Nakisa Abbasi

Personalized Food Recommendation System

Springboard Final Project
April 2022

Introduction and Problem

- Many types of food and cuisine available for users to order from different online applications.
- Hard for the users to pick and order their desired meal especially from a new cuisine that they have never tried before
- Most of the current recommending systems are mostly restaurants based
- We need a recommending system which is based on food items similarities

Data Source and Constrains

- Date used from Fooreco - an online ordering application
  - Orders.csv - 14K
  - Users.csv - 5K
  - Items.csv - 80K
- Current available ordered data is limited - few restaurants
- Users information is very limited - email and names
- Ratings and comments for foods are very limited

Current Recommendation Solutions

- Most popular item
  - Assumption: All the customers would like the most popular item
- Association model
  - Based on current order basket - recommends associated items
  - Example: Basket contains coffee - recommends coffee filters, coffee mug
- Content based filtering
  - Based on previous ordered items, watched movies - similar items
- Collaborative filtering
  - Base on collecting preferences or tasted based on other similar users
- Hybrid approche
  - Combination of two or more strategies

Our Approche

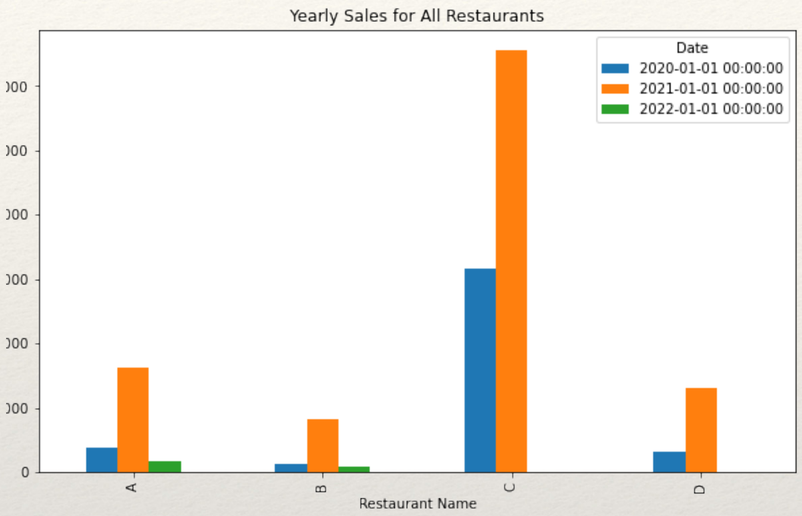
- Association Model
  - Network of Ordered Items
- Collaborative based Model
  - User similarity based
- Content based Model - Winner
  - Item similarity based

Restaurants Exploratory Analysis

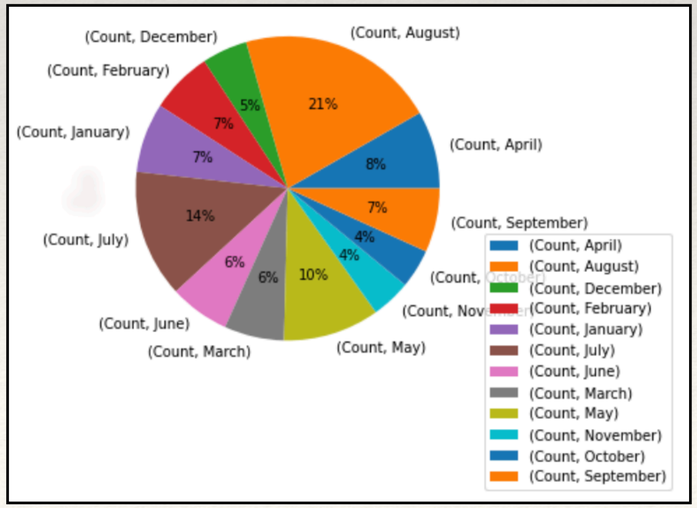
- Fooreco is a double sided application - Users and Restaurant based
  - One of the benefits of recommending system is to increase restaurant's sales.
  - Lets explore the current restaurants sales
    - Overall yearly comparison
    - Monthly sales
    - Daily and hourly sales

Continued

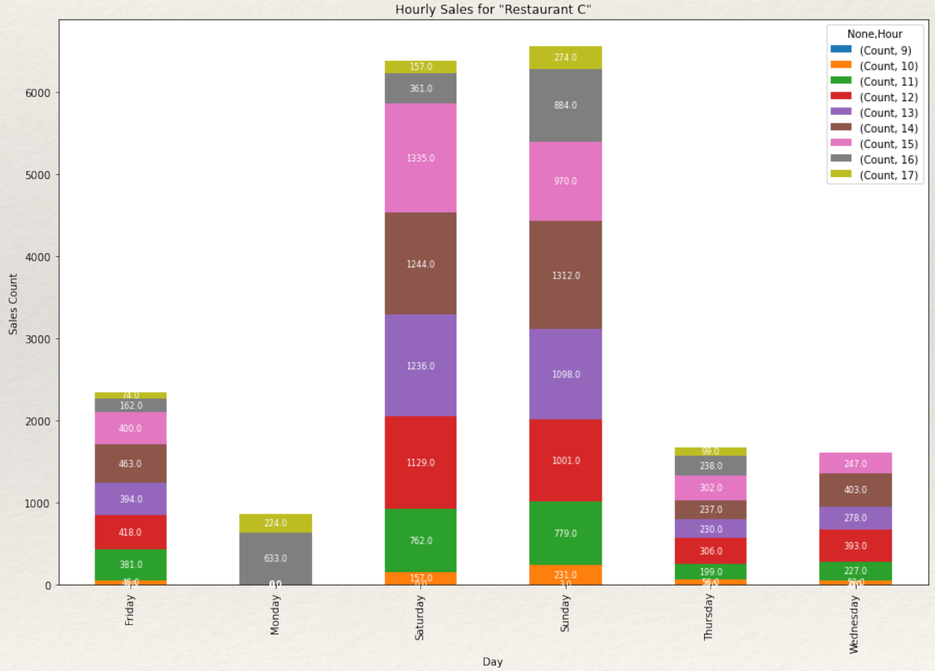
- Yearly



- Monthly



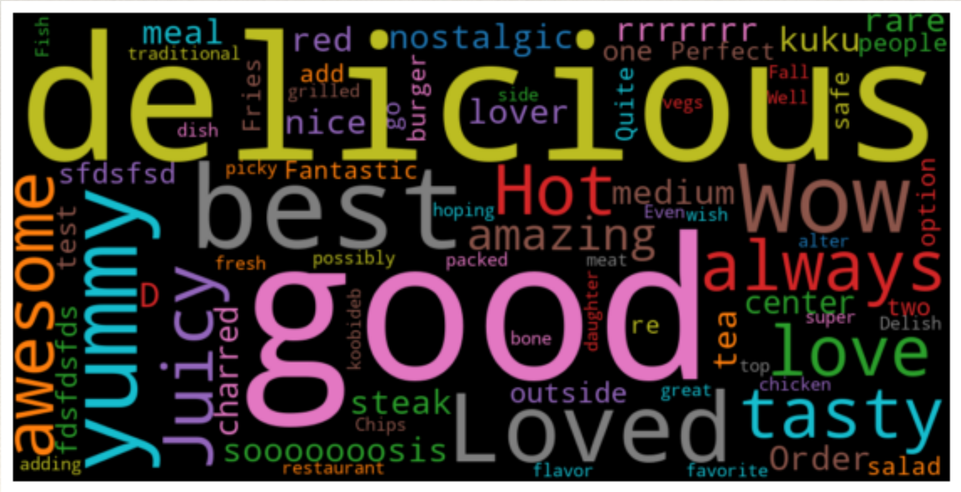
- Daily - Hourly



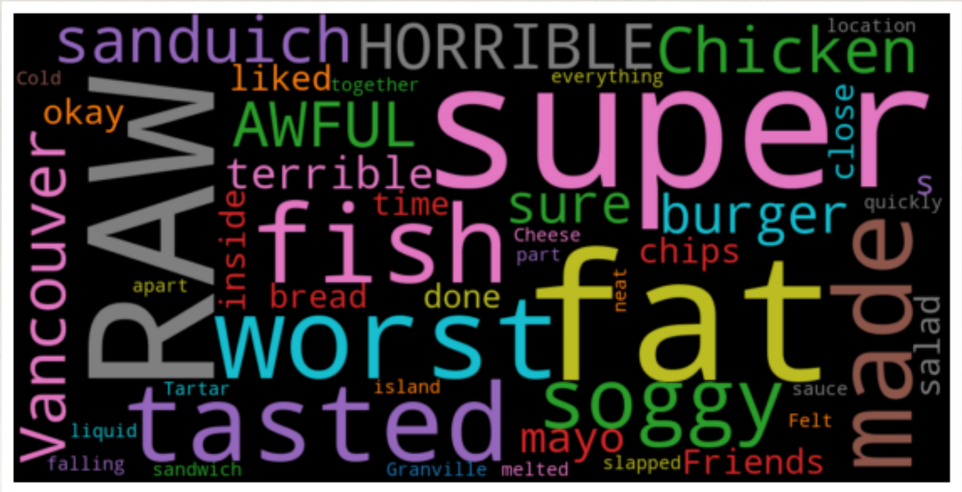
Feedback Sentimental Analysis

- Which words the customers used the most to describe their food
- After pre-processing and tokenizing we user WorldCloud package to visualize the words in the the feedback:

A. Positive Sentiment



B. Negative Sentiment



Continued

- A model to predict the the new feedback sentiment without reading it.
- Preprocess:
  - Clean ,Lemmatize and tokenize the text
  - CountVectorizer() and TF-IDF - convert text to vectors
- Model
  - Naive Bases Classifier - best for text analysis
  - Specifically MultinamialBV
- Metrics
  - Accuracy score
  - Confusion matrix
  - Classification reports - F1 score, recall and precisions

Sentiment Model Performance

- Yelp Data:
  - As mentioned before, as our data was very limited, for the modelings we used food reviews in Yelp database.

Models/ Metric	Accuracy	Precision Macro Average	Recall Macro Average	F1 score Macro Average
Countvectorize NB	88.37	83%	81%	82%
TFIDF NB	79.72%	90%	52%	48%

First Recommendation approche

- Association Model:
  - Nodes: Food Items
  - Edges: Foods ordered together
  - Weight: Number of times ordered together

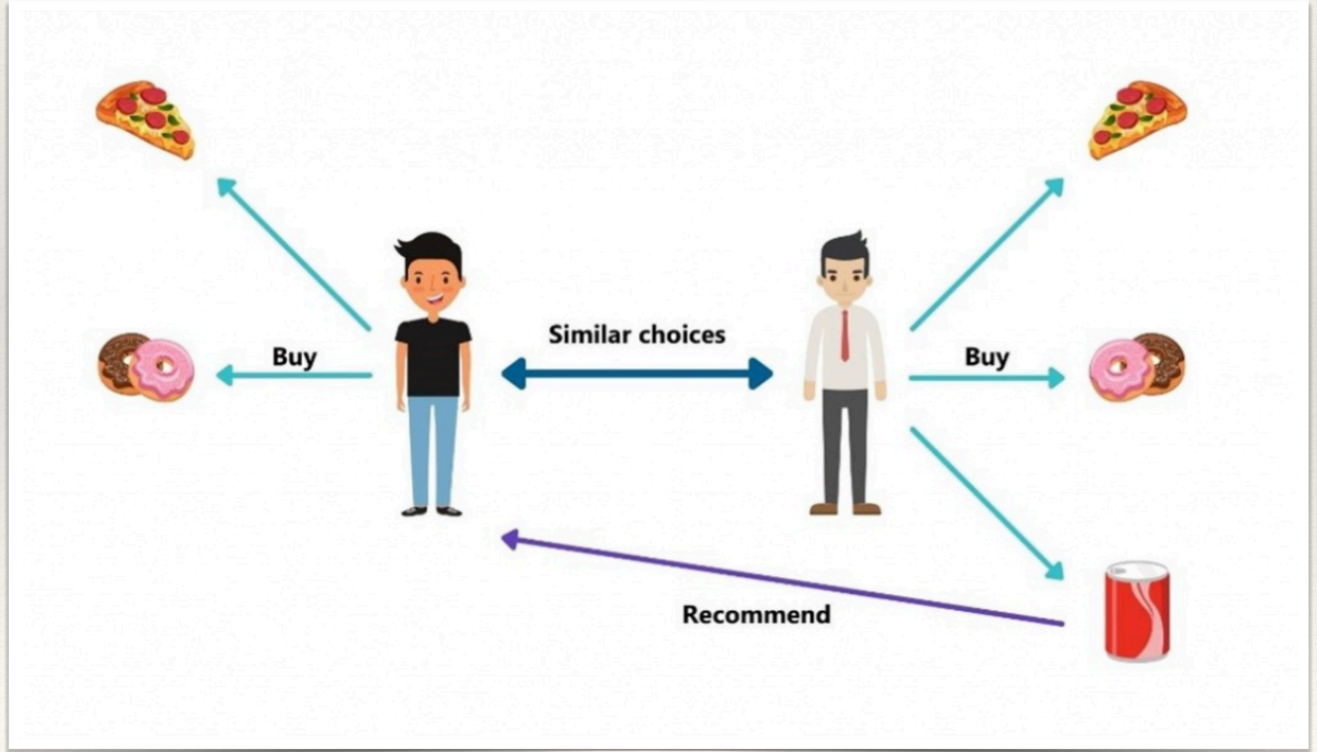
Example:

```
food = random.choice(items)

print('food: ',food)

recommend_top_addon(food)
```

```
food:  Popcorn Chicken
total found:  24
1 : HH Shoestring Fries ( 44 )
2 : Lager Hoi Polloi ( 22 )
3 : HH Crispy Chicken Sandwich ( 16 )
4 : HH Cheese Burger ( 15 )
5 : Gin & Tonic ( 12 )
6 : Pino Gris White 5oz ( 11 )
7 : Vodka Soda ( 8 )
8 : Root Beer ( 6 )
9 : Coca Cola ( 6 )
10 : Gin & Soda ( 6 )
11 : Rum & Coke ( 5 )
12 : Merlot Red 5oz ( 5 )
13 : Hammer Hot Chicken ( 5 )
14 : Sparkling Water ( 4 )
15 : Fish And Chips (G) ( 4 )
16 : Ginger Ale ( 4 )
17 : Clam Chowder ( 4 )
18 : Popina Bottled Water ( 3 )
19 : Shoestring Fries (G) ( 2 )
20 : East Coast Lobster Roll ( 2 )
21 : HH Falafel Sandwich ( 1 )
22 : Powell Street, Ode to Citra  (473 Ml) ( 1 )
23 : Hoi Polloi Lager ( 1 )
24 : Grass Fed BC Double Cheeseburger (8oz) ( 1 )
```

Collaboration Based Recommendation

- Assumption: If a user ordered a food, the system infers that the user is interested - a random rating from 3-5 added to the foods with no rating for the foods that user has already ordered.



Model details

- Package:
  - Surprise - A Python scikit for building and analyzing recommender systems
- Model
  - SVD - Singular Value Decomposition - factorization o a matrix in to 3
  - KNN with means - K Nearest Neighbours
- Metrics
  - RMSE

Performance Evaluation

- Fitting the data to the models and the results:

Models/ Metric	MAE	RMSE	Fitting Time(s)	Predicting Time(s)
SVD	0.70	0.84	0.037	0.003
KNN with mean	0.76	0.93	0.49	6.72e-05

Example:

- Using both model to predict a rating for a user and an item:
  - sample-itemId = 17550
  - sample-userId = 2036
- svd.predict(sample-itemId, sample-userId)
  - The predicted rating that User 2036 would give item 17550 is 5
- knn.predict(sample-itemId, sample-userId)
  - The predicted rating that User 2036 would give item 17550 is 4.15
- As seen in the performance table, SVD has a better prediction with lower RMSE.

- Actual ratings:

	userid	itemid	rating
2347	553	17550	5.0
4848	1004	17550	3.0
5264	1028	17550	5.0
6214	1262	17550	5.0
6587	1334	17550	3.0
10248	2036	17550	5.0
10317	2046	17550	3.0
11660	2399	17550	5.0
11793	1909	17550	5.0
17383	3901	17550	5.0

Content Based Recommendation

- This recommendation is based on items similarities.
- Tokenize:
  - Create bag of words of item name + item description
- Vectorize:
  - CountVectorizer()
- Similarity calculation:
  - Cosine Similarity

Procedure:

- This recommendation is based on the similar foods to the most ordered food by the user.
- content-based-recommender(sample_userid)
  - Backend finds the top ordered food
  - Finds the most similar food and return
    - The threshed for the similarity can be tuned(for example above %50)

Example:

- content-based-recommender(sample_userid)

```
Top order food is:  Fish And Chips (G)
Top order food description is:  cod, fries, coleslaw, tartar sauce
```

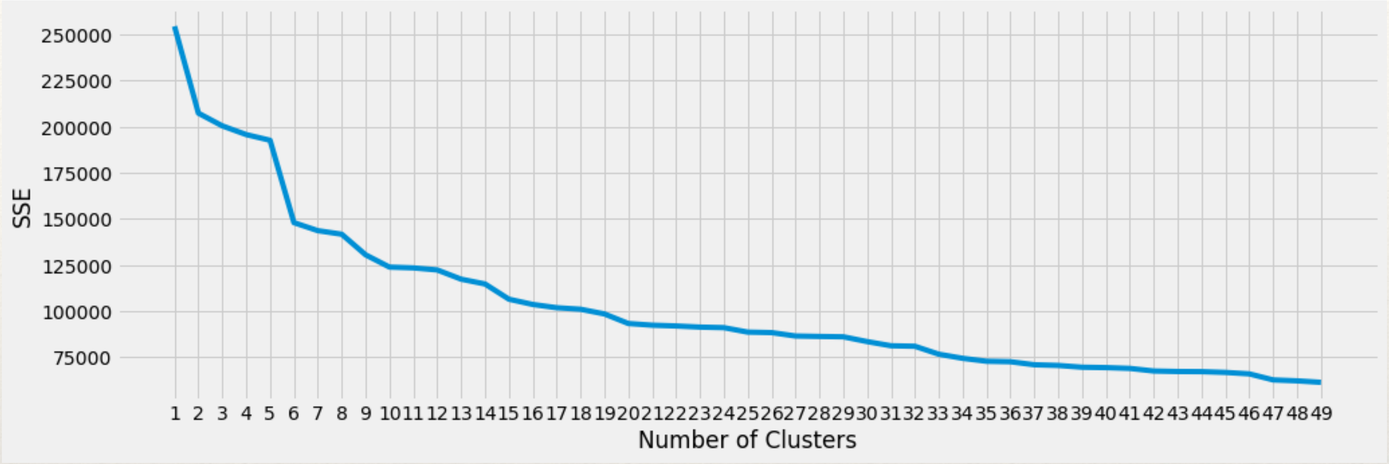
```
item:  Fish N' Chips
item:  Pacific Cod (2 Pcs) And Chips
item:  Cod And Chips (2 Pcs)
item:  Pacific Cod And Chips
item:  Cod & Chips
item:  4. Homestyle Fish And Chips With Coleslaw
item:  Tartar Sauce
item:  Fish Burger
item:  Fish (3 Pcs) And Chips Combo
item:  2 Piece Fish And Chips
item:  Fish (1 Pc) And Chips
item:  Fish And Chips
item:  Japanese-style Fish And Chips
item:  London-style Fish And Chips
```

Clustering

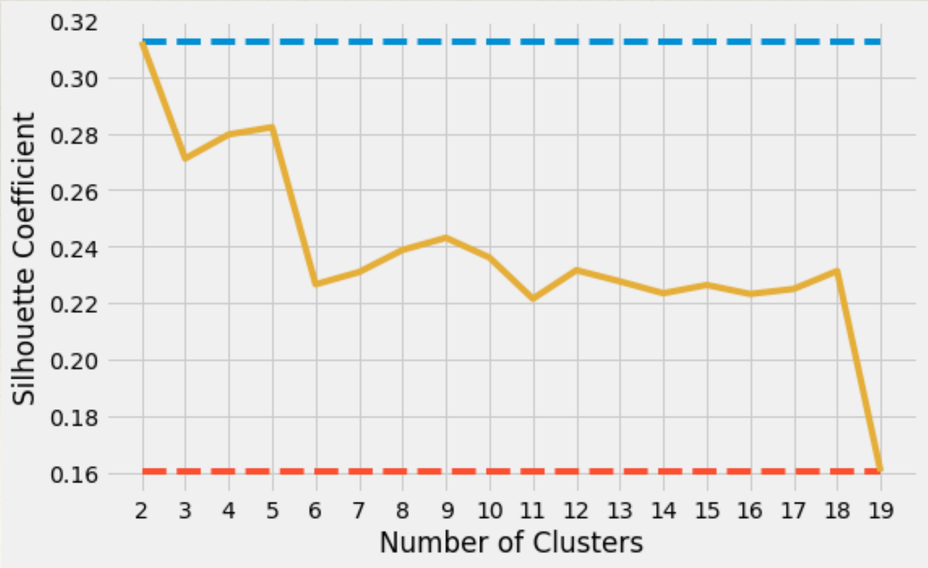
- Lets cluster the food based on the similarity and recommend the foods in the same clusters
- The procedure is same as before, before fitting the data to the Kmeans model, we need to reduce the dimension of input data
- Cosine similarity matrix has around 41K features
- Fitting the matrix to the PCA and decrease the dimension to 50 components

Model Metrics

- SSE the efficiency metric



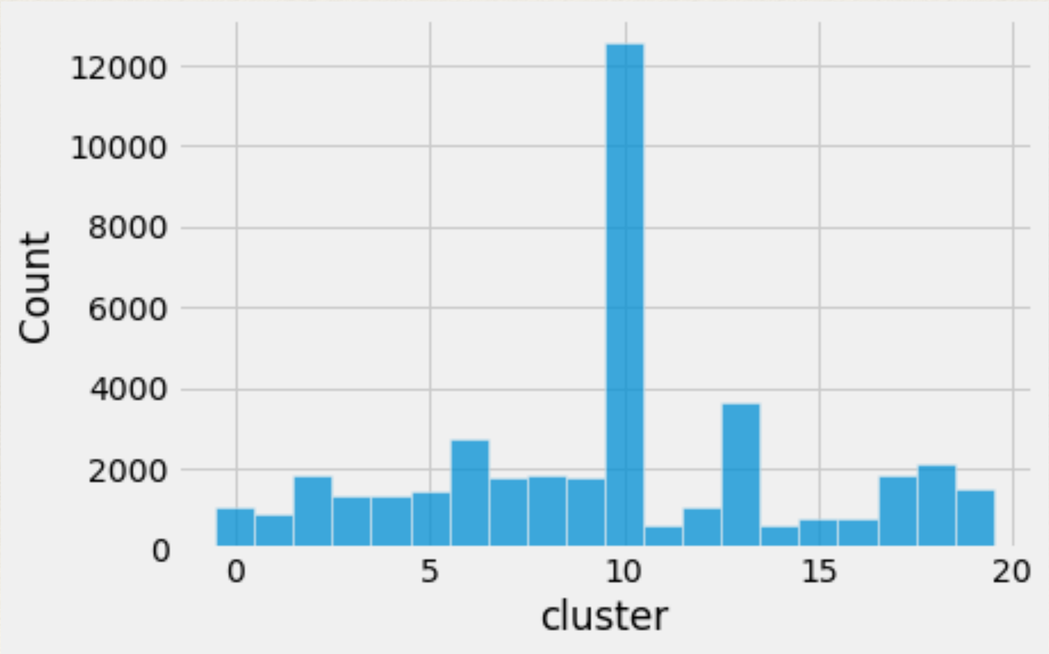
- Silhouette of Coefficients



Continued

- Based on the graphs, data was fitted in the KMean model with 18 clusters

- Clusters distribution



Example

- Recommendation based on similar clusters:
  - sample_food = ‘Strike Signature Fried Chicken’
  - clusteredbased_recommend(sample_food)

```
Chicken Scramble
Chicken Strips And Fries
Chicken Hash-n-eggs
Chicken Pear Sandwich
Clubhouse Wrap
Chicken Caesar Wrap
Chicken And Waffles
Chicken Cucumber Wrap
Grilled Chicken Caesar Wrap
Chicken Burger
```

Conclusion

- We can use any of the recommendations based on the business need.
  - If user is filling up the order, we can recommend foods based on the "Association" model
  - In user home page, we can load the foods based on item based recommendation systems
  - And etc.

Future work

- For the future work, we need more data to train out model with more restauarants food items, and more orders.
- Gathering information about the users, can help to have a better collaboration recommend recommendation
- Building a hybrid system can benefit from different models and recommend a better and more accurate food item.