TEAM 2

PROCESSING BIG DATA

FINAL ASSIGNMENT

TOPIC: PERFORM EXPLORATORY DATA ANALYSIS (EDA) AND FEATURE EXTRACTION FROM RAW RECIPE DATA TO DESIGN A RECOMMENDER SYSTEM.

PREPARED BY: TEAM 2 - JACK NGUYEN & VU MANH TRUNG HAI

CONTENT OUTLINE

ASSIGNMENT OVERVIEW

Key Objective: Perform Exploratory Data Analysis (EDA) and features extraction from raw recipe datasets, supporting the process designing a recommender system.

Including: 6 required tasks and 3 optional tasks

ASSIGNMENT APPROACH

Key approach to implement 9 given tasks is
to import and transfer the two datasets to Python dataframe.

- Objective: Perform Exploratory Data Analysis (EDA) and feature extraction from raw recipe data to design a recommender system.
- RAW_recipes_cleaned.csv: Contains recipe-related information. Link: https://raw-recipes-clean-upgrad.s3.amazonaws.com/RAW_recipes_cleaned.csv
- RAW_interactions_cleaned.csv: Contains user-recipe interaction data. Link: https://raw-interactions-upgrad.s3.amazonaws.com/RAW_interactions_cleaned.csv
- Total 9 tasks are required to complete as below.

Prepared By Jack Nguyễn & Vũ Mạnh Trung Hải

TASKS AND SOLUTION

I. Task 1 – Create New Session – Importing & Reading Dataset

- Read RAW_recipes_cleaned.csv from the provided link.
- Ensure all fields have the correct data types.

	user_id	recipe_id	date	rating	review
0	38094	40893	2003-02-17	4	Great with a salad. Cooked on top of stove for...
1	1293707	40893	2011-12-21	5	So simple so delicious! Great for chilly fall...
2	8937	44394	2002-12-01	4	This worked very well and is EASY. I used not...
3	126440	85009	2010-02-27	5	I made the Mexican topping and took it to bunk...
4	57222	85009	2011-10-01	5	Made the cheddar bacon topping adding a sprin...

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 231637 entries, 0 to 231636
Data columns (total 12 columns):
 #   Column          Non-Null Count   Dtype 
---  ------          --------------   ----- 
 0   name            231636 non-null  object
 1   id              231637 non-null  int32 
 2   minutes         231637 non-null  int32 
 3   contributor_id  231637 non-null  int32 
 4   submitted       231637 non-null  object
 5   tags            231637 non-null  object
 6   nutrition       231637 non-null  object
 7   n_steps         231637 non-null  int32 
 8   steps           231637 non-null  object
 9   description     231637 non-null  object
 10  ingredients     231637 non-null  object
 11  n_ingredients   231637 non-null  int32 
dtypes: int32(5), object(7)
memory usage: 16.8+ MB
```

TASKS AND SOLUTION

I. Task 1 – Create New Session – Importing & Reading Dataset

- **Read RAW interactions from the provided link.**
- **Ensure all fields have the correct data types.**

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1132367 entries, 0 to 1132366
Data columns (total 5 columns):
 #   Column     Non-Null Count    Dtype
---  ------     --------------    -----
 0   user_id    1132367 non-null  int32
 1   recipe_id  1132367 non-null  int32
 2   date       1132367 non-null  object
 3   rating     1132367 non-null  int32
 4   review     1132367 non-null  object
dtypes: int32(3), object(2)
memory usage: 30.2+ MB
```

	name	id	minutes	contributor_id	submitted	tags	nutrition	n_steps	steps	description	ingredients	n_ingredients
0	arriba baked winter squash mexican style	137739	55	47892	2005-09-16	['60-minutes-or-less', 'time-to-make', 'course...	[51.5, 0.0, 13.0, 0.0, 2.0, 0.0, 4.0]	11	['make a choice and proceed with recipe', 'dep...	autumn is my favorite time of year to cook! th...	['winter squash', 'mexican seasoning', 'mixed ...	7
1	a bit different breakfast pizza	31490	30	26278	2002-06-17	['30-minutes-or-less', 'time-to-make', 'course...	[173.4, 18.0, 0.0, 17.0, 22.0, 35.0, 1.0]	9	['preheat oven to 425 degrees f', 'press dough...	this recipe calls for the crust to be prebaked...	['prepared pizza crust', 'sausage patty', 'egg...	6
2	all in the kitchen chili	112140	130	196586	2005-02-25	['time-to-make', 'course', 'preparation', 'mai...	[269.8, 22.0, 32.0, 48.0, 39.0, 27.0, 5.0]	6	['brown ground beef in large pot', 'add choppe...	this modified version of 'mom's' chili was a h...	['ground beef', 'yellow onions', 'diced tomato...	13
3	alouette potatoes	59389	45	68585	2003-04-14	['60-minutes-or-less', 'time-to-make', 'course...	[368.1, 17.0, 10.0, 2.0, 14.0, 8.0, 20.0]	11	['place potatoes in a large pot of lightly sal...	this is a super easy great tasting make ahea...	['spreadable cheese with garlic and herbs', 'n...	11
4	amish tomato ketchup for canning	44061	190	41706	2002-10-25	['weeknight', 'time-to-make', 'course', 'main-...	[352.9, 1.0, 337.0, 23.0, 3.0, 0.0, 28.0]	5	['mix all ingredients& boil for 2 1 / 2 hours ...	my dh's amish mother raised him on this recipe...	['tomato juice', 'apple cider vinegar', 'sugar...	8

TASKS AND SOLUTION

II. Task 2: Extract Nutrition Features

calories	total_fat_PDV	sugar_PDV	sodium_PDV	protein_PDV	saturated_fat_PDV	carbohydrates_PDV
51.5	0.0	13.0	0.0	2.0	0.0	4.0
173.4	18.0	0.0	17.0	22.0	35.0	1.0
269.8	22.0	32.0	48.0	39.0	27.0	5.0
368.1	17.0	10.0	2.0	14.0	8.0	20.0
352.9	1.0	337.0	23.0	3.0	0.0	28.0

Extracting the "Nutrition" Column" into individual features, specifically into 7 separate columns:

- **calories**
- **total_fat_PDV**
- **sugar_PDV**
- **sodium_PDV**
- **protein_PDV**
- **saturated_fat_PDV**
- **carbohydrates_PDV**

This helps us easily compare the dataset and usefully for the analysis processing

TASKS AND SOLUTION

III. Task 3: Standardize Nutrition Values

TARGET/OBJECTIVE # 1

Standardize the nutrition values to a per 100 calories basis to account for serving size variations.

TARGET/OBJECTIVE # 2

Divide each of the nutrition columns (except calories) by the calories value, then multiply by 100.

TARGET/OBJECTIVE # 3

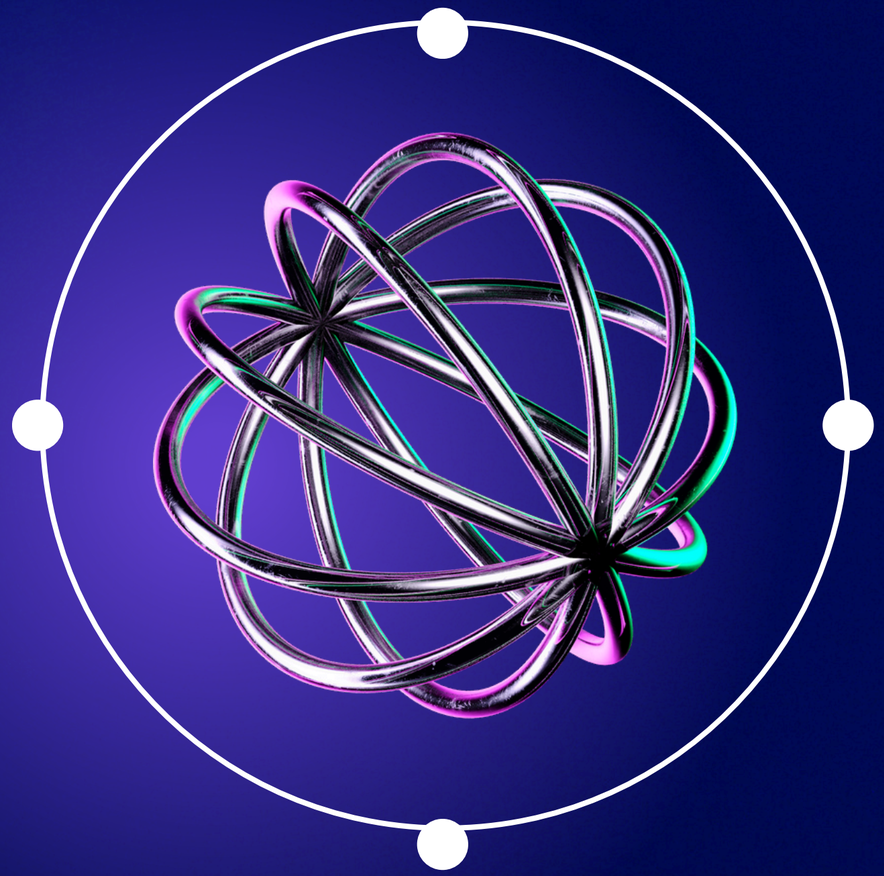
This helps avoid confusion between nutritional values and allows for fair comparisons between recipes with different serving sizes.

calories	total_fat_PDV	sugar_PDV	sodium_PDV	protein_PDV	saturated_fat_PDV	carbohydrates_PDV
51.5	0.000000	25.242718	0.000000	3.883495	0.000000	7.766990
173.4	10.380623	0.000000	9.803922	12.687428	20.184544	0.576701
269.8	8.154188	11.860638	17.790956	14.455152	10.007413	1.853225
368.1	4.618310	2.716653	0.543331	3.803314	2.173322	5.433306
352.9	0.283366	95.494474	6.517427	0.850099	0.000000	7.934259

TASKS AND SOLUTION

IV. Task 4: Convert Tags to Array

The tags column is currently read as a string. Convert it to an array of strings for ease of handling.



```
0    ['60-minutes-or-less', 'time-to-make', 'course...
1    ['30-minutes-or-less', 'time-to-make', 'course...
2    ['time-to-make', 'course', 'preparation', 'mai...
3    ['60-minutes-or-less', 'time-to-make', 'course...
4    ['weeknight', 'time-to-make', 'course', 'main-...
Name: tags, dtype: object

'Tags' Column's Type:
<class 'str'>
```

- **RESULT:**

```
0    [60-minutes-or-less,  time-to-make,  course,  ...
1    [30-minutes-or-less,  time-to-make,  course,  ...
2    [time-to-make,  course,  preparation,  main-di...
3    [60-minutes-or-less,  time-to-make,  course,  ...
4    [weeknight,  time-to-make,  course,  main-ingr...
Name: tags, dtype: object

'Tags' Column's Type:
<class 'list'>
```

TASKS AND SOLUTION

V. Task 5: Read and Join Interaction Data

- Join this interaction data with the recipe data using the appropriate key.
- The resulting dataframe contains all interactions with corresponding recipe information.
- Creating complete datasets for analysis.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1132367 entries, 0 to 1132366
Data columns (total 5 columns):
 #   Column     Non-Null Count    Dtype
---  ------     --------------    -----
 0   user_id    1132367 non-null  int32
 1   recipe_id  1132367 non-null  int32
 2   date       1132367 non-null  object
 3   rating     1132367 non-null  int32
 4   review     1132367 non-null  object
dtypes: int32(3), object(2)
memory usage: 30.2+ MB

Kiểm tra loại dữ liệu 2 cột 'id' của 2 bảng:
int32
int32
```

- **RESULT:**

```
Int64Index: 1132367 entries, 0 to 1132366
Data columns (total 23 columns):
```

	user_id	recipe_id	date	rating	review	name	id	minutes	contributor_id	submitted	...	description	ingredients	n_ingredients	calories	total_fat_PD
0	38094	40893	2003-02-17	4	Great with a salad. Cooked on top of stove for...	white bean green chile pepper soup	40893	495	1533	2002-09-21	...	easy soup for the crockpot.	['great northern beans', 'yellow onion', 'dice...	9	204.8	2.4414
1	1293707	40893	2011-12-21	5	So simple so delicious! Great for chilly fall...	white bean green chile pepper soup	40893	495	1533	2002-09-21	...	easy soup for the crockpot.	['great northern beans', 'yellow onion', 'dice...	9	204.8	2.4414
2	8937	44394	2002-12-01	4	This worked very well and is EASY. I used not...	devilicious cookie cake delights	44394	20	56824	2002-10-27	...		[devil's food cake mix, 'vegetable oil', 'eggs...	4	132.3	8.3144
3	126440	85009	2010-02-27	5	I made the Mexican topping and took it to bunk...	baked potato toppings	85009	10	64342	2004-02-25	...	these toppings sure makes a nice change from p...	['mayonnaise', 'salsa', 'cheddar cheese', 'ref...	13	2786.2	12.2747
4	57222	85009	2011-10-01	5	Made the cheddar bacon topping adding a sprin...	baked potato toppings	85009	10	64342	2004-02-25	...	these toppings sure makes a nice change from p...	['mayonnaise', 'salsa', 'cheddar cheese', 'ref...	13	2786.2	12.2747

TASKS AND SOLUTION

VI. Task 6: Create Time-Based Features

- Create features that capture the time elapsed between a review date and the recipe's submission date.
- Helps us analysing the rating based on time.
- Helps us analysing the trend of time.
- Through this, we can detect the time sample in user behavior.

	date	submitted	elapsed_days	elapsed_months	elapsed_years
1132357	2009-05-18	2007-08-30	627	20	1
1132358	2009-06-19	2007-08-30	659	21	1
1132359	2010-05-15	2007-08-30	989	32	2
1132360	2011-08-21	2007-08-30	1452	47	3
1132361	2013-04-17	2011-12-31	473	15	1
1132362	2007-09-05	2007-07-19	48	1	0
1132363	2010-09-06	2007-07-19	1145	37	3
1132364	2011-08-07	2007-10-08	1399	45	3
1132365	2003-12-09	2003-10-06	64	2	0
1132366	2009-09-29	2009-08-24	36	1	0

TASKS AND SOLUTION

VII. Task 7: Process Numerical Columns

1. Column years_since_submission_on_review_date – elapsed_years:

- **Introduce non-linearity to numerical columns by converting them into categorical columns using binning (percentile-based bucketing).**
- **Analyze the average rating for each bucket to determine the usefulness of the bucketed column.**
- **Handle any data inconsistencies or edge cases appropriately.**

- **Drop rows with elapsed_years < 0; and re-calculate the new range and average rating**
- **Create 4 equally spaced bins based on the new range**

- **BEFORE:**

```
>> The range of values for each bucket:
IntervalIndex([(-4.0, 0.0], (0.0, 2.0], (2.0, 5.0], (5.0, 19.0]], dtype='interval[float64, right]')


>> The average rating for each bucket:
  elapsed_years_bucket    rating
0                   Q1  4.538891
1                   Q2  4.527487
2                   Q3  4.441549
3                   Q4  4.118415
```

- **AFTER:**

```
New range of values for each bucket:
Q1: (0.0, 4.75]
Q2: (4.75, 9.5]
Q3: (9.5, 14.25]
Q4: (14.25, 19.0]
---------------------------------


Average rating for each bucket:
  elapsed_years_bucket    rating
0                   Q1  4.514166
1                   Q2  4.284806
2                   Q3  3.854682
3                   Q4  3.583277
```

TASKS AND SOLUTION

VII. Task 7: Process Numerical Columns

2. Other columns:

```
>> Processing column: minutes
>> The range of values for each bucket in column 'minutes':
IntervalIndex([(0.0, 20.0], (20.0, 40.0], (40.0, 70.0], (70.0, 2147483647.0]], dtype='interval[float64, right]')

>> The average rating for each bucket in column 'minutes':
  minutes_bucket    rating
0             Q1  4.464342
1             Q2  4.413638
2             Q3  4.402361
3             Q4  4.357621


--------------------------------------
>> Processing column: calories
>> The range of values for each bucket in column 'calories':
IntervalIndex([(0.0, 176.4], (176.4, 312.7], (312.7, 512.7], (512.7, 434360.2]], dtype='interval[float64, right]')

>> The average rating for each bucket in column 'calories':
  calories_bucket    rating
0              Q1  4.415222
1              Q2  4.428538
2              Q3  4.418094
3              Q4  4.382222
```

1. minutes:

- **Q4 range (70.0, 2147483647.0) is disproportionately large compared to other ranges, likely due to an extreme outlier.**
- **Consider filtering out excessively large values (e.g., minutes > 500 or a reasonable threshold).**

2. calories:

- **Q4 extends to 434360.2, another potential outlier. Such a high calorie value may indicate data entry errors or uncommon cases.**
- **Suggest capping calorie values at a plausible limit for human consumption (e.g., 5000).**

3. total_fat_PDV:

- **The range and average rating distribution seem consistent without extreme outliers.**
- **Average rating increases from Q1 to Q4, possibly indicating customer preference for higher-fat content.**

4. sugar_PDV:

- **Q4 extends to 108.15. While possible, values above 100% DV for sugar may warrant closer scrutiny.**
- **The rating in Q4 is lower, possibly indicating customer dissatisfaction with high sugar content.**

```
--------------------------------------
>> Processing column: total_fat_PDV
>> The range of values for each bucket in column 'total_fat_PDV':
IntervalIndex([(0.0, 4.4], (4.4, 6.973], (6.973, 9.312], (9.312, 18.71]], dtype='interval[float64, right]')

>> The average rating for each bucket in column 'total_fat_PDV':
  total_fat_PDV_bucket    rating
0                   Q1  4.384768
1                   Q2  4.396886
2                   Q3  4.415400
3                   Q4  4.447074


--------------------------------------
>> Processing column: sugar_PDV
>> The range of values for each bucket in column 'sugar_PDV':
IntervalIndex([(0.0, 2.76], (2.76, 8.111], (8.111, 28.276], (28.276, 108.15]], dtype='interval[float64, right]')

>> The average rating for each bucket in column 'sugar_PDV':
  sugar_PDV_bucket    rating
0               Q1  4.411859
1               Q2  4.426201
2               Q3  4.410769
3               Q4  4.395269
```

TASKS AND SOLUTION

VII. Task 7: Process Numerical Columns

2. Other columns:

```
>> Processing column: sodium_PDV
>> The range of values for each bucket in column 'sodium_PDV':
IntervalIndex([(0.0, 2.347], (2.347, 5.004], (5.004, 8.944], (8.944, 199909.091]], dtype='interval[float64, right]')

>> The average rating for each bucket in column 'sodium_PDV':
  sodium_PDV_bucket    rating
0                Q1  4.411953
1                Q2  4.399911
2                Q3  4.423292
3                Q4  4.408967


-------------------------------------
>> Processing column: protein_PDV
>> The range of values for each bucket in column 'protein_PDV':
IntervalIndex([(0.0, 2.907], (2.907, 6.069], (6.069, 11.822], (11.822, 50.107]], dtype='interval[float64, right]')

>> The average rating for each bucket in column 'protein_PDV':
  protein_PDV_bucket    rating
0                Q1  4.412504
1                Q2  4.412819
2                Q3  4.417047
3                Q4  4.401750
```

5. sodium_PDV:
   - Q4 range (8.944, 199909.091) is exceptionally wide, likely skewed by outliers.
   - Sodium percentage values rarely exceed 200% in realistic scenarios. Consider capping.
6. protein_PDV, saturated_fat_PDV, and carbohydrates_PDV:
   - Bucket ranges are generally reasonable, though Q4 still contains higher values than most typical datasets.
   - If these values are critical for analysis, review the data source to validate upper-range accuracy.

```
-------------------------------------
>> Processing column: saturated_fat_PDV
>> The range of values for each bucket in column 'saturated_fat_PDV':
IntervalIndex([(0.0, 3.417], (3.417, 7.819], (7.819, 12.73], (12.73, 50.128]], dtype='interval[float64, right]')

>> The average rating for each bucket in column 'saturated_fat_PDV':
  saturated_fat_PDV_bucket    rating
0                       Q1  4.395406
1                       Q2  4.405552
2                       Q3  4.411922
3                       Q4  4.431243


-------------------------------------
>> Processing column: carbohydrates_PDV
>> The range of values for each bucket in column 'carbohydrates_PDV':
IntervalIndex([(0.0, 1.602], (1.602, 3.273], (3.273, 4.756], (4.756, 16.204]], dtype='interval[float64, right]')

>> The average rating for each bucket in column 'carbohydrates_PDV':
  carbohydrates_PDV_bucket    rating
0                       Q1  4.434505
1                       Q2  4.424427
2                       Q3  4.399632
3                       Q4  4.385553
```

7. carbohydrates_PDV:
   - Q4 extends to 16.204, which is reasonable compared to other columns.
   - The drop in ratings from Q1 to Q4 aligns with customer concerns about high carbohydrate levels.
   -

TASKS AND SOLUTION

VII. Task 7: Process Numerical Columns

2. Other columns:

OBJECTIVE

- 1. Outlier Ranges:
- Some buckets have unusually large upper limits, such as:
- minutes: Q4 extends to 2147483647.0, likely due to outliers.
- sodium_PDV: Q4 extends to 199909.091, which may also be an extreme outlier.
- These extreme ranges can skew analysis, especially in the Q4 buckets.
- 2. Rating Consistency Across Buckets:
- In most columns, there is a noticeable decrease in average rating from Q1 to Q4 (e.g., minutes, calories, carbohydrates_PDV).
- This pattern might reflect diminishing customer satisfaction or other factors as the metric increases.
- Columns like total_fat_PDV and saturated_fat_PDV show the opposite, with ratings increasing slightly toward Q4.

TASKS AND SOLUTION

VIII. Task 8: Create User-Level Features

user_id	user_avg_rating	user_avg_n_ratings	user_avg_years_betwn_review_and_submission	user_avg_prep_time_recipes_reviewed	user_avg_n_steps_recipes_reviewed
1533	4.710938	128	0.671875	90.890625	9.000000
1535	4.473552	794	2.780856	128.691436	8.151134
1581	5.000000	1	4.000000	200.000000	18.000000
1634	3.616667	60	0.416667	39.083333	6.750000
1676	4.677419	31	3.193548	54.580645	10.806452
1755	4.500000	2	0.000000	72.500000	8.500000
1773	4.000000	4	0.500000	148.750000	14.750000
1792	4.516129	31	0.612903	217.032258	8.419355
1891	4.837838	37	1.405405	68.432432	9.405405
1938	0.000000	1	3.000000	70.000000	18.000000

Object:

- **Understand user's preferences through how they rate the recipes**

user_avg_n_ingredients_recipes_reviewed	user_avg_calories_recipes_reviewed	user_avg_total_fat_per_100_cal_recipes_reviewed	user_avg_sugar_per_100_cal_recipes_reviewed	user_avg_sodium_per_100_cal_recipes_reviewed
9.726562	402.182813	7.232297	13.635123	28.080239
7.498741	459.144710	6.080650	24.520015	9.022532
10.000000	955.500000	11.407640	0.104657	3.244375
7.283333	441.733333	7.413221	12.370164	44.969933
10.806452	492.374194	6.871258	7.649615	7.425225
8.500000	502.500000	6.311691	6.051129	16.342920
12.500000	376.225000	5.205000	15.434200	11.951161
10.419355	805.951613	6.875646	16.338751	12.777067
9.000000	728.321622	7.033525	21.912074	9.275158
16.000000	322.100000	8.382490	5.898789	20.180068

Next --->

TASKS AND SOLUTION

VIII. Task 8: Create User-Level Features

user_avg_protein_per_100_cal_recipes_reviewed	user_avg_saturated_fat_per_100_cal_recipes_reviewed	user_avg_carbohydrates_per_100_cal_recipes_reviewed	user_avg_prep_time_recipes_reviewed_high_ratings
8.730892	9.841813	3.013691	91.222222
5.634724	8.183975	3.903838	168.285714
15.175301	16.117216	0.104657	200.000000
8.435200	9.216597	2.660588	17.958333
14.660073	6.470179	2.221801	56.038462
12.776813	7.659390	3.004027	120.000000
10.969341	6.869033	3.584875	275.000000
8.420937	8.994926	3.281245	290.619048
6.858461	8.810185	3.413732	67.806452

user_avg_prep_time_recipes_reviewed_high_ratings	user_avg_n_steps_recipes_reviewed_high_ratings	user_avg_n_ingredients_recipes_reviewed_high_ratings	user_avg_years_betwn_review_and_submission_high_ratings
91.222222	8.842593	9.481481	0.740741
168.285714	8.039916	7.262605	2.693277
200.000000	18.000000	10.000000	4.000000
17.958333	6.375000	7.250000	0.500000
56.038462	11.269231	10.846154	3.230769
120.000000	9.000000	12.000000	0.000000
275.000000	19.000000	16.000000	0.000000
290.619048	7.666667	10.047619	0.619048
67.806452	8.967742	8.677419	1.161290
0.000000	0.000000	0.000000	0.000000

Next --->

TASKS AND SOLUTION

VIII. Task 8: Create User-Level Features

Post-processing:

- **Check data thoroughly after creation**
- **Handle null values that may appear during conversion**

```
1  user_level_features.info()
```

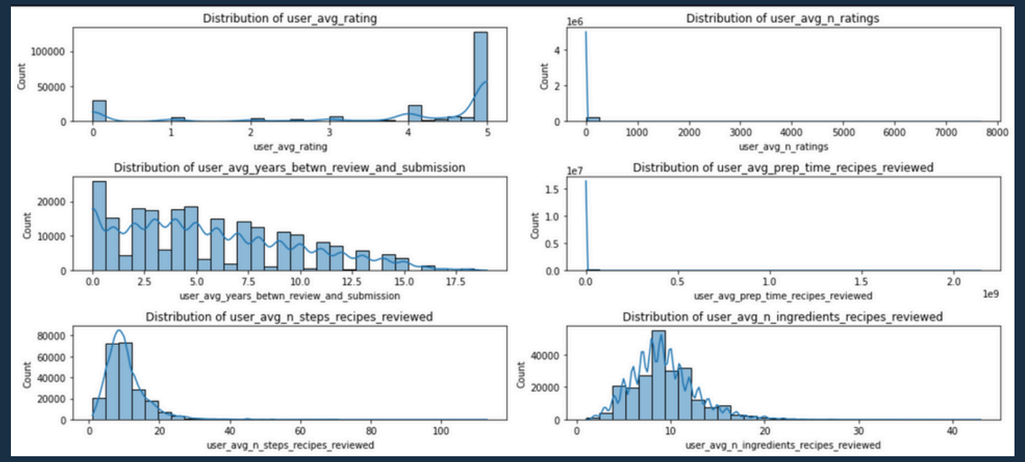
Output | Terminal | Debug Console

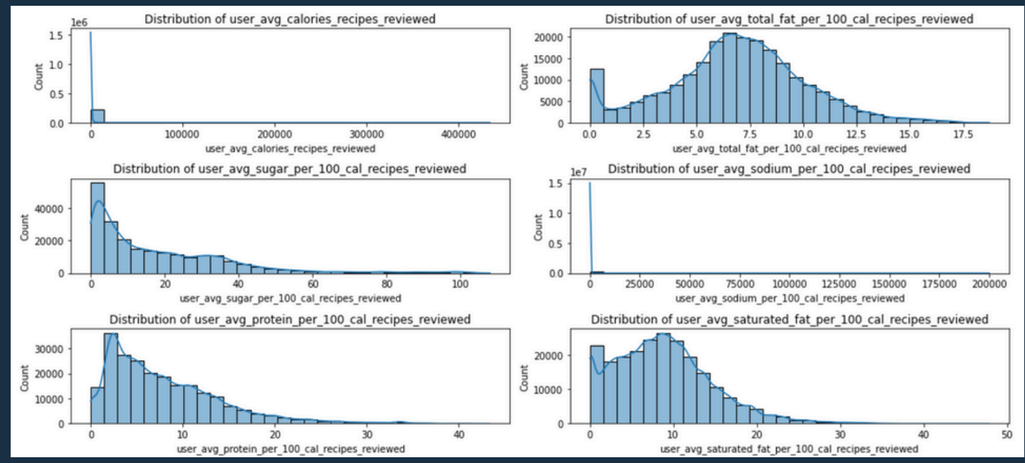
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 226527 entries, 1533 to 2002372706
Data columns (total 17 columns):
 #   Column                                          Non-Null Count   Dtype
---  ------                                          --------------   -----
 0   user_avg_rating                                 226527 non-null  float64
 1   user_avg_n_ratings                              226527 non-null  int64
 2   user_avg_years_betwn_review_and_submission      226527 non-null  float64
 3   user_avg_prep_time_recipes_reviewed             226527 non-null  float64
 4   user_avg_n_steps_recipes_reviewed               226527 non-null  float64
 5   user_avg_n_ingredients_recipes_reviewed         226527 non-null  float64
 6   user_avg_calories_recipes_reviewed              226527 non-null  float64
 7   user_avg_total_fat_per_100_cal_recipes_reviewed 226527 non-null  float64
 8   user_avg_sugar_per_100_cal_recipes_reviewed     226527 non-null  float64
 9   user_avg_sodium_per_100_cal_recipes_reviewed    226527 non-null  float64
 10  user_avg_protein_per_100_cal_recipes_reviewed   226527 non-null  float64
 11  user_avg_saturated_fat_per_100_cal_recipes_reviewed 226527 non-null  float64
 12  user_avg_carbohydrates_per_100_cal_recipes_reviewed 226527 non-null  float64
 13  user_avg_prep_time_recipes_reviewed_high_ratings    226527 non-null  float64
 14  user_avg_n_steps_recipes_reviewed_high_ratings      226527 non-null  float64
 15  user_avg_n_ingredients_recipes_reviewed_high_ratings 226527 non-null  float64
 16  user_avg_years_betwn_review_and_submission_high_ratings 226527 non-null  float64
dtypes: float64(16), int64(1)
memory usage: 31.1 MB
```

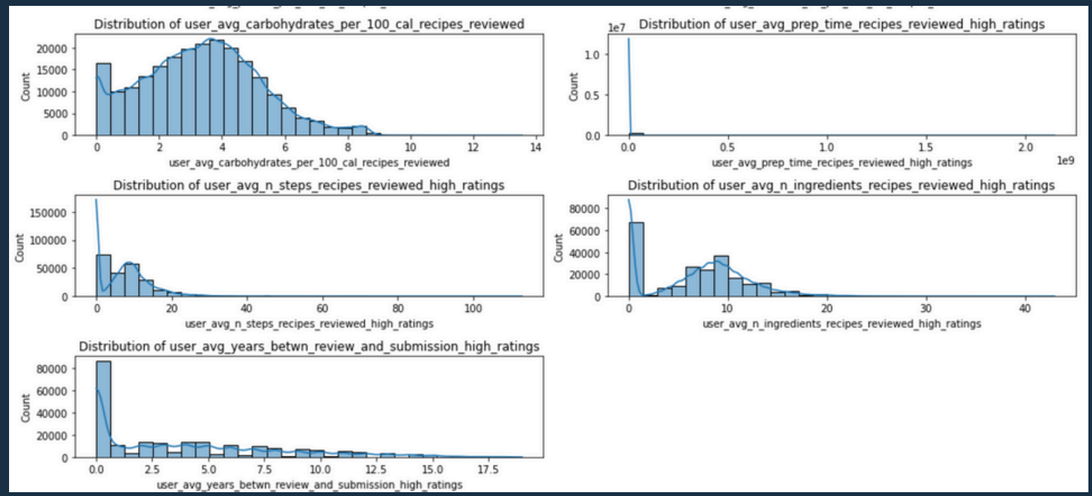
Next --->

TASKS AND SOLUTION

VIII. Task 8: Create User-Level Features







Next --->

TASKS AND SOLUTION

IX. Task 9: Create Tag-Level Features

APPROACH FOR TASK

Given the task and the tags column structure, the solution involves analyzing the tags, selecting valuable ones, and encoding them for further analysis. Here's the suggested method:

> Step 1: Preprocessing the Tags Column Objective: Parse the tags column, which contains lists of tags as strings.

> Step 2: Identify Valuable Tags

- Frequent Tags: Identify the top 5% most frequently occurring tags.
- Highest-Rated Tags: Compute average ratings per tag and select the top 5% highest-rated and bottom 5% highest-rated tags.

Method:

- Explode the tags column so each row contains one tag.
- Calculate tag frequency and average rating per tag.
- Use percentile thresholds to filter top/bottom tags.

> Step 3: Encoding Tags Use one-hot encoding as the default encoding method. Explore advanced encoding techniques like binary encoding or frequency encoding if the one-hot matrix becomes too large.

> Step 4: Assert for Data Integrity Validate that selected tags and encoded features are created as expected. Use assert statements to ensure consistency (e.g., tag count matches expected values, no missing data).

TASKS AND SOLUTION

IX. Task 9: Create Tag-Level Features

EXTRACTING INFORMATION FROM TAGS

```
1  from IPython.display import display
2
3  # Display DataFrame with all columns visible
4  with pd.option_context('display.max_columns', None):
5      display(join_df)
```

put Terminal Debug Console

tag_main-ingredient	tag_honduran	tag_american	tag_ultimate fiesta with this sopaipillas recipe from Food.com.	tag_easy	tag_60-minutes-or-less	tag_beef-crock-pot	tag_eggs-breakfast	tag_crock-pot-main-dish	tag_low-sodium	tag_for-large-groups-holiday-event	tag_heirloom-historical-recipes	tag_preparation	tag_north-american	tag_breakfast-casseroles	tag_roast-beef-main-dish	tag_equipment	tag_baked-beans	tag_mushroom-soup	tag_bear	tag_cuisine	tag_beginner-cook	tag_occasion
1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

TASKS AND SOLUTION

IX. Task 9: Create Tag-Level Features

EXTRACTING INFORMATION FROM TAGS

```
Top 5% Most Of Frequent Tags:
[' preparation', ' course', ' time-to-make', ' dietary', ' main-ingredient', ' easy', ' occasion', ' equipment', ' cuisine', ' low-in-something', ' main-dish', ' n
umber-of-servings', ' meat', '60-minutes-or-less', ' taste-mood', ' north-american', ' vegetables', ' oven', ' 4-hours-or-less', '30-minutes-or-less', ' low-carb',
' holiday-event', ' desserts', ' healthy', ' dinner-party', ' low-sodium', ' american', ' beginner-cook', ' low-cholesterol']

Top 5% Highest-Rated Tags:
[' baked-beans', ' beans-side-dishes', ' beef-kidney', ' beef-sauces', ' breakfast-casseroles', ' breakfast-eggs', ' breakfast-potatoes', ' cabbage', ' cranberry-s
auce', ' danish', ' desserts-easy', ' desserts-fruit', ' eggs-breakfast', ' for-large-groups-holiday-event', ' heirloom-historical-recipes', ' irish-st-patricks-da
y', ' main-dish-pasta', ' middle-eastern-main-dish', ' pasta-salad', ' pork-loin', ' pork-loins-roast', ' prepared-potatoes', ' ragu-recipe-contest', ' roast-beef-
comfort-food', ' roast-beef-main-dish', ' side-dishes-beans', ' simply-potatoes2', 'Throw the ultimate fiesta with this sopaipillas recipe from Food.com.', 'health
y']

Bottom 5% Lowest-Rated Tags:
[' bean-soup', ' bear', ' beef-barley-soup', ' beef-crock-pot', ' birthday', ' black-bean-soup', ' celebrity', ' chicken-crock-pot', ' chicken-stew', ' chicken-ste
ws', ' crock-pot-main-dish', ' fillings-and-frostings-chocolate', ' ham-and-bean-soup', ' high-in-something-diabetic-friendly', ' honduran', ' jellies', ' lamb-she
ep-main-dish', ' main-dish-beef', ' main-dish-pork', ' main-dish-seafood', ' marinara-sauce', ' mushroom-soup', ' pork-crock-pot', ' pot-roast', ' pressure-cannin
g', ' served-hot-new-years', ' shrimp-main-dish', ' snacks-kid-friendly', ' snacks-sweet', ' stews-poultry', ' sugar-cookies', ' unprocessed-freezer', ' water-bat
h', 'cuisine']
```

THANK YOU