CS598 – Data Cleaning

Final Project: Data Cleaning Project

***1. Introduction……………………………....................................................................................2***

***2.Overview and Initial Assessment………………………….......................................................2***

**2.1 Dataset…................................................................................................................... 2**

**2.2 Proposed UseCase....................................................................................................2**

**2.3 Data Cleaning Goals .................................................................................................3**

**2.4 Data Cleaning Tools and Approach.........................................................................3**

**2.5 Data file and cleaning summary...............................................................................4**

**2.6 Initial Issues Summary .............................................................................................5**

***3 Data Cleaning with OpenRefine and Trifecta……... .............................................................6***

***4 Relational Database Schema.................................................................................................7***

**4.1 Schema.....................................................................................................................7**

**4.2 Integrity Constraints…............................................................................................7**

***5 Workflow Model.............................................................................................................8***

**5.1 Key Inputs/Outputs……………………............................................................8**

**5.2 Workflow Diagram...................................................................................8**

***Appendix A – Trifecta Data Cleaning Scripts....................................................................13 Appendix B – SQLite Logical Integrity Checking Script and Output ...............14***

***Appendix C – YesWorkflow Script....................................................................................15***

**1.Introduction**

In this report farmers market data is analyzed and cleansed to solve some common use cases.

This report is done as part of Individual submission of data cleaning project done by varun sharma at UIUC.

The tools used in the project are OpenRefine for clustering and refining data, SQLite for relational database, Trifacta Data Wrangler and YesWorkflow for provenance.

**2. Overview and Initial Assessment**

2.1 Dataset

I am using data from the United States Department of Agriculture. Maintained by the Agricultural Marketing Service, the Directory is designed to provide customers with convenient access to information about farmers market listings to include: market locations, directions, operating times, product offerings, accepted forms of payment, and more.

URL for the data set:

https://www.ams.usda.gov/local-food-directories/farmersmarkets

2.2 Proposed Use Cases

Proposed use case for this dataset is to provide easy access to consumers to clean Farmer market data where they can easily browse trusted data to figure out details of food markets in the area. Consumers can have more details about the farmer markets.

This data can be fed to other web application or IA engines/automated assistant like Alexa or google voice to make them smarter. This use case is not part of the project but worth mentioning because questions like which markets have organic food zip? 2. Which market are open during night time of 8-11pm? can be answered easily using the clean data set used in this project.

2.3 Data Cleaning Goals

The main goal for this project is to clean data based on data quality dimensions using farmer’s market dataset.

1.**Completeness**: There are lot of columns with missing information. For example youtube column has 8558 null values which lowers the completeness score for the data.

2.**Consistency:** Data set has consistency issues like values NA,N.A,\_ are used for missing values. Also, city values like ST. Louis and ST Louis exist in the dataset.

**3.Uniqueness**: Data set scores high on unique as there are no duplicates in the data set. Though there are multiple records for same store but their location are different in country.

**4.Validity:** There are some validity problems in data set like Facebook column has URL and some places it doesn’t have typical URL format.

**5.Accuracy:** Accuracy % drops because of inconsistent formats used for the fields for example:

Update time field =201 has only year mentioned in it in one record and most of records have

**6.Timeliness:** Data is downloaded from the website directly and cleansed which has some delay but it’s up to date as its maintained by the United States Department of Agriculture.

Following whitepaper was consulted for this.

<https://www.whitepapers.em360tech.com/wp-content/files_mf/1407250286DAMAUKDQDimensionsWhitePaperR37.pdf>

As discussed above data set has multiple issues. Following are the major high-level issues in the data set:

1: Lots of leading and trailing whitespaces in column values.

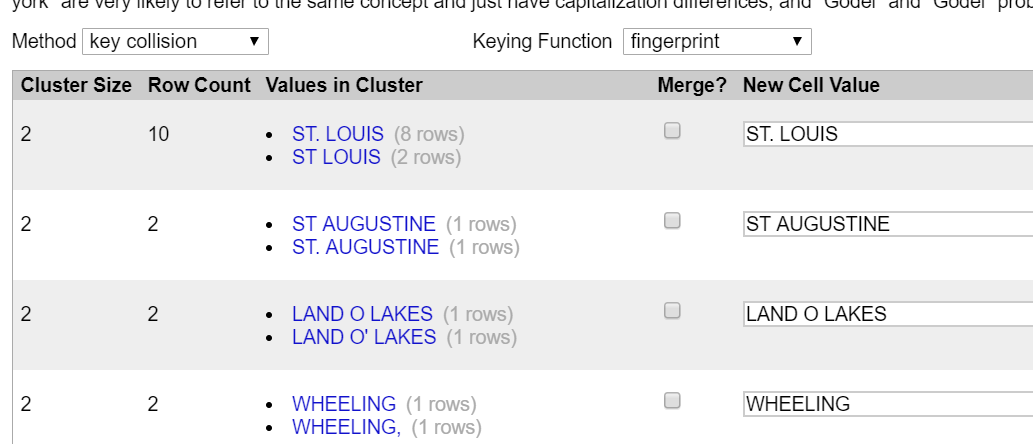
2: Markets with missing information like youtube

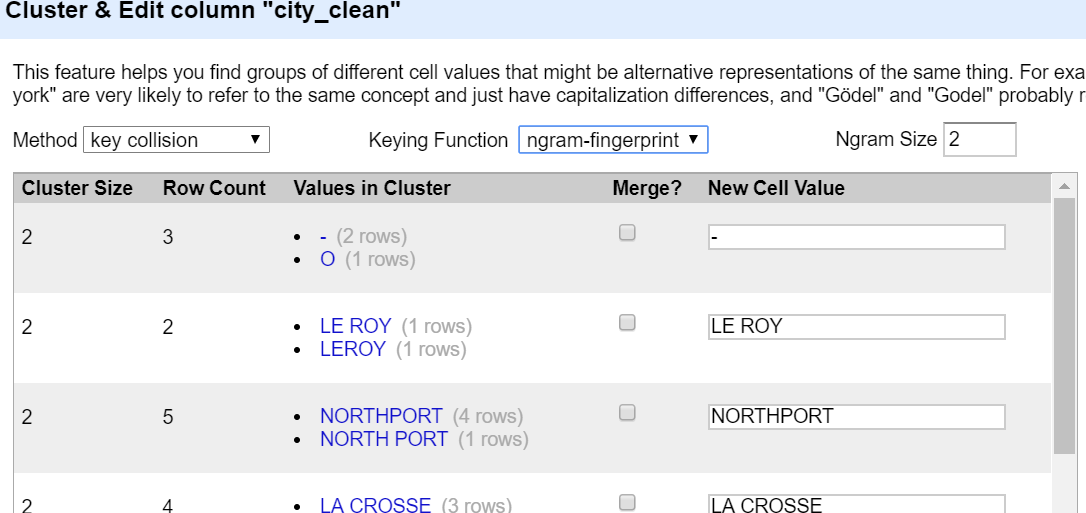
3: Standardization for date/time formats.

4.Standard values for cities.

2.4 Data Cleaning Tools and Approach

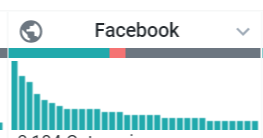
OpenRefine is used in this project for common data cleaning steps like removing white spaces, line breaks and doing basic transformations. Clustering capability helped to achieve data consistency goal for data quality. Clustering in openrefine was used for columns like county, cities:





After OpenRefine step Trifecta Data Wrangler is used to check data quality. I felt trifacta data wrangler’s visual representation is better than open refine. Wrangler gives capability to profile data column wise and understand data better.

Each column data pattern can be seen in the UI



Red here marks invalid values or mismatched values.

I have used YesWorkflow to create a graphical representation of the data cleaning process of the farmers market data. With<http://try.yesworkflow.org/>

I have developed one workflow for steps in OpenRefine.

2.5 Data file and cleaning summary

Farmers market data, and this a summary of some of the columns and cleaning method adopted

Summary of the file:



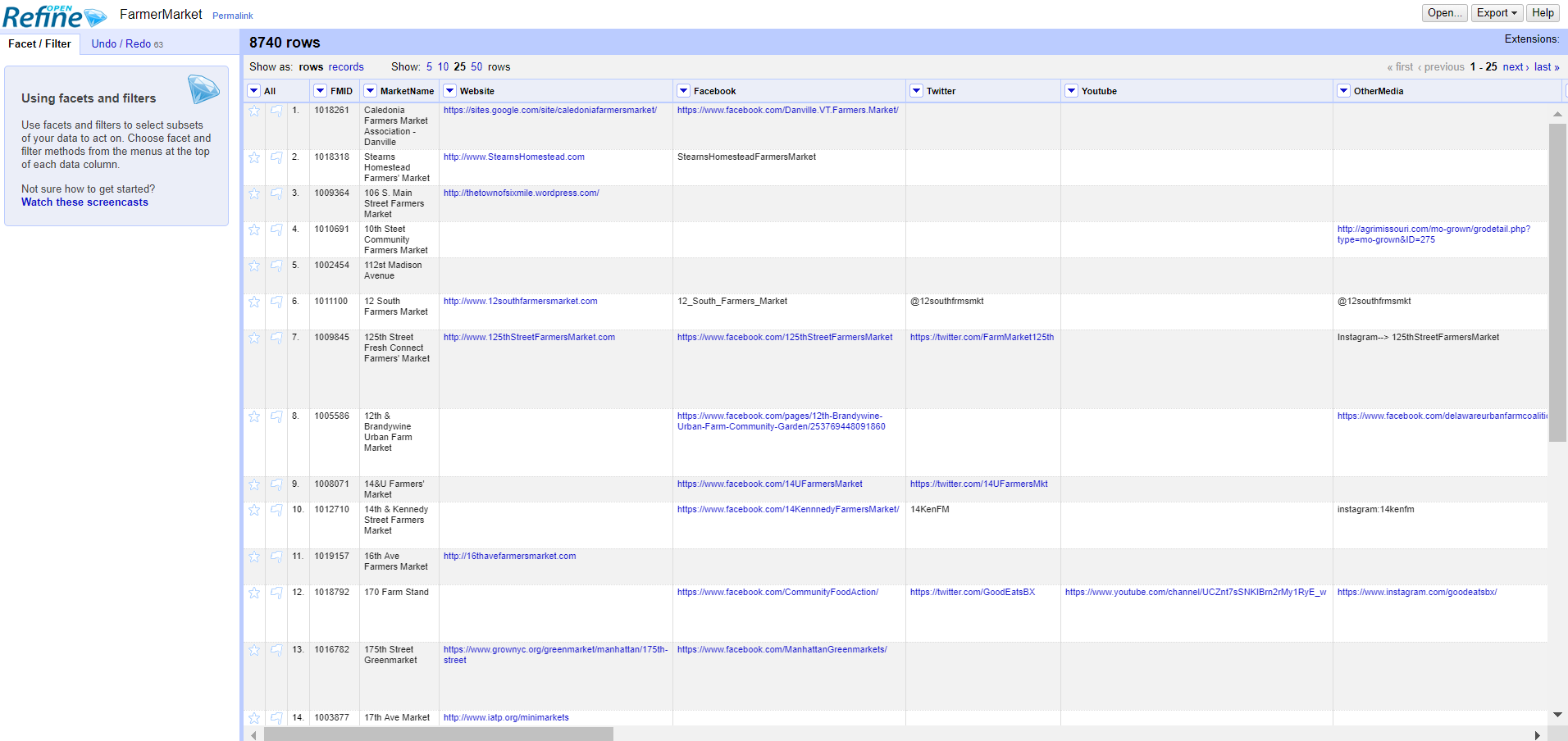
|  |  |  |
| --- | --- | --- |
| Column Name | Description | Cleaning Operations |
| FMID | ID farmer Market | Data is clean no operations on this column |
| MarketName | Name of the Market | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization. |
| Website | Website URL | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization.  Example:  ht[tp://www.seela.org](javascript:%7b%7d)  [http://www.seela.org/](javascript:%7b%7d)  Standardized value:  ht[tp://www.seela.org](javascript:%7b%7d)  Trifacta: |
| Facebook | Facebook link | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization.  Trifacta: |
| Twitter | Twitter Username | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization  Not available information is converted to space or null in open refine.   * [n/a](javascript:%7b%7d)(7 rows) * [N/A](javascript:%7b%7d)(2 rows) * [NA](javascript:%7b%7d)(2 rows)   Converted to space  Trifacta |
| YouTube | Youtube Channel | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization  This info is missing in the file so standardized to NA |
| OtherMedia | Other social media accounts | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and ngram fingerprint data standardization  Ngram size 2 |
| Street | Street address for store | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization |
| City | City location for store | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Upper case for city name  3.Cluster and edit using key collision and finger print for data standardization  Created city clean column |
| County | Name of county | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization |
| State | State location for farmers market | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization |
| Zip | Zip code for farmers market | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  No clustering is required for this column |
| Season1Date | First season date for maket | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization |
| Season1Time | First season time for market | Trim white spaces and replace new lines in OpenRefine |
| Season2Date | Second season date for market | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization |
| Season2Time | Second season time for market | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization |
| Season3Date | Third season date for market | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization |
| Season3Time | Third season time for market | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization |
| Season4Date | Fourth season date for market | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization |
| Season4Time | Fourth season time for market | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  2.Cluster and edit using key collision and finger print for data standardization |
| Location | Location of the market | Data was clean for this |
| Organic | Flag to indicate if store sells organic products or not | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces  Changed face – to NA to identify if FLAG is NA i.e. Not Available. |
| updateTime | Time the info was updated | Open Refine:  1.Trim white spaces and collapse consecutive whitespaces |

3. Data Cleaning with OpenRefine and Trifecta Data Wrangler

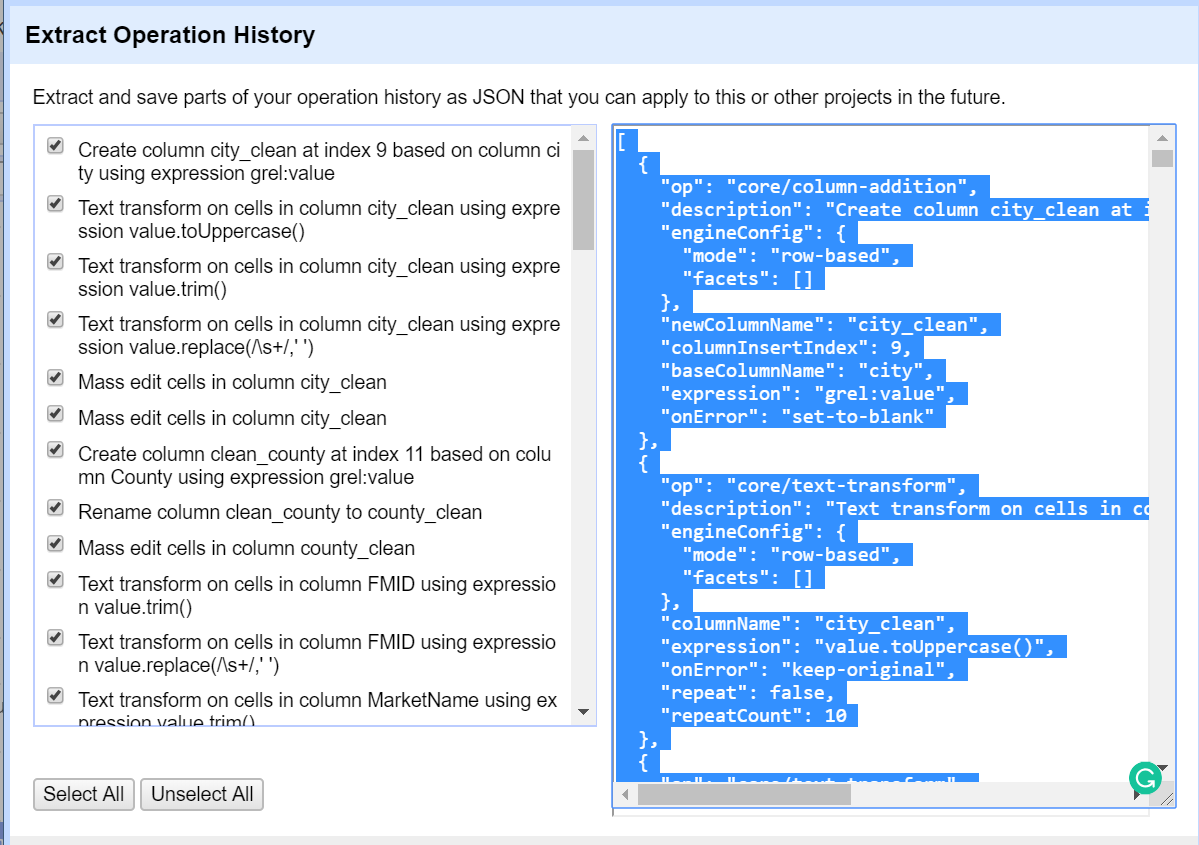
Open Refine tool was used to do data cleaning. Common transforms were used to remove white spaces and to collapse consecutive white spaces. Clustering was done to do data standardization for fields.

Trifecta data wrangler was used for data wrangling and cleanup individual fields.

Sample screenshot for the openrefine operations:

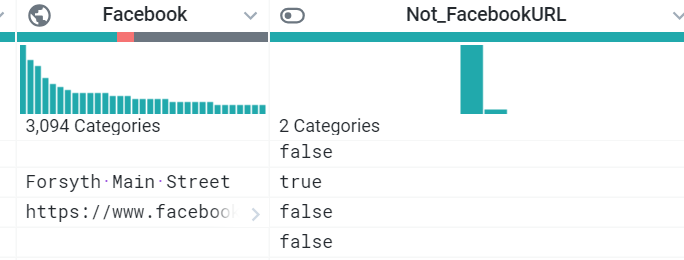


Here is the sample extract for open refine operations done for data quality:



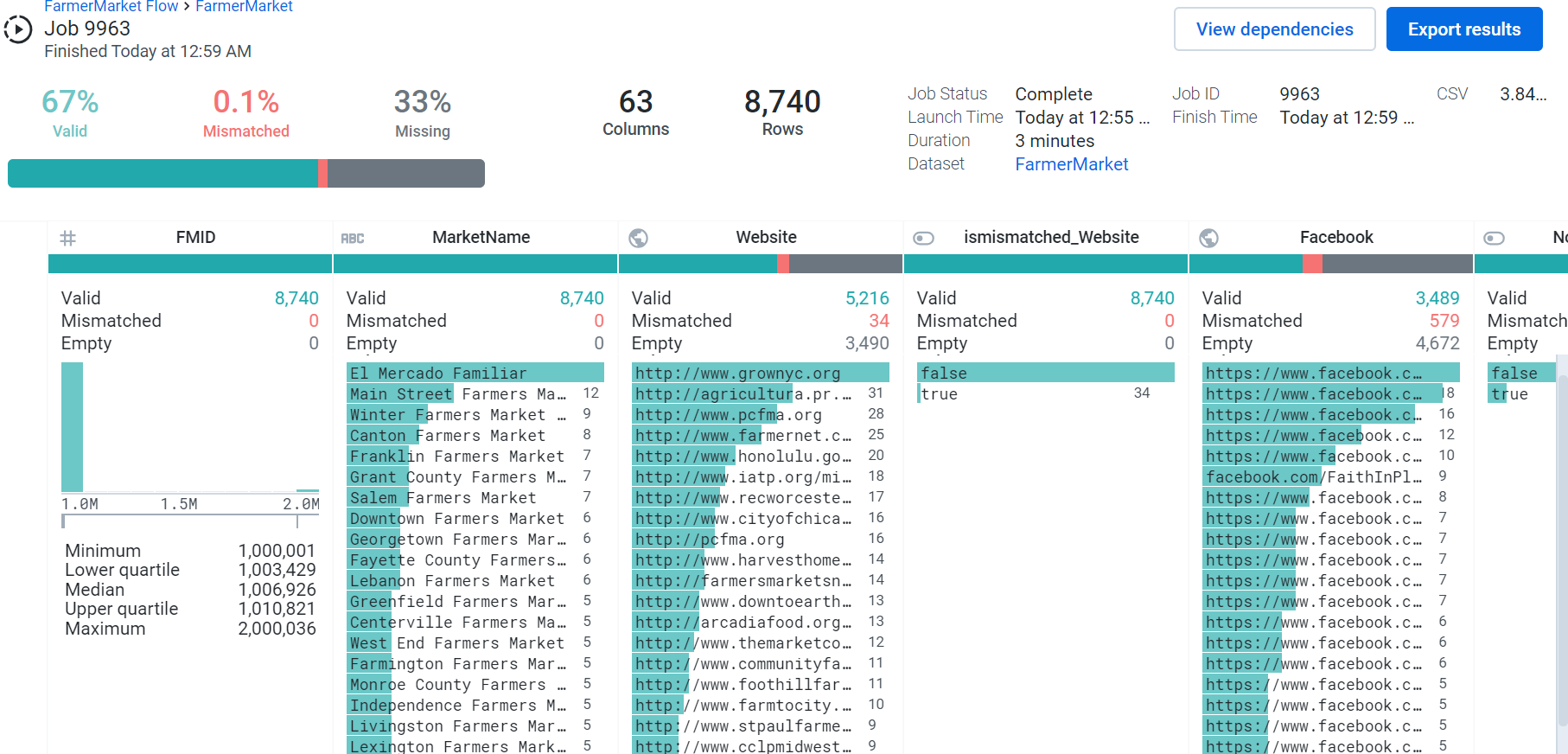
More details are provided as file in the final deliverables.

Trifacta data wrangler steps:



Some Facebook column value don’t have URL. They are represented in different column by flag called Not\_facebookURL to identify such kind of columns.

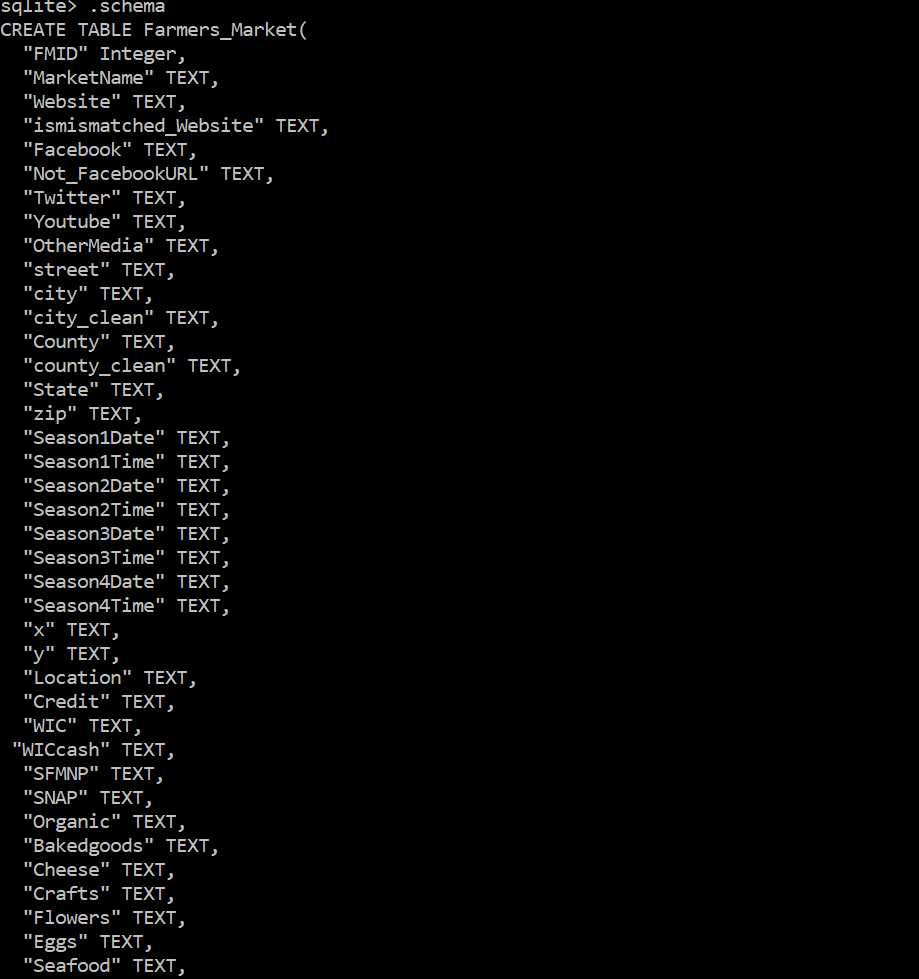
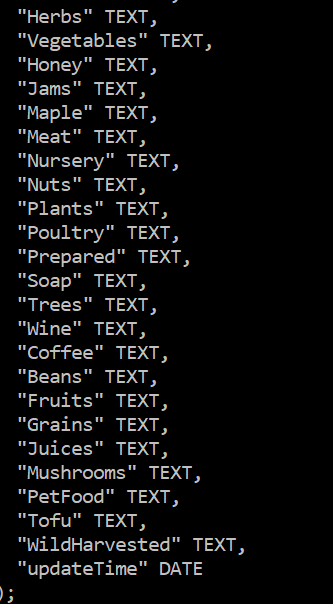
Data Profiling in Trifacta Wrangler tool:



4 Relational Database Schema

4.1 Schema

Sqlite3 was used to load the data file exported from the trifacta data wrangler for next step in the project. Screenshot of the schema:

4.2 Integrity constraints:

Following integrity constraints where identified and described here:

Data Integrity Constraints:

Basic data sanity check:

1. Check number of rows loaded into the sqlite3. Please note If the table already exists, the sqlite3 tool uses all the rows, including the first row, in the CSV file as the actual data to import. Therefore while matching data rows I have deleted the first row from the file as its metadata and not the actual data.

Following IC are written in denial form:

1. Every market has a website URL.
2. URL format is consistent for farmer’s market
3. Every record has city and county information.

5 Workflow Model

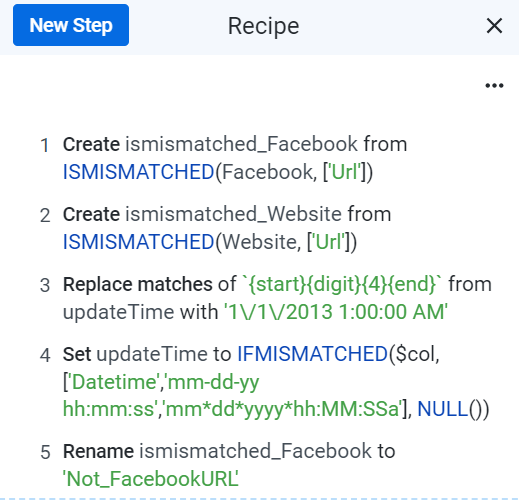
5.1 Key Inputs/Outputs

Our first key input is the farmers market dataset retrieved from the United States Department of Agriculture’s website. After inputting this into OpenRefine and making necessary changes we received two outputs, the modified dataset and the code used by OpenRefine to perform the modifications. The next input is the modified dataset. After inputting the modified dataset into Trifacta and making further changes we received two more outputs, another modified dataset and the code used by Trifacta to perform the modifications. We then used this last modified dataset to input into our database.

5.2 Workflow Diagram

We used YesWorkflow to create a graphical representation of the data cleaning process of the farmers market data. With<http://try.yesworkflow.org/> we were able to create the workflow with the web-based version which allows us to see our changes as they are made. We developed one workflow focusing on the two main steps used to clean the data, OpenRefine and Trifacta. Within those two main steps, there are numerous sub-steps to show the exact steps taken to clean the steps. Below is the workflow showing the steps involved. (with the data removed)

Appendix A – Trifecta Recipe



Appendix B – SQLite Logical Integrity Checking Script and Output

1.Every market has a website URL

Not all farmer's market have website

select count(\*) from farmers\_market where Website="" ;

3490

2.IC check website validity

In negate form finds the non compliant number

select count(\*) from farmers\_market where not\_FacebookURL="TRUE";

564

3.Every record has city and county information

sqlite> select count(\*) from farmers\_market where city="" OR county="";

523

Appendix C – YesWorkflow Script

This is the sample workflow text, more details on the github page**2**

# @begin Final\_Project @desc cleaning farmers market data

# @in farmers\_market

# @out farmers\_martket\_final

# @begin import\_data @desc use OpenRefine to clean data

# @in farmers\_market

# @out MarketName

# @out Website

# @out Facebook

# @out Twitter

# @out YouTube

# @out OtherMedia

# @out Street

# @out City

# @out County

# @out State

# @out Zip

# @out Season1Date

# @out Season1Time

# @out Season2Date

# @out Season2Time

# @out Season3Date

# @out Season3Time

# @out Season4Date

# @out Season4Time

# @out Season4Time

# @out location

# @out columns\_not\_needing\_cleaning

# @end import\_data

# @begin trim @desc trim leading and trailing white spaces

[1] <https://www.ams.usda.gov/local-food-directories/farmersmarkets>

[2] https://github.com/fuxes/cs513-farmersmarkets