Restaurant Menu Data
New York Public Library
(Data Cleaning Case Study)

Alpas, Michael
Department of Computer Science, UIUC
(309)533-8994
malpas2@illinois.edu

Singh, Mohit
Department of Computer Science, UIUC
(309)750-7110
Mohits3@illinois.edu

Yadav, Upendra Singh
Department of Computer Science, UIUC
(215)439-4579
Usyadav2@illinois.edu

1. INTRODUCTION

Our team once again embarked to a new journey as part for CS513 Theory and Practice of Data Cleaning class. This is our third project as a team and always excited for the learning and camaraderie.

We understand that data cleaning is a very important process in providing Data Science or Machine Learning solutions. Most of the time, engineers need to spend a vast majority of their time in cleaning the data to be able to provide even the simplest data visualization to business partners.

For this project, we selected the historical restaurant menu offered by NY Public Library, we think the dataset is challenging for us to apply the data cleaning techniques we learned from the class. The focus of this project is to identify data quality issues from the data set and provide data cleaning solutions to address the problem.

2. DATASET

For this project we have used open source publicly available restaurant Menu dataset from New York Public Library. This dataset is one of the largest restaurant menu collection in the world which is used by historians, nutritional scientists, chefs, novelists and food enthusiasts. The main credit of this dataset goes to Miss Frank E. Buttolph, who collected more than 25,000 menus between 1900 to 1924 on behalf of NYPL library. This collection of data contains approximately 45,000 menus dating from 1840s to present in which about quarter of menus are digitized and made available in NYPL digital library. [1]

This data can be downloaded from below location –

http://menus.nypl.org/data

This data contains below 4 files which are related to each other. High level description of these files is given below –

File Name	Description	No. of Fields
Menu.csv	• Contains Information about Menus of different restaurants.	20
MenuPage.csv	• Information of Menu Pages referenced to Menu. • Each Menu can have multiple Menu Pages.	7
MenuItem.csv	• Information about Menu Items referenced to each Menu Page. • Each Menu Page can contain multiple Menu Items.	9
Dish.csv	• Contains information about dishes mapped to each Menu Item. • Referenced to MenuItem.csv	9

2.1 Menu.csv

This is one of the primary files among all the 4 files, which has menu information associated with different restaurants. Each menu is uniquely identified by Menu Id. Menu.csv file has relationship with MenuPage.csv file using Menu Page Id. One menu item can have multiple Menu Pages.

Following fields are present in the Menu.csv.

- Id - unique id for each Menu.
- Name – name of Menu
- Sponsor – name of the restaurant
- Event – name of the event for which menu was created.
- Venue – type of place where food was served from Menu, Ex – Educational, Private etc.
- Place – address where the menu was used. (Includes city and state name)
- Physical_Description – size of menu card.
- Occasion – type of event such as anniversary, birthday etc.
- Notes – any comments about Menu.
- Call_Number – menu's call number.

- Keywords – keywords for Menu.
- Language – language in which Menu was printed.
- Date – date in which menu was collected.
- Location – place where Menu was used.
- Location_Type – type of location.
- Currency – currency used in Menu.
- Currency_Symbol – symbol of the currency used in Menu.
- Status – transcription status of the Menu such as complete or under review.
- Page_Count – total number of pages in the Menu.
- Dish_Count – total Number of dishes in the Menu.

2.2 MenuPage.csv

Following fields are present in MenuPage.csv.

- Id - unique Identifier of MenuPage.csv
- Image_Id – id of the image of Menu page.
- Menu_Id – foreign key from Menu.csv
- Page_Number – page number of Menu.
- Full_Height – height of menu page
- Full_Width – width of menu page
- UUID – unique Identifier

2.3 MenuItem.csv

Following fields are present in MenuItem.csv.

- Id – unique identifier for each row in Menu Item.
- Menu Page Id – menu item referenced to Menu Page.
- Price – price of the menu item.
- High_Price – price of the costliest portion of the item,
- Dish_Id – relationship between menu item and dish.
- Created_At – record creation date.
- XPos – X axis coordinate of the menu item in the scanned menu page.
- Ypos – Y axis coordinate of the menu item in the scanned menu page.
- Updated At – record last updated date.

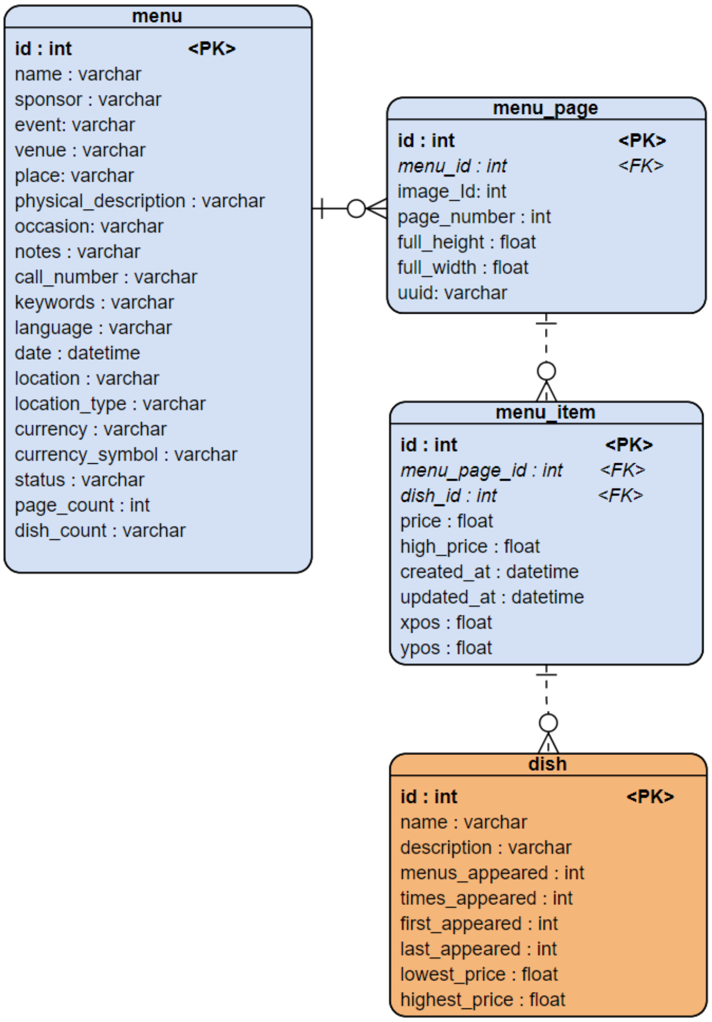
2.4 Dish.csv

Following fields are present in Dish.csv.

- Id – Unique Identifier
- Name – dish name
- Description – description of the dish
- Menus_Appeared – number of menus where dish appeared.
- Times_Appeared – how many times dish appeared in a menu.
- First_Appeared – when dis appeared at first time.
- Last_Appeared –Year when dish appeared last time in menu.
- Lowest_Price –lowest price of the dish in menu.
- Highest_Price – highest price of the dish in menu

2.3 Entity Relationship Diagram

In this below diagram, we have explained above dataset by creating a relation database and also by creating entity relationship diagram by depicting different tables as entities.[7]



NYPL dataset ER Diagram

3. TARGET USE CASE

Our team explored the dataset and below are the possible use cases –

3.1 Main Use case (U1)- Data Cleaning is necessary

- Most popular dishes as per different location and most popular dishes as per different occasions.

  After cleaning this dataset, we can easily find out which dish was served most in which occasion? Such as which dish was served most in Easter, thanksgiving or wedding anniversaries? We can also find out which dish was more popular in which location or place?

  To work on above use case, we will have to Join Menu, Menu Page, Menu Item and Dish tables and we will have to clean Dish name, menu appeared fields from Dish table, Location, Place and occasion fields from Menu table.

3.2 Corner Use case (U0) – Zero Data Cleaning

- What was the costliest (highest price) dish sold in last 100 years?

  There are some use cases where we can easily do some analysis in dataset without doing any data cleaning. One of the examples of such use case should be to find out the dish name which was sold in highest price in last 100 years of data.

  To obtain this information, we just need two columns of Dish table (Dish Name and Highest Price), we can sort dishes based on prices and can identify dish name which was sold in highest price.

- When (In which particular year) a new dish was introduced and in which year it was removed from menu.

  Getting above information will help to many food enthusiasts to understand when a particular dish was introduced in restaurants or when it was removed from their menu completely? This data we can easily get by analyzing Dish table and using two columns such as – Dish name, first appeared and last appeared.

3.3 Corner Use case (U2) – No amount of Data Cleaning

- Price trend (highest and lowest price) of the dishes over the years (timeline) in different events, occasions and locations.

  Usually most of the relevant use cases we can get from this dataset by vigorously cleaning the dataset using the data cleaning technique taught in CS513 class.

  However, in some use cases such as finding out highest and lowest price of dishes over the years in different locations, events and occasions. In such use cases where multiple columns are involved which needs cleaning of multiple columns, it becomes quite challenging to give detail of every costliest and cheapest dish in every year as per different location and event.

  One of the major issue, we will get here that while cleaning multiple columns (such as dish name, occasions, event, location, sponsor names), we will encounter multiple blank data and also bad data across multiple columns, due to that it would be very difficult to provide correct metric for this analytic query.

  Along with above mentioned use cases, we can also find out multiple supplement use cases as mentioned below -

- Cheapest and costliest restaurants according to location over the timeline

  By this dataset, we can identify which restaurants served costliest and which served cheapest dishes. We can also categorize costliest and cheapest restaurants according to location. This use case will help us to understand which locations had costliest restaurants in last 100 years.

- Trend analysis of a dish price fluctuation over the years

  By analyzing this dataset, we can try to analyze how price of a dish changed over years in last century. Are there any dishes which became costlier over the time?

- Restaurants serving unique dishes.

  We can also find out, are there any unique dishes which were served by some unique restaurants. Such as we can find out restaurants which are serving Japanese, Chinese or Filipino dishes.

- Trend of dishes over the timeline

  We can also find out an interesting trend about which dishes were more popular in which decade.

4. DATA QUALITY ISSUES

During our dataset exploration, it seems the NY Public Library dataset will be challenging for the team – it will build our confidence and technical acumen by applying our knowledge learned from CS 513 Theory and Practice of Data Cleaning. For implementing the use case U1 the common data quality issues we observed are trailing white spaces and special characters. We also discussed clustering similar words to fix possible duplicate data. There are also several data columns having blank values that we might disregard during the process.

Another consideration we discussed was adding another dimensionality in some of the dataset. Basically, extracting a new column depending on the final use case of the data. In relation to this, removing some of the columns that might not be needed for the use case is another data cleaning opportunity.

Since for implementation of U1 use case, we will have to join all 4 tables (Menu, MenuPage, MenuItem, Dish), so we analyzed all the columns given in all 4 tables.

Below is the snippet of data columns describing quality issue:

4.1.1 Menu.csv

Column	Description	Sample Value	Quality Issue
name	Menu name	"Victoria Luise"	- Contains special character. - Some of the rows has blank value.
sponsor	Menu	? HOTEL	- Contains

	sponsor		special character. - Some of the rows has blank value.
event	Time of menu (i.e., breakfast, lunch)	?	- Some rows have question mark as a value. - Some of the rows has blank value.

4.1.2 Dish.csv

Column	Description	Sample Value	Quality Issue
name	Dish name	" sautees	- Contains special character. - Some has numeric value.
description	Dish description		- Blank description
first appeared	Year dish was introduced	1	- Invalid year format

4.1.3 MenuItem.csv

Column	Description	Sample Value	Quality Issue
price	Item price		- Some of the rows has blank value.
high_price	Item high price		- Some of the rows has blank value.
xpos	Menu X position of the item	0	- Contains zero value.

4.1.4 MenuPage.csv

Column	Description	Value	Quality Issue
page_number	Page reference on the menu		- Some of the rows has blank value

5. DEVISE THE PLAN

Our initial thought process is to use OpenRefine to clean the data, applying the standard data cleaning operations – unnecessary space removal, purging special characters by applying RegEx patterns and clustering similar words.[3]

We learned that OpenRefine's clustering feature has the advantage compared to other tools -- it eliminates the tedious and time-consuming process by simply uploading the file and apply the Text facet.

Each CSV files will be separately loaded to OpenRefine and will apply data cleaning strategies. Once satisfied and reached the desired data sanity, we will export the cleaned data to another CSV file.

Below is the summary of cleanup activities we are planning to execute:

5.1.1 Menu.csv (size: 3.2 MB, rows: 17547)

These are the columns we initially we want to clean.

- *Name*
  - Remove special characters using GREL.
- *Sponsor*
  - Remove special characters using GREL.
  - Trim white spaces
  - Standardized name format (either all uppercase or all lowercase)
  - Apply facet, cluster operation technique, and merge relevant clusters.
- *Event*
  - Determine approach for blank values, maybe remove it
  - Possibly standardized values to 3 categories – breakfast, lunch and dinner
- *Date*
  - Standardized date format to YYYY-MM-DD

5.1.2 Dish.csv (size: 26.8 MB, rows: 428086)

This is the column we initially we want to clean.

- *Name*
  - Remove special characters using GREL
  - Trim white spaces
  - Apply facet, cluster operation technique, and merge relevant clusters.

5.1.3 MenuItem.csv (size: 118.6 MB, rows: 1048575)

No immediate concern on the data set when we initially loaded it through OpenRefine. Though blank values have been observed, we will determine our approach depending on the use case we want to pursue.

5.1.4 MenuPage.csv (size: 4.7 MB, rows: 66937)

No immediate concern on the data set when we initially loaded it through OpenRefine. Though blank values have been observed, we will determine our approach depending on the use case we want to pursue.

5.2 Tentative Work Assignments

Each team members are eager to contribute to this project and we discussed to give individual opportunity to clean the data, since there are 4 datasets available. While performing data cleaning activities, each team members will need to produce screenshots of the process performed and journal challenges encountered.

There is other aspect of the project that each team members will be responsible:

Member	**Responsibility**
Michael/Mohit	Develop relational database schema. Formulate SQL code.[6]
Upendra	Create workflow models. [4] [5] Possible Python code to complete other data cleaning activities. [8]

6. REFERENCES

[1] http://menus.nypl.org/about

[2] https://regexr.com/

[3] Data Munging: String Manipulation, Regular Expressions, and Data Cleaning DOI - 10.1002/9781119092919.ch4

[4] https://www.researchgate.net/publication/335136628_Towards_Automated_Data_Cleaning_Workflows

[5] Model based approach for developing Data Cleaning Solutions https://dl.acm.org/doi/10.1145/2641575

[6] https://www.sqlite.org/index.html

[7] ER Diagram - https://online.visual-paradigm.com/drive/#diagramlist:proj=0&new=ERDiagram

[8] https://realpython.com/python-data-cleaning-numpy-pandas/