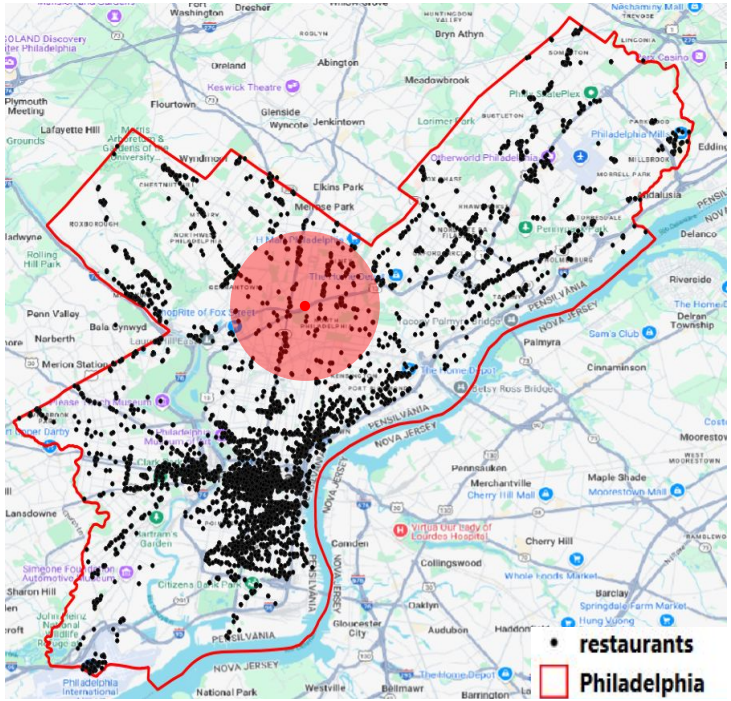
		RMSE	Precision@3	Recall@3	Precision@5	Recall@5	Precision@10	Recall@10
Pre-processing	Lemmatization	1,13	0,51	0,19	0,58	0,36	0,53	0,37
	Stemming	1,12	0,53	0,18	0,59	0,38	0,52	0,37
Feature extraction	LDA	1,11	0,51	0,19	0,58	0,35	0,54	0,39
	LSA	1,17	0,50	0,19	0,58	0,36	0,52	0,36
	Doc2Vec	1,09	0,54	0,21	0,59	0,38	0,53	0,36
Recommendation algorithm	**UBH**	**0,96**	**0,75**	**0,29**	**0,75**	**0,48**	**0,65**	**0,49**
	IBH	1,28	0,17	0,05	0,33	0,18	0,33	0,18
	UIBH	1,14	0,62	0,23	0,66	0,42	0,60	0,45

*Averages

- ➔ All are **better than random** (previous slide)
- ➔ We didn't find any meaningful difference in results between pre-processing methods.
- ➔ Regarding feature extraction, Doc2Vec achieved, on average, the best results.
- ➔ The biggest differences were found at the recommendation algorithm level where the User-based Hybrid approach was at the top.
- ➔ Computationally, LSA is much faster than LDA, which is much faster than Doc2Vec.
- ➔ UIBH is much faster than UBH and IBH.
- ➔ IBH has significantly worse performance than UBH likely because of the lack of comparable examples. I.e. it is easier to correctly predict a rating based on a large number of similar users (UBH). IBH relies on the existing reviews of the user which are usually very limited and there is not necessarily a restaurant to make a good comparison with within those reviews.

Deployment

- ➔ Input is the user, its location (which we generate within the boundaries of the city), and the radius of search.
- ➔ Using the location of the user, the restaurants outside the buffer are excluded.
- ➔ Recommendation is given for the remaining restaurants.
- ➔ For new users (**cold start**), the system still filters out the restaurants outside the boundaries and recommends based on the average rating of each restaurant.
- ➔ Note: distance is calculated as the crow flies (straight line).



U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Part II: Reviews Forecasting based on Network Communities

Find Communities of Restaurant categories

To visualize how different restaurant categories are interconnected based on the frequency of restaurants that belong to multiple categories, we made several important adjustments.

- Firstly, some services are labeled as restaurants but don’t fit the traditional definition. For instance, gas stations with food options, arts and entertainment venues with a jazz or blues ambiance, and airport lounges restricted to certain areas are not true restaurants and shouldn't be included.

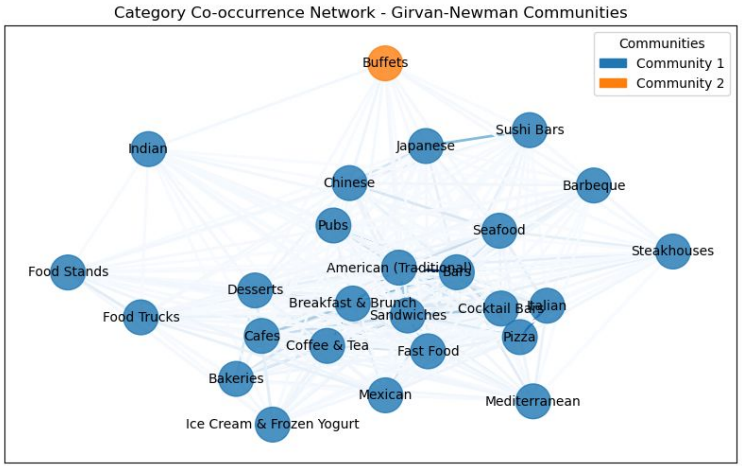
- To ensure accuracy, we defined a more rigorous set of restaurant categories: only restaurants that are **combinations of our predefined categories** were included in the analysis.
- Any establishment with categories outside this scope was excluded.

This approach helped maintain the coherence and relevance of our analysis.

Find Communities of Restaurant categories

Nodes = restaurant categories (Italian, Vegan, Fast Food,...).

Edges = co-occurrence of two categories within the same restaurant.



Girvan-Newman Algorithm

1. Calculate edge betweenness for each edge
2. Remove the edge with the highest betweenness
3. Recompute edge betweenness
4. Repeat until the graph breaks down into multiple components or communities.

Edge weights = color intensity

The algorithm learned in the lecture didn't yield interesting communities, so after this, we tried a different one (next slide).

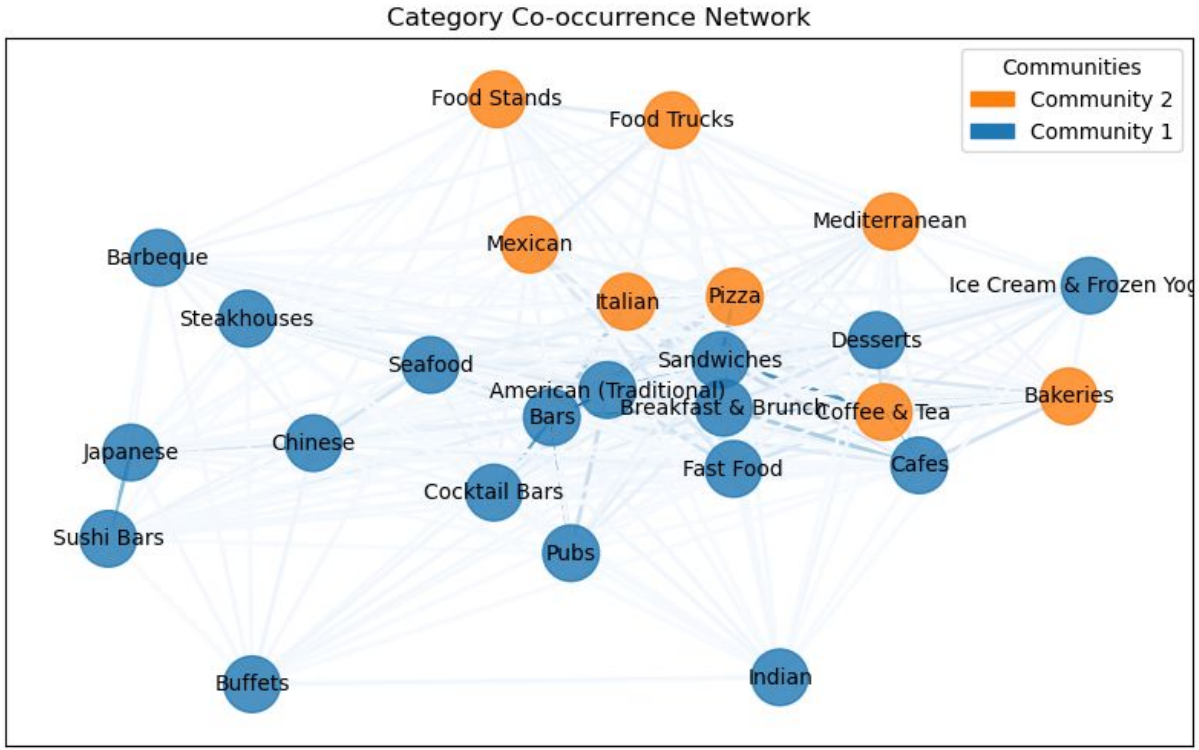
Find Communities of Restaurant categories

Clauset-Newman-Moore greedy modularity maximization Algorithm

Repeatedly join the pair of communities that lead to the largest modularity until no further increase in modularity is possible

Modularity:
Measures the density of links inside communities compared to links between communities.

Higher modularity = stronger community



Accessing influence

		Community 1	Community 2
Mean degree centrality	Average number of connections a node has to other nodes	**0.88**	**0.84**
Sum of degree centrality	Total number of connections across all nodes	15.80	6.68
Size	Total number of nodes in the community	**18**	**8**
# Edges	The sum of all the weights of the connections in the network, where each edge might have a different value	142	28
Total weight of edges	Sum of all the weights of the connections in the network	3156	548
Average weight of edges	Average weight of all connections, calculated by dividing the total weight by the number of edges	**22**	**19**
# Edges / Size	Ratio of the number of edges to the number of nodes, showing the average number of connections per node	**7.88**	**3.5**

Accessing influence

Conclusions for Community 1

- These categories are better and stronger connected to other categories
- Nodes are more influential in terms of direct connections within the overall network
- Multi-Cuisine and more **diverse** options
- **Potential for cross-promotions**

Community 2

- More **specialized**
- Maybe fast and convenient

	Community 1	Community 2
Mean degree centrality	**0.88**	**0.84**
Sum of degree centrality	15.80	6.68
Size	**18**	**8**
# Edges	142	28
Total weight of edges	3156	548
Average weight of edges	**22**	**19**
# Edges / Size	**7.88**	**3.5**

Additional observation:
If we were to launch a marketing campaign, we would target Community 1, as the categories within this community are more interconnected and likely belong to the same group of services. Offering promotions at one restaurant in this community could more easily encourage people to visit others. In contrast, the restaurants in Community 2 appear less connected, making cross-promotion less effective.

Measuring homophily

Community A (Size: 18) P=18/26

Community B (Size: 8) Q=8/26

Observed fraction of intra-community edges: 0.7539

2*P*Q = 0.4260
2*P*Q = 0.4260 < 0.7539

No evidence for homophily -> Good separation

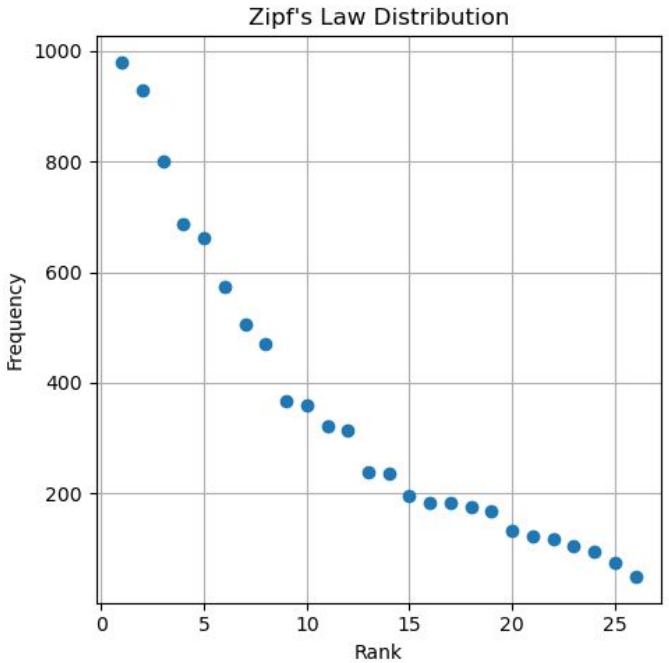
Interpretation of the Homophily Index

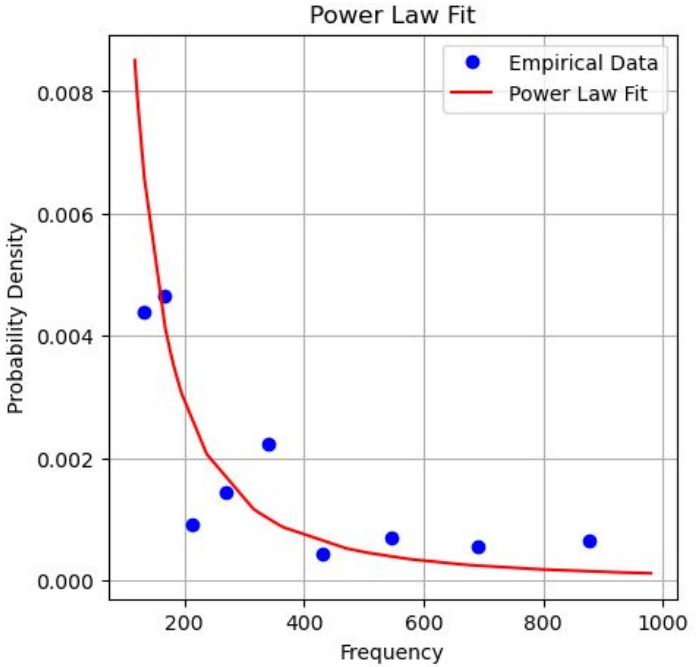
0 < H < 1

H = 1: This indicates complete homophily, meaning all connections are within the same community.

H = 0: This indicates complete heterophily, meaning **all connections are between different communities**.

Zipf’s curve and Power law





Only a few categories are very popular (high frequency of restaurant categories on the left, and higher number of connections on the right), while most categories are underrepresented

Find Communities of Users

Steps:

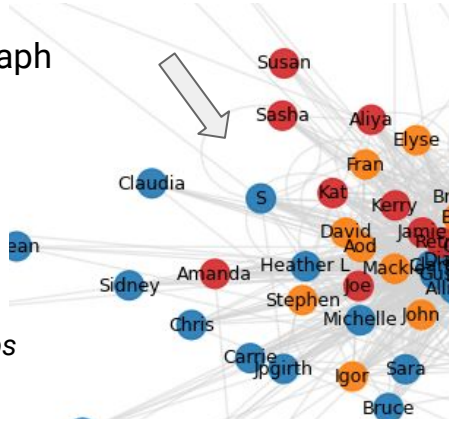
- Girvan-Newman Algorithm
  - Not effective

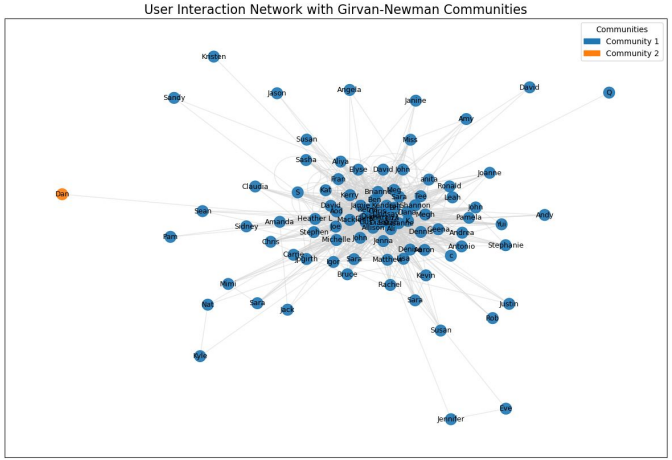
- Clauset-Newman-Moore greedy modularity maximization
  - Circularity removal
  - Conversion to a Directed Graph

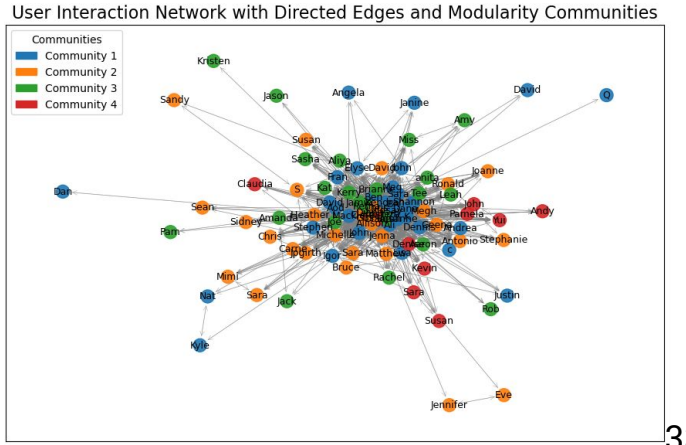
Nodes = Users

Edges = Friendship between users

Sample of 100 -> hard to visualize real friendships







Accessing influence

Conclusions for communities 1 and 2:

- Larger
- Generally weaker individual friendships

Communities 3 and 4:

- Smaller
- Stronger friendships

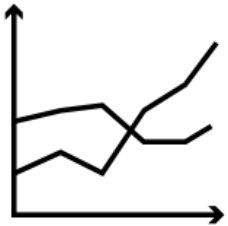
Additional observation: If we were to select a user from a specific community to promote a product, we would choose someone from Community 3 or 4. Although these communities have fewer people, they are more closely connected (higher average weight, higher mean centrality). This suggests that the connections within these communities are stronger, increasing the likelihood of the product spreading effectively.

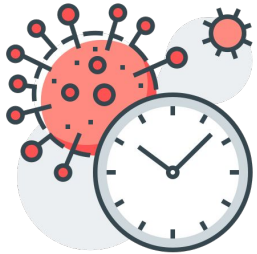
Most influential User	*Degree Centrality*	*Betweenness Centrality*	*Closeness Centrality*
Kristen (C.4)	1.4947	0.0677	0.7983

	Community 1	Community 2	Community 3	Community 4
Mean degree centrality	*0.23*	*0.27*	*0.25*	*0.25*
Sum of degree centrality	*7.39*	*8.51*	*4.50*	*3.70*
Size	**32**	**31**	**18**	**15**
# Edges	155	192	64	49
Total weight of edges	603	542	322	558
Average weight of edges	3.89	2.82	5.03	**11.39**
# Edges / Size	4.84	6.20	3.56	3.26

Time Series forecast - Introduction

1. Analysis of the monthly sum of reviews for restaurants in Philadelphia.
2. Compare the users behaviors for two different communities obtained on the previous analysis.
3. It was possible to predict users behaviors during covid time?
4. Are the typical forecast models able to predict the number of reviews?



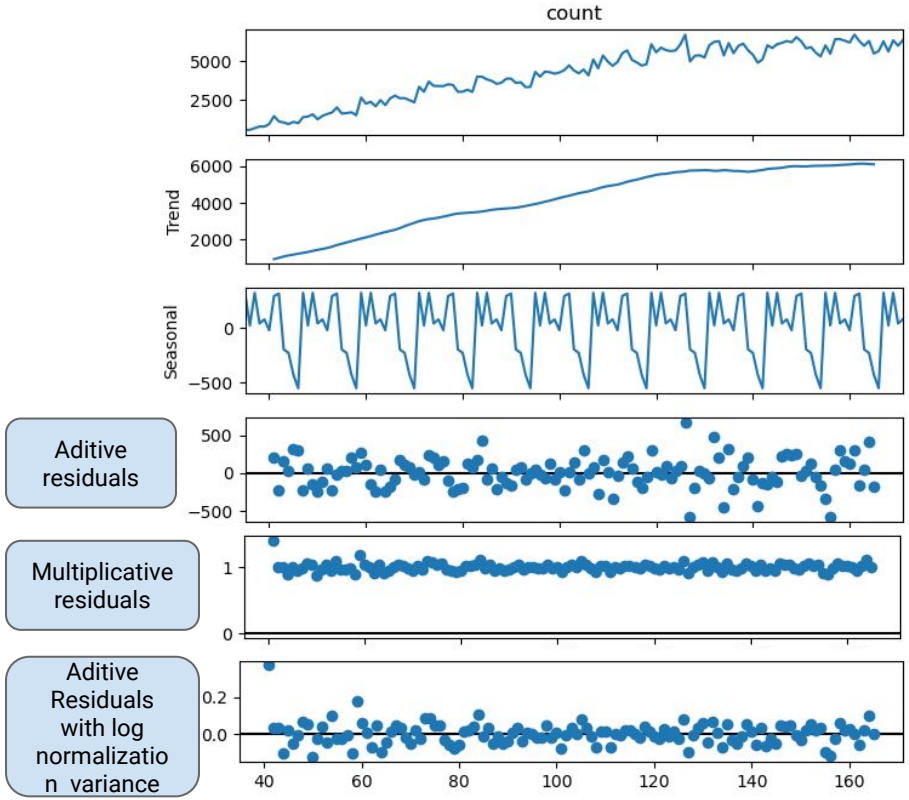


Time Series Forecast - Part 1

We excluded the first year (2006) and the most recent years, as they were impacted by COVID-19. We then performed a residual analysis using both additive and multiplicative seasonal decomposition methods, observing an improvement in the residuals.

However, the residuals still indicated increasing variance over time. To address this, we applied a logarithmic transformation to the signal, which helped normalize the variance. While this transformation improved the residuals, it made the seasonality slightly less pronounced, as shown in the next slide.

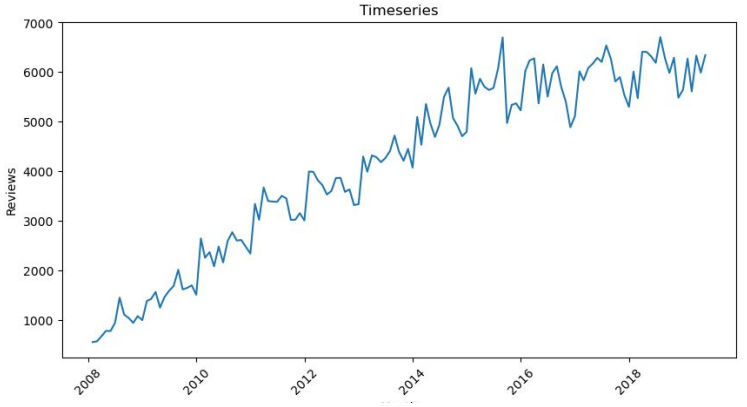
Using AutoARIMA, the recommended model parameters (p, d, q) changed from (1, 1, 0) to (0, 1, 1) using the logarithmic scale.

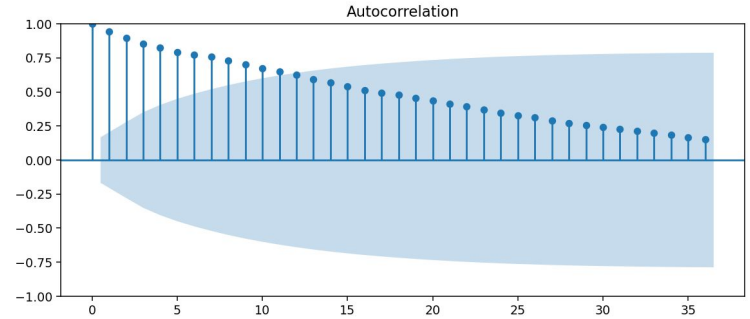


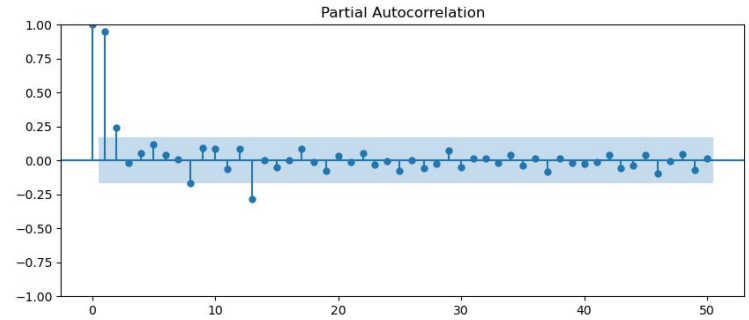
Time Series Analysis - Part 2

To identify seasonality, we examined the partial autocorrelation function and observed a spike at the 12-month mark, suggesting a yearly seasonal pattern. This was further confirmed by counting the repetition of patterns in the seasonality plot, as seen in the previous slide.

Additionally, the autocorrelation analysis revealed that each month is strongly positively correlated with the two preceding months, with this correlation gradually decreasing over the next 36 months (or 3 years).





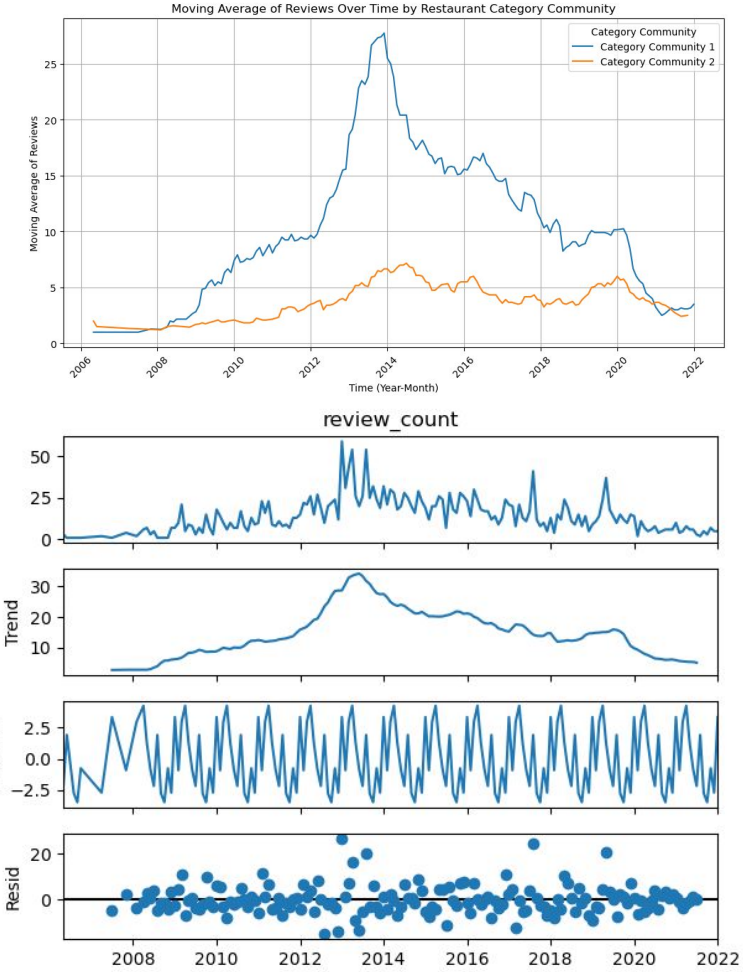


Time Series Analysis - Part 3

When analyzing the number of reviews within sampled user communities, we observe an initial upward **trend**, followed by a decline. One possible explanation is the impact of a marketing campaign, which drives a rapid increase in reviews that later drops off as incentives wane. However, this results in an overall increase in reviews over time. Another factor could be the rise of Instagram in 2013, which led to a shift in platform usage.

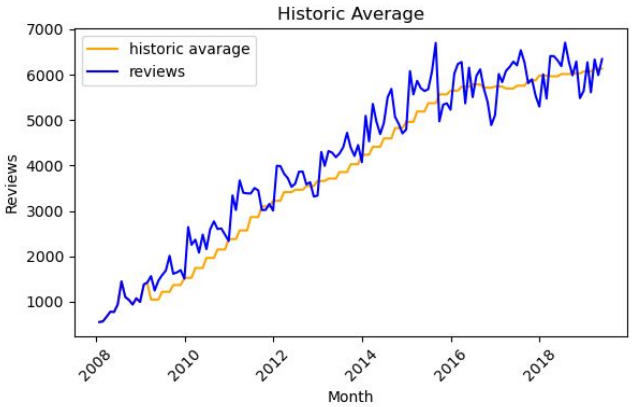
Additionally, we notice strong yearly **seasonality**, though it is less clear at the beginning of the data, as normal.

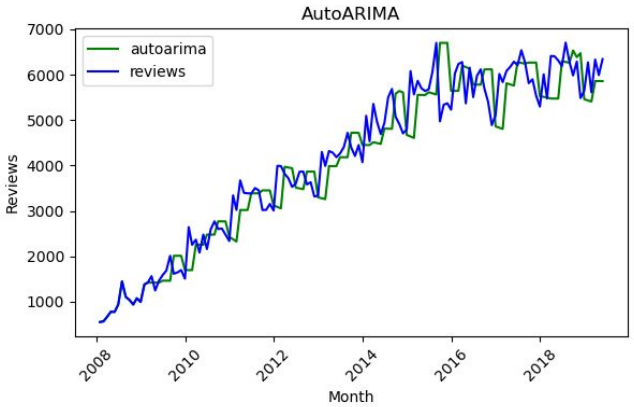
Regarding the **residuals**, they highlight an abnormal pattern in 2013 without an obvious explanation, as discussed, and a noticeable final increase, likely due to the impact of COVID-19.

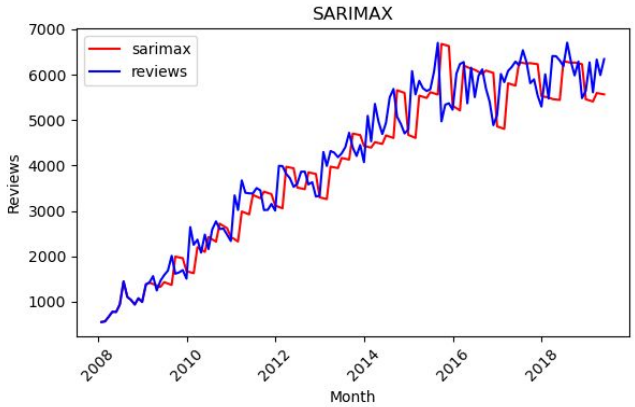


Forecast using sliding window

- Train in 12 months (1 year)
- Predict with an horizon of 3 months (1 quarter)

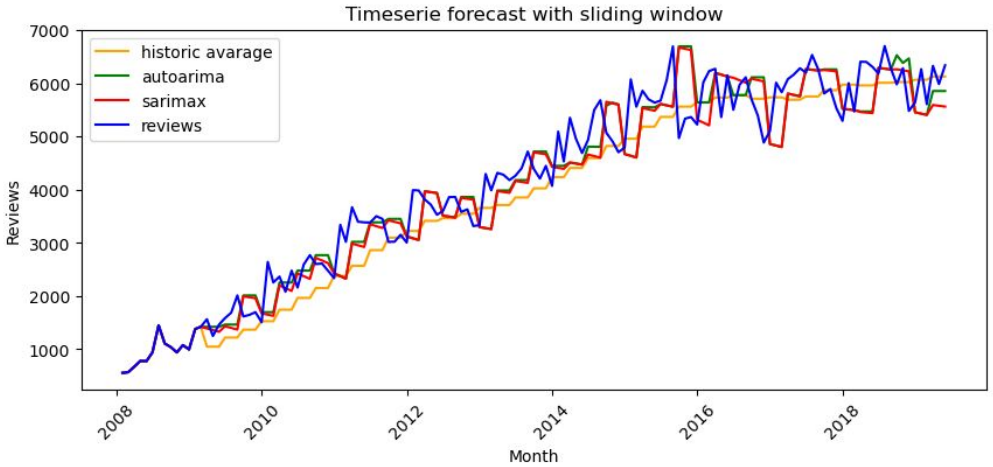






Forecast using sliding window - metrics

Model	RMSE	MAE	MAPE	AIC	BIC
Hist avarage	491.6	394.79	10.17	–	–
SARIMA	548.2	407.6	9.5	2869	2875
AutoARIMA	534.8	395.3	9.3	3491	3498



- Information Criterion (IC): as low as possible. Differences larger than 2 can be considered significant.
- Although error metrics show that AutoArima is better, information metrics that take into account the complexity of the model show that Sarima is better, as expected.

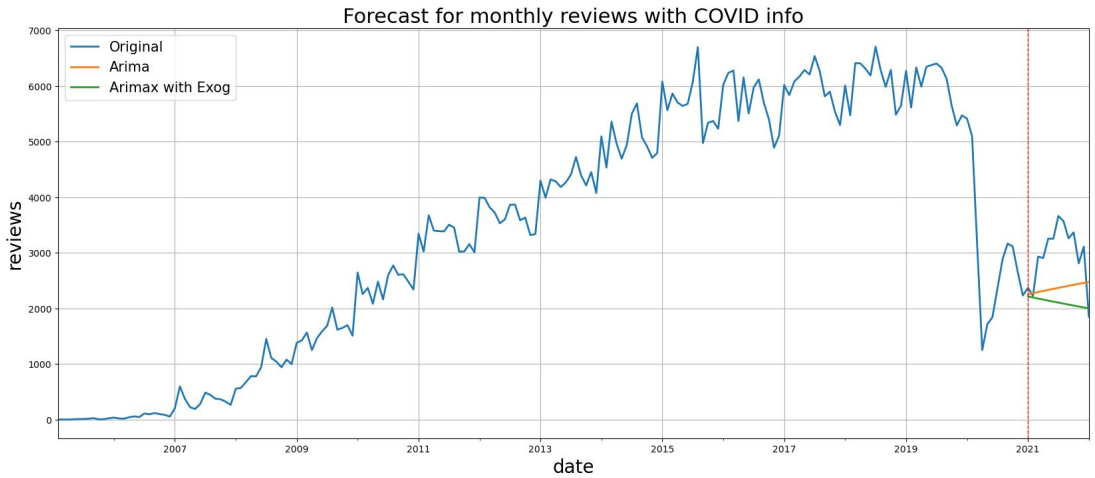
- *Additional observation: More tests were made for SARIMA and AutoARIMA using different window ranges for both train and test, AutoARIMA proves always being the better one as unexpected.*

Time Series + Covid

City of Philadelphia

collection_date	count	test_result
11/03/2020 04:00	6	positive
13/03/2020 04:00	12	positive
14/03/2020 04:00	13	positive
15/03/2020 04:00	15	positive
16/03/2020 04:00	12	negative
16/03/2020 04:00	34	positive
17/03/2020 04:00	15	negative
...		

ARIMA	RMSE	MAE	MAPE
Without exog	788.4	702.8	22.8
With exog	1010.7	884.9	27.6



Using data from the Philadelphia Website, we incorporated the number of positive cases per month as an exogenous variable in the ARIMAX model to compare it with a standard ARIMA.

- From 2020 to 2021, the model utilizes the actual monthly case totals.
- For the period after 2021 (the test set), the exogenous variable is approximated by using the average number of positive cases from the preceding months, simulating a potential forecast for those future dates.

Although the error increases, we consider this an improvement because the model begins to recognize that, based on the COVID trend, the number of reviews will decrease (green line) rather than increase (orange line).

Conclusion

- **Recommender System**: SVD outperformed other baseline methods for accuracy. Our User-based hybrid approach with topic modelling was the only better alternative. In isolation, LSA was faster but less accurate.
- **Communities Analysis**: Restaurant categories formed two distinct communities while Users formed 4 communities, with different characteristics. Larger user communities had weaker ties, while smaller groups had stronger connections.
- **Time Series Forecasting**: Clear seasonality was observed, with COVID-19 causing disruptions. AutoARIMA was effective, but simpler models often performed comparably.
- **Future improvements:**
    - Explore the fact that some reviews are positive and others are negative. It would be interesting to have a vector representation of what a user likes and dislikes specifically. Likewise, to have a vector representation of what a restaurant is good or bad at.
    - Test other text pre-processing techniques as n-grams and regular expressions.
    - Deployment using geographic distances to the restaurant as a weight for the predicted ratings.