CS513 Data Cleaning Group Project

Phase-I

Team32: Ruoyu QIAN (ruoyuq2@illinois.edu), Yasmin YIN (yueyin8@illinois.edu)

Dataset Chosen

The dataset we chose is Farmers Markets.

Description of Dataset

The Farmers Markets dataset contains 59 columns. It starts with FMID which is the id number of farmer markets. FMID contains seven digits. Then it has Market name which contains several characters, numbers and special characters along with whitespaces. The next 5 columns are different social medias: Website, Facebook, Twitter, YouTube, and OtherMedia in different formats. Most of their values are URLs of websites, and some of them is account name or username of social media with or without social media name and '@'. For example, some values of Twitter column are '@BenWallach1' and 'athertonmillmkt'. They may contain null or empty value.

The following few columns are address of farmers markets: street, city, county, state and zip. They don't have any format limitation as well, for instance, zip column can have '-' special character. They may also contain null or empty values. The next few columns are season date and time. There are 8 columns to include 4 seasons which represent opening period in one year and they are nullable. For instance, one market was opened from 09/02/2017 to 06/30/2018 and from 8:30 AM-1:00 PM on Saturday.

The next 2 columns are x and y which represent the latitude and longitude of the farmer's location. Then the following column is location which is string and nullable.

The following 35 columns are all mainly boolean values (Y/N). Amongst these, the value of the first 5 columns is not nullable, which is either Y or N. The next column (Organic) is nullable as well, but it contains special character '-'. Then the following 29 columns are nullable.

The last column is 'updateTime' which does not have specific format, some of them only have a year, some of them have date and time. Date and time does not follow any format as well, some is formatted as 'mm/dd/yyyy hh:mm:ss AM', some as 'mmm dd yyyy hh:mm AM'.

Farmers Markets	
FMID	int
MarketName	text
Website	text
Facebook	text
Twitter	text
Youtube	text
OtherMedia	text
street	text
city	text
County	text
State	text
zip	text
Season1Date	text
Season1Time	text
Season2Date	text
Season2Time	text
Season3Date	text
Season3Time	text
Season4Date	text
Season4Time	text
x	text
y	text
Location	text
Credit	Boolean
WIC	Boolean
WICcash	Boolean
SFMNP	Boolean
SNAP	Boolean
Organic	Boolean
Bakedgoods	Boolean
Cheese	Boolean
Crafts	Boolean
Flowers	Boolean
Eggs	Boolean
Seafood	Boolean
Herbs	Boolean
Vegetables	Boolean
Honey	Boolean
Jams	Boolean
Maple	Boolean
Meat	Boolean
Nursery	Boolean
Nuts	Boolean
Plants	Boolean
Prepared	Boolean
Soap	Boolean
Trees	Boolean
Wine	Boolean
Coffee	Boolean
Beans	Boolean
Fruits	Boolean
Grains	Boolean
Juices	Boolean
Mushrooms	Boolean
PetFood	Boolean
Tofu	Boolean
WildHarvested	Boolean
updateTime	Text

Fig 1. Single table

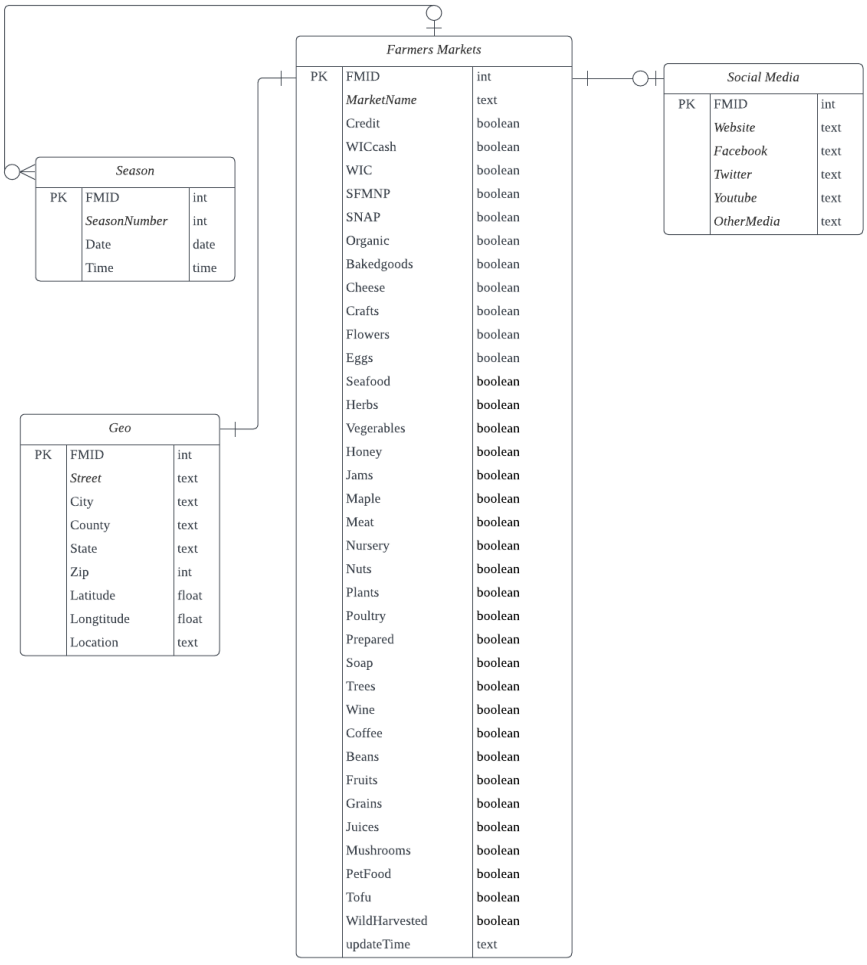

Fig 2. Splitted tables

Use Cases

1. “Zero cleaning” use case U0: data cleaning is not necessary

   U0: How many farmer’s markets in the dataset accept credit card payment?

   We first load the dataset into python:

   ```
   import pandas as pd
   data = pd.read_csv('./datasets/farmersmarkets-2017-01-10.csv')
   data.head()
   ```

   It is pretty clear that the first column, `FMID`, is the unique identifier of each farmer’s market. We also have a column, `Credit`, that we can use to determine whether a farmer’s market accepts credit card payment.

   Before calculating the number, we do some sanity checks to see if there is any data cleaning required:

   1) Are there any missing values in these columns?

      ```
      data.isnull().sum()
      ```

      It returns 0 for both columns, which means we don’t have any missing data in either column `FMID` or `Credit`.

   2) Do we have multiple data types in the `FMID` column?

      ```
      data.FMID.dtypes
      ```

      It returns `dtype('int64')`, which means the datatype of the `FMID` column is consistent for all the records.

   3) Do we need any data cleaning for the `Credit` column?

      ```
      data.Credit.unique()
      ```

      It returns `array(['Y', 'N'], dtype=object)`, indicating this column only takes two possible values and no data cleaning is needed.

   With those checks, we believe this is a “zero cleaning” use case where data cleaning is not necessary.

2. “Main” use case U1: data cleaning is necessary and sufficient

   U1: How many farmer’s markets in each county sold coffee and accepted WIC payment in 2016?

We first need to do some sanity checks to determine what data cleaning we need to do:

1) Is the `County` column clean?

```
data.County.unique()
```

Based on the results, we saw some values clearly referring to the same county but in different formats, for example, we have both `Adair` and `Adair County`. Therefore, we need to clean up this column and standardize how we name counties such that we can group by the `County` column later on.

2) Do we need any data cleaning for the `Coffee` column?

```
data.Coffee.unique()
```

It returns `array(['Y', 'N', nan], dtype=object)`, indicating we have some missing value in this column. There are a couple of different approaches to deal with missing values –

   a) Removing the rows with missing data;
   b) Filling missing values with mode.

We need to decide which approach we want to take.

3) Do we need any data cleaning for the `WIC` column?

```
data.WIC.unique()
```

It returns `array(['Y', 'N'], dtype=object)`, indicating this column only takes two possible values and no data cleaning is needed.

4) How do we want to get the information about open years?
We found there are 8 columns including up to 4 seasons when each farmer's market was open. However, the data formats are not consistent across rows and we need to figure out a way to first extract year information out of those 8 columns, and then validate if 2016 was included.

This is a use case where data cleaning is necessary and sufficient to perform the analysis.

Updates:

U1: Build a recommendation system that given a buyer's preferred payment type, the list of the items they need to buy, the county they live in, and their gps location (i.e., longitude and latitude), returns the list of the farmer's markets in the same county that satisfy all the needs, along with their most recent opening season, rank ordered by the distance between user and the farmer.

1. we need to check and clean all the columns related to payment types;
2. we need to check and clean all the columns related to products;
3. we need to rank order the opening seasons for each market;
4. we need to clean x, y (longitude and latitude) in the dataset, and calculate haversine distance based on users' latitude and longitude for all eligible markets, then rank order them based on the distance;
5. we need to clean and preprocess input data too, in order to best match with the database.

3. "Never enough" use case U2 : data cleaning is not sufficient

   U2: Which farmer's market most recently updated their information?
   At first sight, we might think this question can be answered by cleaning the last column of the dataset `updateTime`. However, further investigation indicates that no amount of data cleaning or wrangling will make the dataset suitable for this use case.

   The `updateTime` is documented very clearly for a subset of the farmer's markets. One such example is the farmer's market `1011967`, whose `updateTime` was `5/18/2016 9:16:39 PM`. However, some farmer's markets only have the information of year in the column. Even if we do the data cleaning and reformat the column with regular expression, we will never be able to backfill missing information about the date and time.

   Another U2 could be: Find the farmer's markets with a Twitter account. Data cleaning won't be sufficient to answer achieve this because –
   1) It's not guaranteed that the URL under the `Twitter` column is a valid twitter website;
   2) Some markets may have twitter accounts listed under other columns, such as `OtherMedia`, but considering URLs are pretty freeform, it's hard to validate whether one URL is `Twitter` or not.

Data Quality Problems

1. List obvious data quality problems with evidence (examples and/or screenshots)

   1) One major issue in this dataset is the inconsistent data format among the rows and even in the columns.

      For columns containing values in string format, we have a very inconsistent format – some of the values are in lower case, some upper case, some with whitespaces before or after. We even have inconsistent data format in column

names, for example, `street` and `city` are in lower case while `County` and `State` have capitalized first character.

Also, there are different naming conventions for the same `County` among different records. For instance, the following two farmer's markets are both in `Albany`, but under County column, one is listed as `Albany` while another is listed as `Albany County`:

FMID	MarketName	Website	Facebook	Twitter	Youtube	OtherMedia	street	city	County	State	zip
1000132	Farmers Market at the Crossings	NaN	NaN	NaN	NaN	NaN	Crossings Park of Colonie, Albany-Shaker Rd, a...	Loudonville	Albany	New York	NaN
1007023	Glenmont/ Mt Moriah Church Farmer's Market	NaN	NaN	NaN	NaN	NaN	Mt. Moriah Church, across from Lowes 262 Route...	Glenmont	Albany County	New York	NaN

Another example is `Baltimore` and `Baltimore City`:

FMID	MarketName	Website	Facebook	Twitter	Youtube	OtherMedia	street	city	County	State	zip
1006929	Eastpoint Farmers' Market	NaN	NaN	NaN	NaN	NaN	7839 Eastern Avenue	Baltimore	Baltimore	Maryland	21224
1007590	Fells Point Farmers' Market	NaN	NaN	NaN	NaN	NaN	700 S Broadway	Baltimore	Baltimore City	Maryland	02011

2) We also have missing data issues with the dataset. For instance, we may miss the information about whether a farmer's market sells certain items and we need to figure out how to deal with other missing data:

Nursery	Nuts	Plants	Poultry	Prepared	Soap	Trees	Wine	Coffee	Beans	Fruits	Grains	Juices
N	N	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y
N	N	Y	N	N	N	N	N	N	N	Y	N	N
N	N	N	Y	Y	Y	N	N	N	N	Y	N	N
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

3) Another issue in this dataset is inconsistent format and recording conventions when it comes to date and time.

We may miss the end date of season information like what's show in the following example:

Season1Date	Season1Time	Season2Date	Season2Time	Season3Date	Season3Time	Season4Date	Season4Time
05/25/2013 to	Sat: 9:00 AM-1:00 PM;	05/24/2014 to 09/27/2014	Sat: 9:00 AM-1:00 PM;	05/23/2015 to 09/26/2015	Sat: 9:00 AM-1:00 PM;	05/28/2016 to 09/24/2016	Sat: 9:00 AM-1:00 PM;

We may only have months but not year info like the next example:

July to November	Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:00 pm;	NaN	NaN	NaN	NaN	NaN	NaN

In some rows, Season1Date to Season4Date stores open seasons from the earliest to the latest, but in some other rows, the information might not be sorted:

05/07/2016 to 07/02/2016	Sat: 8:00 AM-12:00 PM;	07/06/2016 to 09/28/2016	Wed: 8:00 AM-12:00 PM;Sat: 8:00 AM-12:00 PM;	10/01/2016 to 10/29/2016	Sat: 8:00 AM-12:00 PM;	11/05/2016 to 12/31/2016	Sat: 10:00 AM-1:00 PM;
04/16/2016 to 11/22/2016	Tue: 3pm-6pm;Sat: 8am-1pm;	11/26/2016 to 12/24/2016	Sat: 10:00 AM-12:00 PM;	01/02/2016 to 04/09/2016	Sat: 10:00 AM-12:00 PM;	NaN	NaN

2. Explain why / how data cleaning is necessary to support the main use case U1
    1) In order to support the main use case U1, we first need to get the county information of each farmer's market. Based on the sanity check mentioned above, we are seeing some values clearly referring to the same county but in different formats. Therefore, we need to clean up that column and standardize how we name counties such that we can group by the `County` column later on. Otherwise, `Baltimore` and `Baltimore City` would be considered as two different counties even though they actually refer to the same one.

    2) We have the missing data issue with the `Coffee` column. As mentioned above, we need to decide which approach we want to take to deal with missing values. Some options include:
        a) Removing the rows with missing data;
        b) Filling missing values with mode.

    3) The use case asks about the year 2016 in particular, therefore we need to get the information about open seasons. We have 8 columns including up to 4 seasons when each farmer's market was open but the data formats are not consistent across rows. Examples are shown above. We need to figure out a way to first extract year information out of those 8 columns, and then validate if 2016 was included for each market.

Initial Plan for Phase-II

Our plan to clean this dataset:

Steps	Details	Assignee	Timeline
S1	Review and update the use case description	Ruoyu	07/12/2023
S2	Profile D to identify DQ problems	Yasmin	07/15/2023
S3	Perform DC “proper” using OpenRefine: 1. Trim whitespaces 2. Upper/lower case 3. Convert to Datetime/number 4. Format website/media 5. Modify typo in address 6. Facet and cluster	Yasmin & Ruoyu	07/19/2023
S4	Data quality checking	Yasmin & Ruoyu	07/23/2023
S5	● Document and quantify change: ○ Transformation description ○ Screenshot of DC ● IC violations detected	Yasmin & Ruoyu	07/29/2023