A Comparison of Recommendation Systems

By: Daria Yip, 500721106

Ryerson University

Introduction

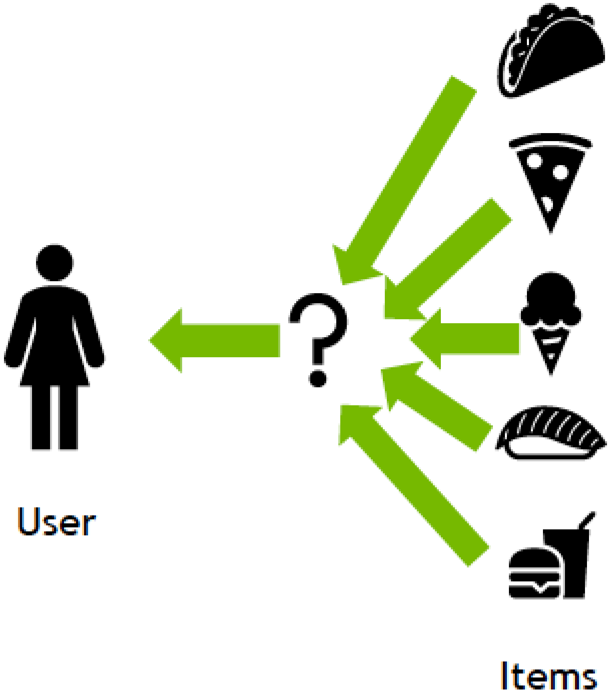
Introduction to Recommendation Systems

- An information filtering system that provides suggestions to users based on supporting data such as ratings, reviews, location, likes, and more
- **Content-Based Filtering Systems** generate recommendations based on similar items. It provides suggestions based on an item’s characteristics
  - Common techniques include distance based approaches (cosine similarity, Euclidean distance, Jaccard similarity) and classification methods (Bayesian classifiers, decision tree models)
- **Collaborative Filtering Systems** generate recommendations based on similar users. It provides suggestions based on users’ and items’ characteristics
  - Techniques include Matrix Factorization, user-based collaborative filtering (UBCF), item-based collaborative filtering (IBCF)

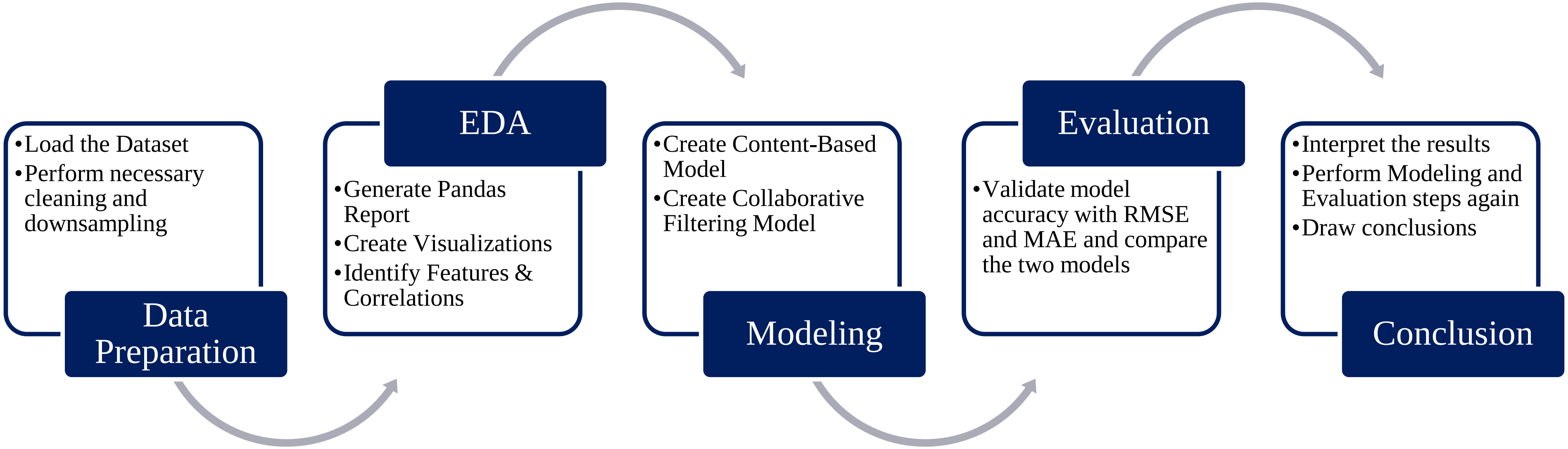
Research Questions

- “For restaurant recommendations, do content-based filtering models or collaborative filtering models provide better suggestions?”

- “What level of accuracy/performance can I achieve given these data points?”



Approach



Data Preparation

- Load the Dataset
- Perform necessary cleaning and downsampling

EDA

- Generate Pandas Report
- Create Visualizations
- Identify Features & Correlations

Modeling

- Create Content-Based Model
- Create Collaborative Filtering Model

Evaluation

- Validate model accuracy with RMSE and MAE and compare the two models

Conclusion

- Interpret the results
- Perform Modeling and Evaluation steps again
- Draw conclusions

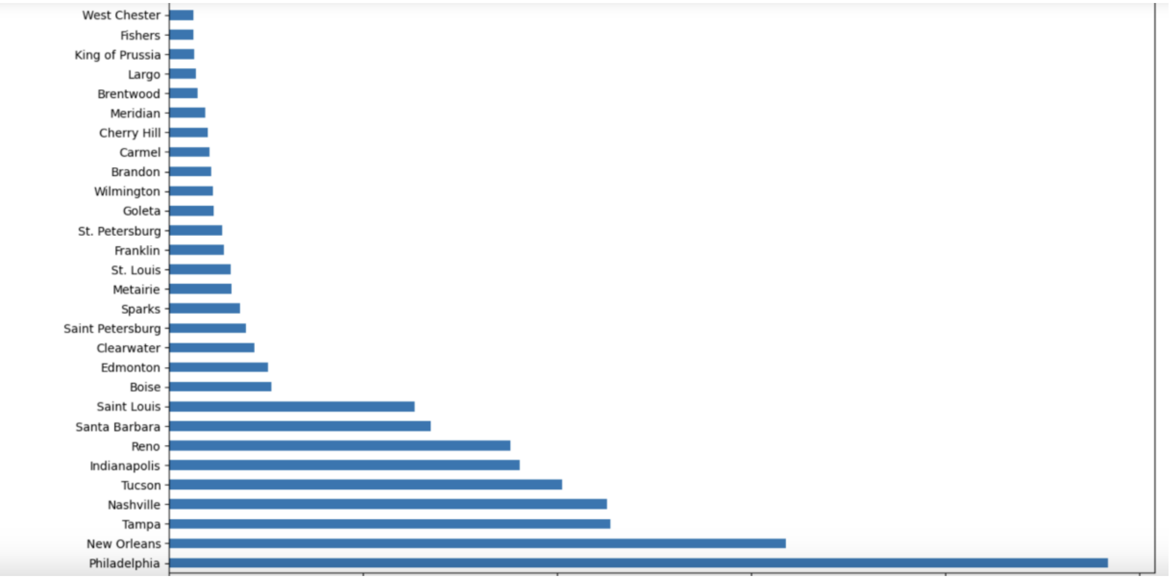
Exploratory Data Analysis

EDA: Yelp Dataset - Overview

- Files used were last modified in January 2022
- Contains data about businesses, users, reviews, tips (short reviews), and check-ins
- Yelp Dataset is made up of 5 files:
  - **Business.json** – Information regarding the businesses – **14 variables** (location, business_ID, name, categories etc.)
  - **Tip.json** – Information regarding tips written by users on businesses – **5 variables** (compliment_count, user_ID, business_ID etc.)
  - **Check-in.json** – Information regarding check-ins into a business – **2 variables** (business_ID, date)
  - **User.json** – Information regarding the users – **22 variables** (User_ID, name, review_count, friends, etc.)
  - **Review.json** – Information regarding reviews, including full review text – **9 variables** (text, user_ID, business_ID, etc.)
- All files were provided in JSON format and are available for public download
- At the time of download, the dataset included 6,990,280 reviews, 150,346 businesses, 908,915 tips, and 11 metropolitan areas to make up 5GB of data

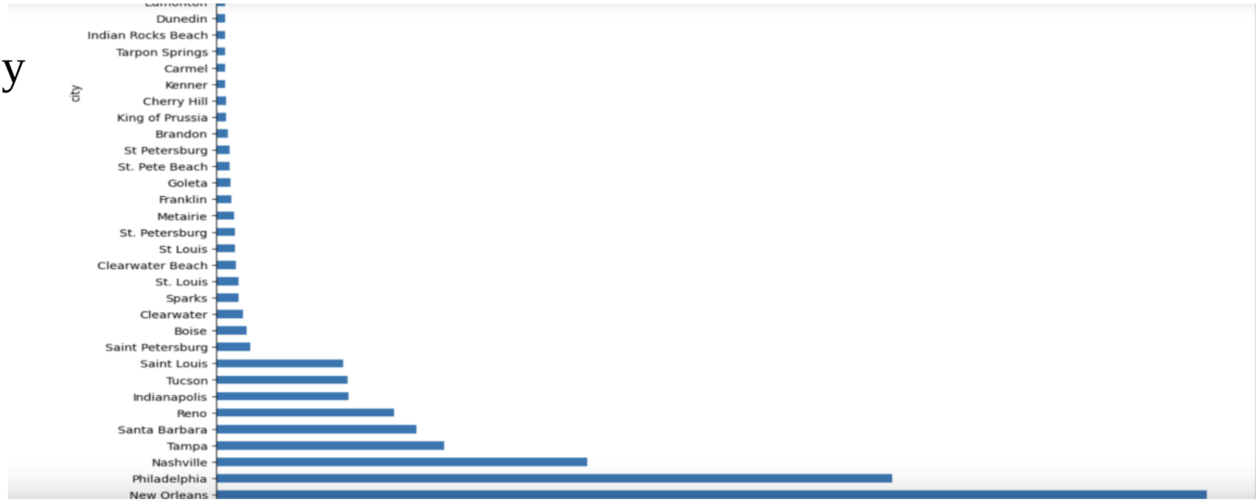
Ryerson University

EDA: Yelp Dataset – Top 50

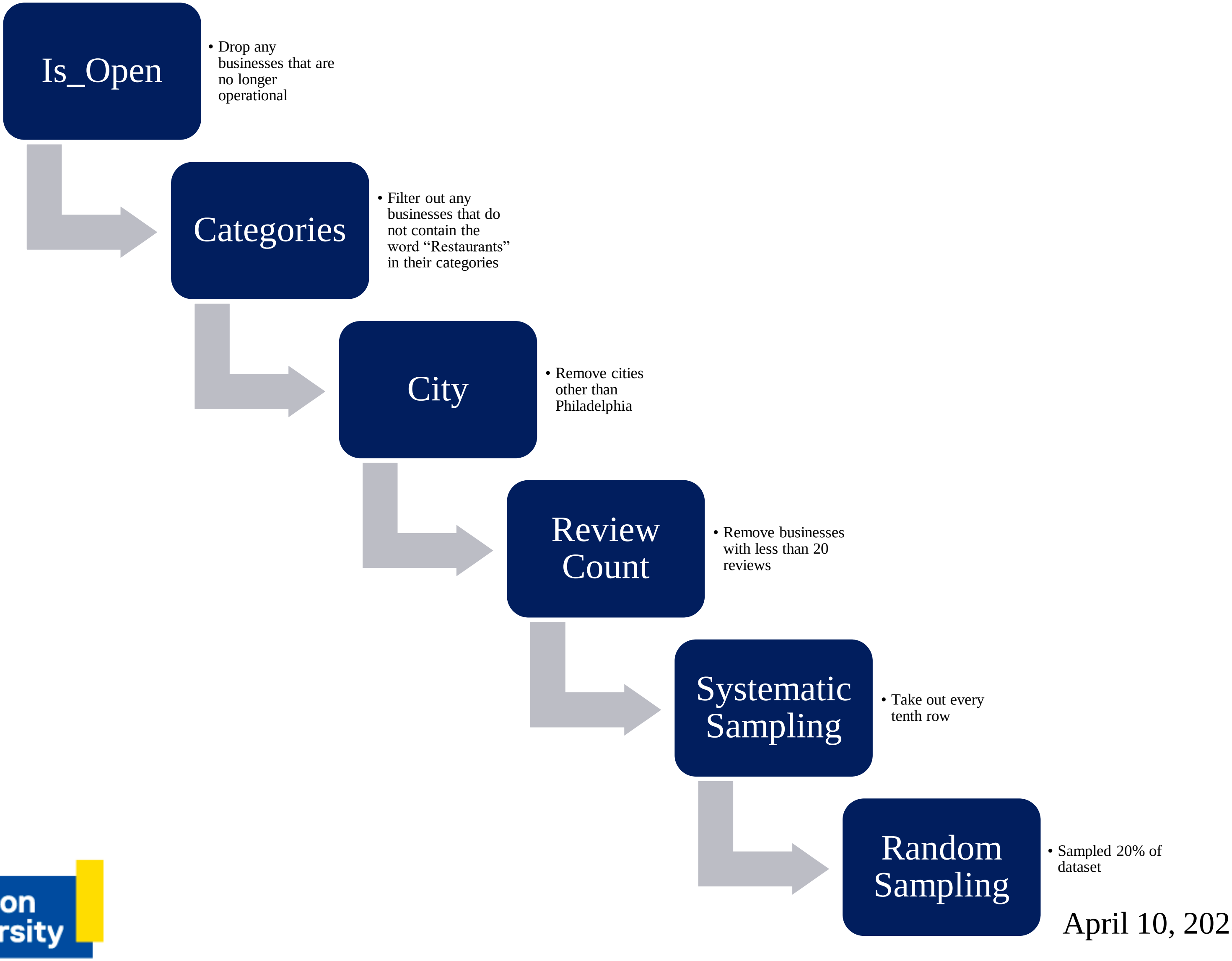


Left: Snippet of Top 50 Cities by Businesses

Right: Snippet of Top 50 Cities by Reviews



EDA: Yelp Dataset – Downsampling



Is_Open

- Drop any businesses that are no longer operational

Categories

- Filter out any businesses that do not contain the word “Restaurants” in their categories

City

- Remove cities other than Philadelphia

Review Count

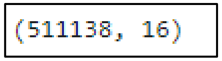
- Remove businesses with less than 20 reviews

Systematic Sampling

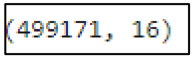
- Take out every tenth row

Random Sampling

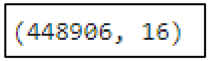
- Sampled 20% of dataset



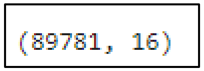
```
(511138, 16)
```



```
(499171, 16)
```

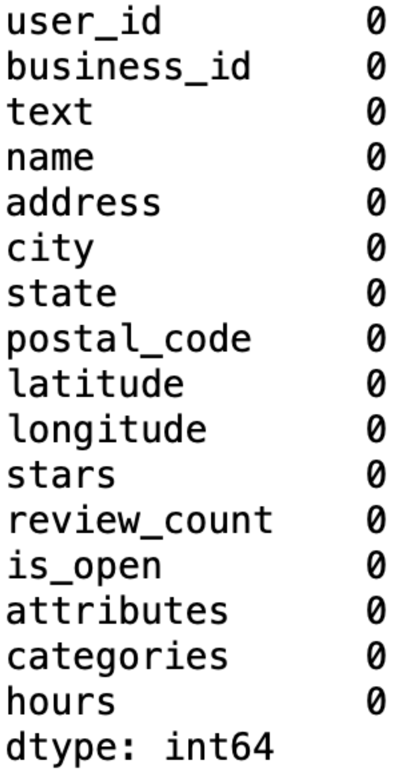


```
(448906, 16)
```



```
(89781, 16)
```

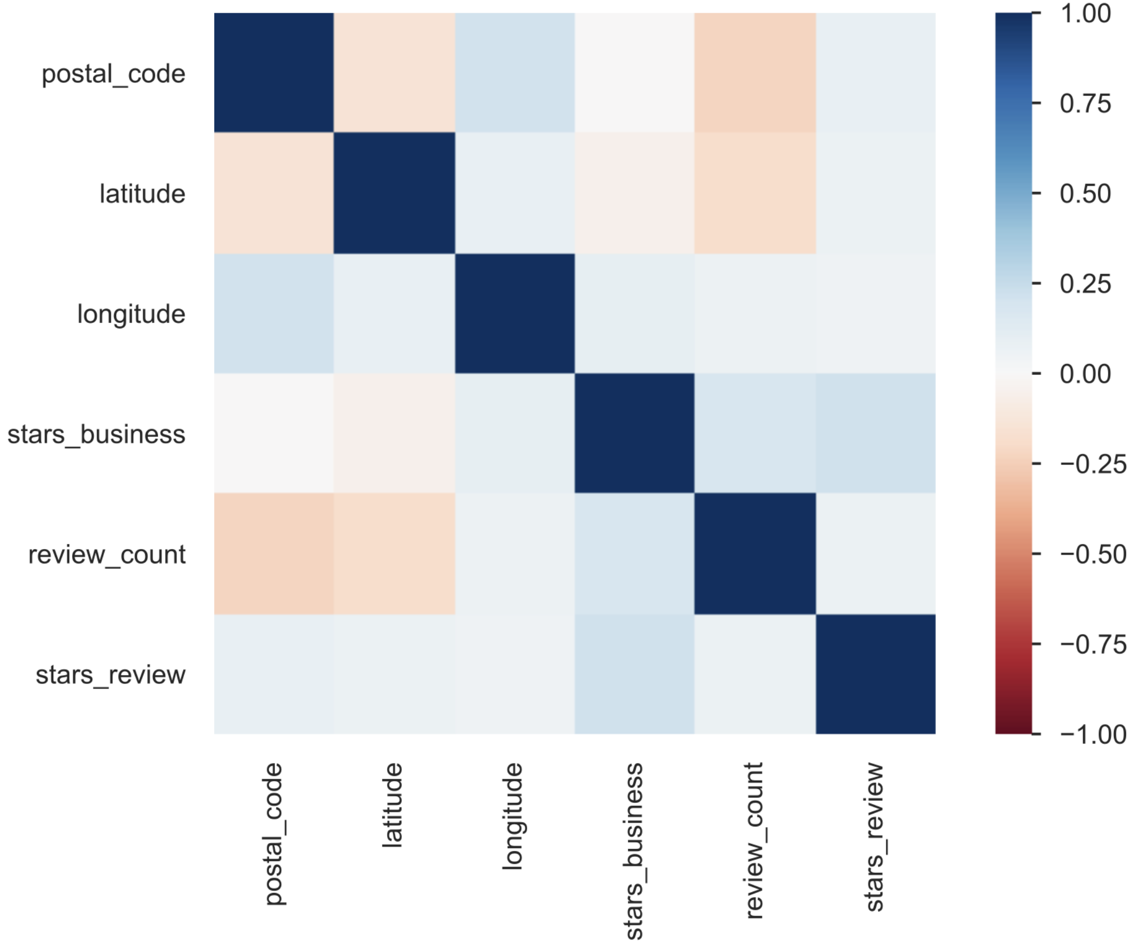
EDA: Yelp Dataset – Missing Data



```
user_id         0
business_id     0
text            0
name            0
address         0
city            0
state           0
postal_code     0
latitude        0
longitude       0
stars           0
review_count    0
is_open         0
attributes      0
categories      0
hours           0
dtype: int64
```

Left: Missing Data from the “businesseswithreviews” dataset

EDA: Yelp Dataset - Heatmap



Ryerson University

Modeling and Evaluation

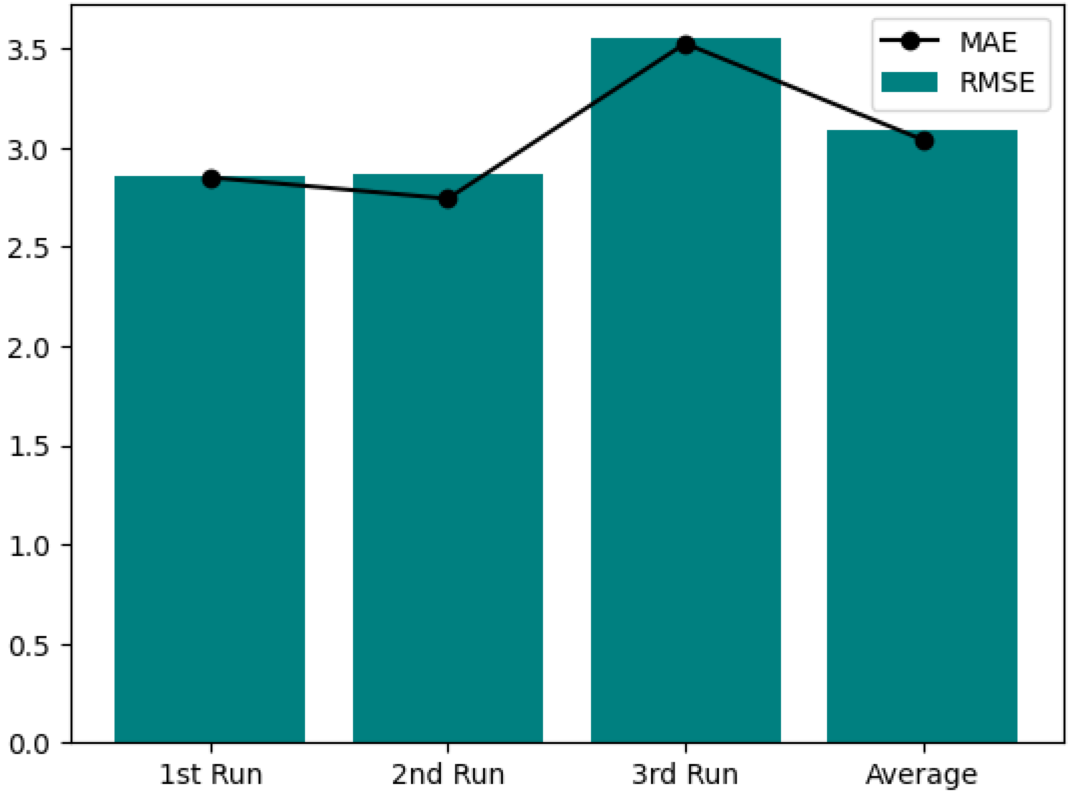
Ryerson University

Content-Based Model

- Baseline Model
  - Created using **TF-IDF** methodology. Goal of the model was to **generate recommendations based on similar businesses**
    - Similar businesses are generated using a similarity measure such as Cosine Similarity or Jaccard Similarity or Euclidean Distance on a “bag of words” related to the business
    - Based on the idea that if a word is important, it will appear frequently throughout the document – in this case, a review
  - The TfidfVectorizer from the Sklearn library was used
    - Data was preprocessed by removing stop words such as “is”, “a”, “the”, and “are”
    - Similarity measure used was cosine similarity metric, which was generated based on restaurant-to-restaurant similarity to generate scores of which restaurants were most similar (the higher the similarity score, the more similar)
  - Recommendations were then generated through a function that given a restaurant name, would generate the Top 10 suggestions for restaurants that were most similar

Run	RMSE	MAE
1	2.853944638566067	2.85
2	2.869544513382867	2.742857142857143
3	3.55	3.5250000000000004

Content-Based Model



Content-Based Model

- Bi-Gram Model
  - To be more similar to the way humans process language, bi-gram analysis was attempted, but it did not yield more accurate results (ie. Did not have a lower RMSE/MAE score)
- Categories Model
  - When the baseline model was run, it performed the TF-IDF analysis on the review text to try and find keywords that would represent each business and then group similar businesses based on these keywords
  - As it did not have high similarity scores (ranged in one example from 0.175711 to 0.373838), the analysis subject was changed from review text to the categories variable to see if it would make a material difference
  - While the similarity scores went up, the RMSE and MAE also did (were higher than baseline model)

Model	RMSE	MAE
Baseline TF-IDF (3 Runs Average)	3.0911630506496	3.0392857142857
Bi-Gram Model	4.06201920231798	4.0
Categories Model	3.643674469044076 8	3.381818181818181823

Collaborative Filtering Model

- Baseline Model
  - Built using the Surprise library
  - One of the major differences from the TF-IDF model's inputs was it used "ratings" instead of review "text" or "categories" (quantitative vs. qualitative) as the matrix factorization model required numerical inputs
    - Using Surprise's Reader function, outputs were normalized to the 1-5 scale that ratings have
  - The baseline model used SVD, or Singular Value Decomposition and was trained with the df_train dataset and tested using the df_test dataset (80/20 split)
    - Recommendations were given on the df_test dataset

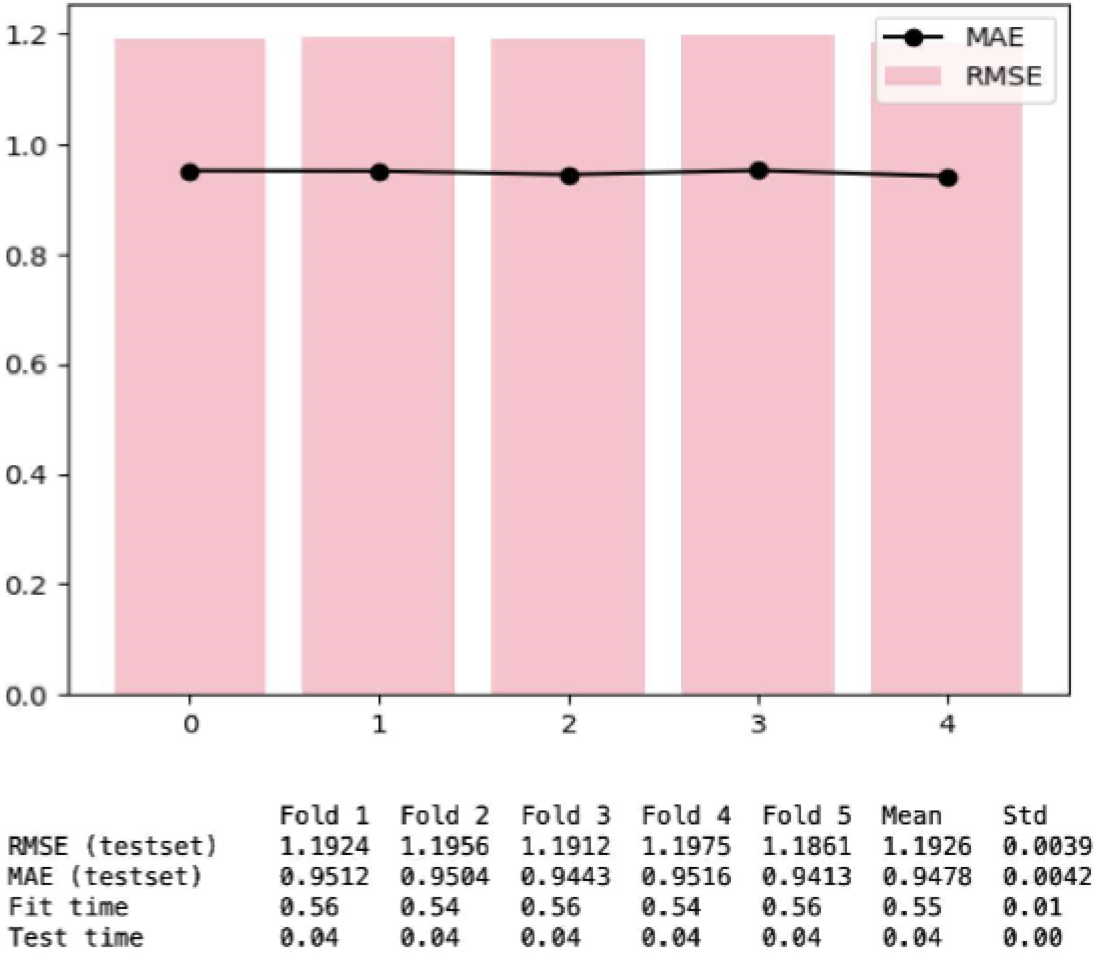
Run	RMSE	MAE
1	1.1845	0.9398
2	1.1976	0.9509
3	1.1976	0.9519

Collaborative Filtering Model

- ALS Model
  - Instead of using SVD, we used ALS or Alternating Least Squares
  - ALS differs slightly from SVD in that SVD decomposes the user-restaurant matrix into a low-rank approximation and then finds the latent factors and makes a lower rank matrix that captures the most important matrix
    - SVD is computationally expensive for large datasets
    - Also a more exact factorization technique
  - ALS can be seen as an optimization algorithm that alternates between optimizing the user and restaurant factors to minimize prediction error
    - It also factorizes the matrix, but does not require the entire matrix to be committed to memory → less storage required
    - Faster than SVD as a result
    - Can be seen as less accurate than SVD for sparse datasets as it essentially uses a "trial and error" method rather than figuring out the latent factors like SVD does

Model	RMSE	MAE
SVD (Baseline – 3 Run Average)	1.1932333333333	0.9475333333333
ALS (5 Fold Average)	1.1926	0.9478

Collaborative Filtering Model



```
                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    1.1924  1.1956  1.1912  1.1975  1.1861  1.1926  0.0039
MAE (testset)     0.9512  0.9504  0.9443  0.9516  0.9413  0.9478  0.0042
Fit time          0.56    0.54    0.56    0.54    0.56    0.55    0.01
Test time         0.04    0.04    0.04    0.04    0.04    0.04    0.00
```

Figure 33: Cross Validation of the ALS algorithm

Challenges

- **Size**
  - The Yelp Dataset includes 5GB of data, stored within 5 JSON files. This also meant that there are a significant amount of user data, business data, and reviews to sift through
  - Thus, in order to optimize the speed of the model's processing (given the limitations of my tiny (but powerful!) laptop, downsampling was performed as detailed
- **Comparison of Content-Based Model & Collaborative Filtering Model**
  - Typically, a content-based model like TF-IDF, we use precision & recall to measure the relevancy of restaurant recommendations returned
  - Since RMSE & MAE were used to measure the collaborative filtering model, to ensure comparability, we would need to measure both models on the same scale
  - Therefore, using RMSE & MAE as the common evaluation metrics, the TF-IDF model was tweaked to choose a random user from a user-item matrix (where user reviews are more than 5)
    - Then, the corresponding ratings were found and then predicted ratings were generated (weighted average of cosine similarity score * star rating for the highest rated restaurant based on the user's actual scoring)
    - Highest rated restaurant was chosen as we are looking to find the restaurants that would be also highly rated (in a recommender system)

Conclusion

Ryerson University

Conclusion

Research Question	Answer
“For restaurant recommendations, do content-based filtering models or collaborative filtering models provide better suggestions?”	As the collaborative filtering model had lower RMSE and MAE metrics (ALS algorithm + cross validation), **average RMSE across 5 folds was 1.192 and average MAE was 0.9469** vs. the content-based model (TF-IDF Baseline) with an **average RMSE across 3 runs of 3.0911630506496 and average MAE of 3.0392857142857**
“What level of accuracy/performance can I achieve given these data points?”	The best RMSE and MAE is presented above at 1.192 and 0.9469 respectively

Conclusion

- **Employed Techniques**
  - Major libraries were JSON, Pandas, Surprise, Numpy, Random, Matplotlib, and Sklearn
  - Pivot Tables were used, left joins, downsampling, systematic sampling, random sampling, cross-validation, TF-IDF for content-based filtering along with bi-gram analysis, correlation heatmap, cosine similarity matrices, and SVD and ALS for collaborative filtering
- **Limitations of Work**
  - Assumes that collaborative filtering models are best and wholly represented by the ALS and SVD techniques (subset of matrix factorization) and content-based filtering models are best and wholly represented by the TF-IDF technique (restaurant-to-restaurant comparison/IBCF)
    - Other Content-Based Filtering techniques include classification methods (Bayesian classifiers, decision tree models)
    - Other Collaborative Filtering techniques include UBCF (user-based CF)
    - Taking reviews as an accurate proxy for restaurant similarity – does not take into account fake reviews or the fact that reviews may have a negative bias (may influence the words used)
- **Next Steps**
  - Future testing can be performed using tri-gram analysis on the review text, incorporating tips.json along with the review text, and/or using Jaccard or Euclidean Distance instead of Cosine Similarity