A Mouse's Paradise: Big Data Reveals Recipes for Disaster in the NYC Restaurant Industry

Carlos Fuentes, Northwestern University

***Abstract*—Anything involving food and mice has an ick factor. This project is based on data from the New York City Department of Mental Health and Hygiene (DOHMH) Restaurant Inspection Results from December 2011 through 2016. The city inspects about 24,000 restaurants a year to monitor compliance with both city and state food safety regulations. The dataset contains infractions, such as "Evidence of mice or live mice present in facility's food and/or non-food areas".**

This project leverages Cloudera's Hadoop Ecosystem to demonstrate big data's capabilities to explore an open government data set. A database comprised of NYC Restaurant violations is used as an example to explore restaurants with mice, identifying trends by cuisine and NYC borough. The project is easily extensible to include data from alternate sources, including social media sources (like Yelp).

***Index Terms*— Big Data, Cloudera, DOHMH, open Government Data, NYC Restaurant Violations, NYC Restaurant Mice**

I. INTRODUCTION

ANYTHING involving food and mice has an ick factor. According to the CDC [1], mouse contaminated food can lead to several zoonotic diseases (animal diseases that can be passed to humans), including Leptospiosis (kidney damage, meningitis, liver failure, death), Rate-Bite Fever (can be serious and fatal if not treated), Salmonellosis (can be life threatening), and Tularmia (can be life threatening).

To mitigate these risks, NYC inspects about 24,000 restaurants a year to monitor compliance with City and State food safety regulations. In 2010 NYC implemented a letter grade system, scoring restaurants based on a sanity inspection [2]. The inspection data, beginning in 2012, is available for public use though the NYC Open Data website.

This paper explores how Big Data Technologies may be used to explore the ever increasing amount of data available for public consumption. The NYC Restaurant Violations Database, with a focus on violations where mice were identified, was selected as it should relate to most readers who eat out at restaurants.

II. BIG DATA OVERVIEW

Big Data is sometimes referred to as the oil of this century [3]. Oil must be refined and is later processed into useful products. Just as is the case with oil, data is inherently crude and must be refined and processed before it may fuel business decisions. Companies are racing to develop capabilities to refine this to develop actionable insights which drive actionable insights to positively impact the bottom line.

Big Data is much more than a new technology as it represents an enabling science that is transforming organizations. Big Data is a term first coined by Gartner analyst Doug Laney to describe very large, diverse data volumes and the associated pace at which data changes and streams into systems for analysis. These attributes are often referred to as Volume, Variety, and Velocity (or Speed) [4] The amount of data generated and captured by business has grown from terabytes to zetabytes. One of the misconceptions is that big data is about size. Big Data isn't just about size; it is about any of the attributes that challenge the constraints of a systems capability or business need [5] Since the original 3V definition (Volume, Variety, and Veracity), some organizations have included Veracity (quality of data), variability (changing Rapidly), Visualization (making it comprehendible) and Value [6].

III. BIG DATA TECHNOLOGIES

Traditional database technologies (e.g., rational databases) cannot scale to handle the magnitude (volume, variety, and speed) of data as they were designed to handle structured data (e.g.: financial transitions). Big Data's potential lies in its ability to mine for intelligence that cannot be found using traditional techniques and technologies. Advances in computing technology have created new larger sources of data, which may only be analyzed with Big Data. These include unstructured data types like audio, video, three-dimensional models, simulations and location data [7]. Big Data breaks several of the V's, including the Volume, Variety and Velocity barrier found with traditional data techniques.

C. Fuentes is a graduate student in the Masters of Predictive Analytics (MSPA) program at the Northwestern University, Chicago IL 60611 US, has held several C level positions in the financial services, insurance, and telecommunications industries, and has been recognized as one of 100 most influential and notable Hispanic information technology professionals by the Hispanic IT Executive Council (e-mail: carlos@myfuentes.com),

Table 1.0 Technology Comparison

Technology Comparison		
Category	Traditional Database	Big Data
Reliability	Centralized, typically with a single point of failure	Distributed and Scalable
Variety	Fixed schema and relational database	Readily supports Structured and unstructured data
Volume (Scale)	Gigabyte to Terabyte	Petabytes and Exabyte's
Dominant Licensing Model	Traditional Licensing, proprietary software	Open Source

IV. BIG DATA ECOSYSTEM

The big data ecosystem is a collection of components that covers all aspects of Big Data Management and analytics. It includes all technologies that enable Big Data collection, storage, transformation, analysis and presentation of results.

Table 2.0 Ecosystem Components

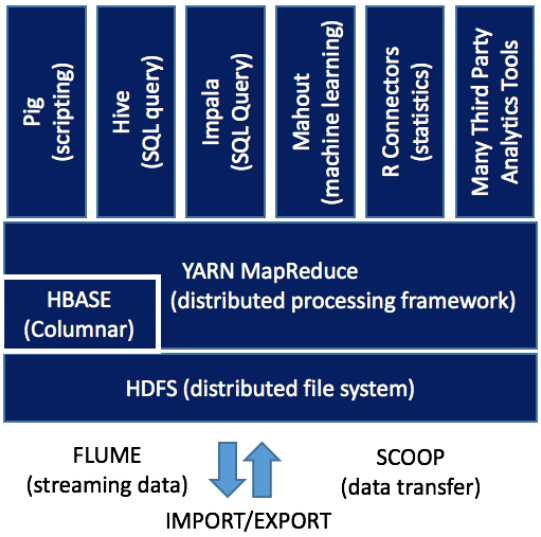
Ecosystem Components		
Key Components	Overview	Examples
Data Sources & Collection	May include structured and unstructured data.	Weather Data, Social Data, Enterprise Data, Machine Sensors
Storage & Computing	Enables the economic storage of large volumes of data, while enabling fast queries.	Mapreduce, Yarn
Transformation & Analytics	May transform data into more useful formats, or perform analytics.	PIG, Hive, Mahout
Visualization & Delivery	Provide the capability insights derived from Big Data's transformation and analyics	Tableau, R, SAS
	Provides the capability to deliver data to other systems.	SCOOP

Data Processing is one of the key components of the Big Data Ecosystem. It is focused on on data collected from a variety of sources with both Variety and Volume. An example would be conducting an analysis of the relationships between NYC Restaurant Violation Data, text within Yelp restaurant reviews and census data (for economics and housing density). This could be done with high velocity to develop actionable insights.

V. HADOOP ECOSYSTEM ADVATAGE

Traditional data analysis leverages tightly coupled servers and storage arrays, consuming processing cycles analyzing subsets of data. The advent of big data has freed us from the constraints, allowing us to tackle massive amounts of data quickly. Apache Hadoop is a leading technology in this space. Hadoop is an open-source framework, which provides distributed storage (HDFS) and distributed processing (MapReduce) across hundreds or thousands of commodity servers making up a cluster. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common and should be automatically handled by the framework.

Figure 1.0 Hadoop Ecosystem



Data Storage:

- HDFS is a Java-based file system that provides scalable and reliable data storage. It was designed to span large clusters of commodity servers.
- Hbase is a column-oriented database management system that runs on top of HDFS

Data Processing:

- Map Reduce is the heart of Hadoop. MapReduce is a software framework that allows developers to write programs that process massive amounts of unstructured data in parallel across many nodes.

- Yarn provides resource management and a central platform to deliver consistent operations, security, and data governance tools across Hadoop clusters.

Working the Data

- Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis. It supports queries expressed in a language called HiveQL,
- Pig is a platform for creating MapReduce programs using Hadoop. A common use of PIG is ETL transaction model that describes how a process will extract data from a source, transporting according to the rules set that we specify, and then load it into a data store.
- Mahout is primarily used in producing scalable machine learning algorithms.
- Scoop stands for SQL to Hadoop. It is used to import individual tables or entire databases into our HDF system.
- Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data.
- Impala is a query engine that runs on Apache Hadoop. It enables users to issue low-latency SQL queries to data stored in HDFS and Apache.

VI. DOHMH RESTAURANT INSPECTION RESULTS DATA

NYC inspects about 24,000 restaurants a year to monitor compliance with City and State food safety regulations. In 2010 NYC implemented a letter grade system, scoring restaurants based on a sanitary inspection. The underlying scoring data, including violations, beginning in 2012 through present. The data is available for public use though the NYC Open Data website. Living and working in NYC, I tend to be suspect of any restaurant that does not have an A or has not posted its letter grade.

Table 3.0 Restaurant Scores

Letter Grade	Point Range
A	0-13
B	14-27
C	28 or more

The database contains 18 fields which make up a record related to an inspection. A single inspection may have many observations. The complete dataset will be loaded, but most queries will focus on the following 7 fields.

Table 4.0 Fields Used in Analysis with Examples

Field	Description	Example
CAMIS	A license identifier that is unique to each business.	40786921
DBA	The business name	BOSTON MARKET
BORO	The NYC borough	Queens
CUISINE DESCRIPTION	Type of cuisine	American
INSPECTION DATE	The inspection date	03/18/2013
VIOLATION CODE	Violation code	04L
INSPECTION TYPE	There are many inspection types	Pre-permit (Operational) / Re-inspection

VII. DATA PREPARATION

I have opted to use the Apache Hue interface for all the steps in this paper as it aggregates the most common Apache Hadoop components into a single interface and targets the user experience. This orchestrated experiences allowed me to to use Hadoop without worrying about the underlying complexity or needing to use command lines.

A data set containing 454,868 records was downloaded from the NYC Open Data website [8] on May 28, 2016. The data was then upload into Hadoops HDFS using the Hue console by clicking “Upload” and selecting the dataset.

The next step involved creating a Hive table using Metastore. This step takes the data the was previously uploaded into HDFS and attempts to define a schema for it.

Figure 1.0 Loading Data Through HUE Metastore Tables

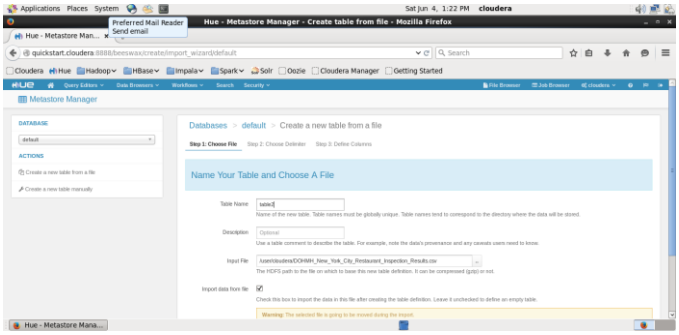
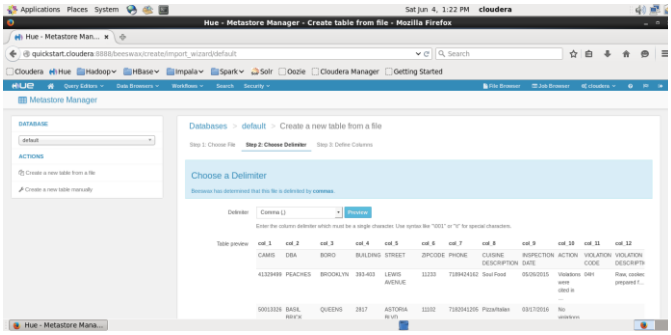


Figure 2.0 Selecting a Delimiter

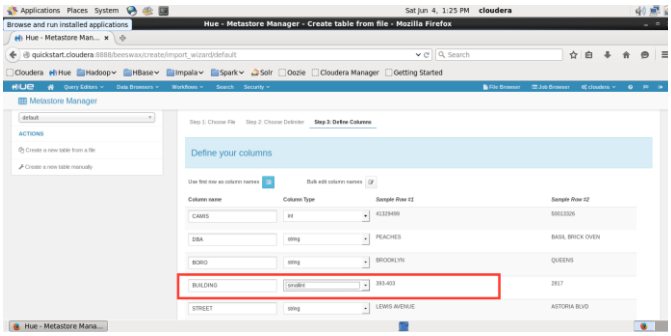
As part of the process, Metastore asks what delimiter to use,

and presents a subset of the data using the delimiter.



Metastore then gives the option of confirming or correcting the data type for each field. While Metastore does a good job at suggesting which data type to use, it can make a mistake as it does mot look at the entire dataset. In this case, it suggested smallint for the building number (Building). Building numbers may contain hyphens or letters. The was quickly remedied by selecting string.

Figure 3.0 Confirming Column Type



The number of rows was validated using a simple SQL statement: "Select count(*) from table2". The advantage of using the HUE interface is that it makes use of headers for column names, and makes a best guess at data types.

It's important to note that the HUE interface is suitable for one-time data loads. If there is a need to reload data on an ongoing basis, it would be more appropriate to use the command line. The end to end process of loading the data through the Hue Metastore interface took less than 5 minutes. Date's in the CSV file were not in an optimal formal, and were left as strings so that they could be dealt with later while working with queries. In later queries, I generate the year field dynamically as part of the query by extracting the last 4 digits of the string containing a date. Had I needed to do this in a relational database environment, I would have spent more time transforming the data as part of the load process.

While I chose to load the table using the Hue Metastore interface, the data could have also been loaded from a command line using the following steps.

1. Remove the header:

sed -i 1d DOHMH_New_York_City_Restaurant_Inspection_Results.csv

2. Move the Data File to HDFS

hdfs dfs -copyFromLocal DOHMH_New_York_City_Restaurant_Inspection_Results.csv /user/cloudera/DOHMH_NYC_REST

Figure 4.0 Loading Data Through the Command Line

```
[cloudera@quickstart Downloads]$ sed -i 1d DOHMH_New_York_City_Restaurant_Inspection_Results.csv
[cloudera@quickstart Downloads]$ hdfs dfs -copyFromLocal DOHMH_New_York_City_Restaurant_Inspection_Results.csv /
[cloudera@quickstart Downloads]$ hdfs dfs -ls /user/cloudera/DOHMH_NYC_REST
-rw-r--r--   1 cloudera cloudera  166058214 2016-05-25 20:08 /user/cloudera/DOHMH_NYC_REST
[cloudera@quickstart Downloads]$
```

Figure 5.0 Create Table and Ingest Data

```
CREATE EXTERNAL TABLE IF NOT EXISTS
DOHMH_NYC_REST(
camis STRING , dba STRING,  boro STRING,
building INT, street STRING, zipcode INT,
phone STRING,  cuisine_description STRING,
inspection_date STRING, action STRING, violation_code
STRING,
violation_description STRING, critical_flag STRING,
score INT,  grade CHAR(1),
grade_date STRING,  record_date STRING,
inspection_type STRING)
   COMMENT ' DOHMH New York City Restaurant
Inspection Results'
   ROW FORMAT DELIMITED
   FIELDS TERMINATED BY ','
                STORED AS TEXTFILE
location '/user/cloudera/';
```

Unfortunately, the data was far from perfect. The CSV file generated by the NYC Open Data website contained business names (DBA field) that contained extra delimiters, resulting in some records having parts of the business name spill into adjoining fields, shifting some records to the right. Other records had inspection years of 1900. For the purposes of this analysis, those were left as is.

VIII. DATA ANALYSIS

Once the data was loaded, I opted to use Impala for all my queries, as it uses ANSI SQL which will be familiar to most readers. The queries could have also been written in PIG.

A. NYC's Top Cuisines

An analysis was conducted of the top cuisines by year. For 2015, the top ten cuisines are

Table 5.0 2015 Top NYC Cuisines

Cuisine	Restaurants

American	5439
Chinese	2142
Cafe/Coffee/Tea	1207
Pizza	1040
Italian	953
Mexican	735
Japanese	691
Bakery	661
Caribbean	600
Spanish	547

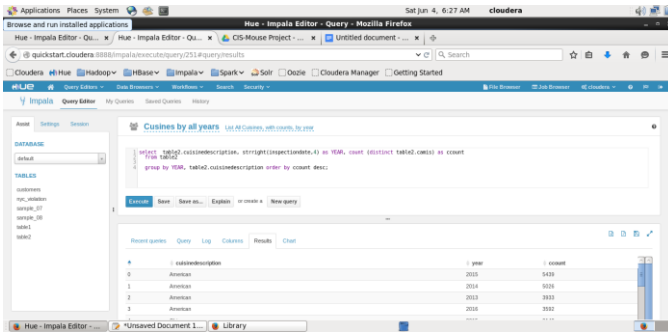
Figure 6.0 SQL Query for Cuisines By Year

```
select  table2.cuisinedescription, strright(inspectiondate,4)
as YEAR, count (distinct table2.camis) as ccount
  from table2

  group by YEAR, table2.cuisinedescription order by ccount
desc;

Count Rows in table2
```

Figure 7.0 Cuisines by Year



B. Percentage of NYC's Restaurants with Mouse Violations, by Cuisine

In order to explore what types of restaurants had the most problems with mice, a query was developed listing Year, Cuisine, Mouse Count, and overall percentage of the cuisine's mouse violations.

A simple SQL query was run in Impala to generate the results for each year within the database.

Table 6.0 Percentage of NYC Restaurants with Mouse Violations, by Cuisine

year	Cuisine	Mouse Violations	Restaurants	Percent
2015	American	1208	5439	22
2015	Chinese	783	2142	37
2015	Pizza	346	1040	33
2015	Italian	257	953	27
2015	Caribbean	256	600	43
2015	Japanese	232	691	34
2015	Mexican	225	735	31
2015	Bakery	201	661	30
2015	Café/Coffee/Tea	188	1207	16
2015	Spanish	187	547	34

Figure 8.0 Percentage of Mouse Violations, By Cuisine, By Year

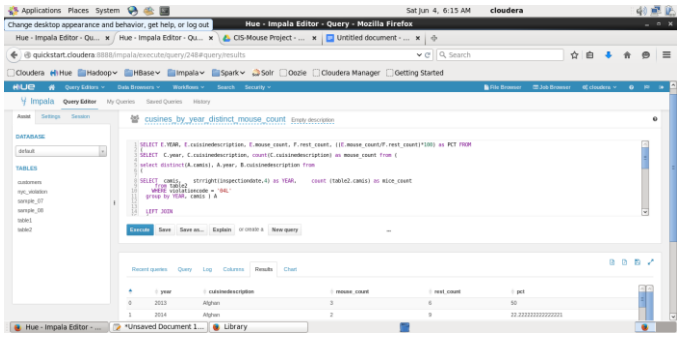


Figure 9.0 SQL Query for Displaying Percentage of Mouse Violations, By Cuisines, by Year

```
SELECT E.YEAR, E.cuisinedescription, E.mouse_count,
F.rest_count, ((E.mouse_count/F.rest_count)*100) as PCT
FROM
(
SELECT  C.year, C.cuisinedescription,
count(C.cuisinedescription) as mouse_count from (

select distinct(A.camis), A.year, B.cuisinedescription from
(

SELECT  camis,    strright(inspectiondate,4) as
YEAR,     count (table2.camis) as mice_count
    from table2
   WHERE violationcode = '04L'
 group by YEAR, camis ) A


 LEFT JOIN
 (

 select camis, cuisinedescription
   from table2 ) B
```

```
on A.camis = B.camis
) C
GROUP BY C.YEAR, C.cuisinedescription
) E

INNER JOIN

(

SELECT cuisinedescription, strright(inspectiondate,4)
as YEAR, count (distinct table2.camis) as rest_count
   from table2

group by YEAR, cuisinedescription) F

on E.cuisinedescription = F.cuisinedescription AND
e.YEAR = F.YEAR

order by E.cuisinedescription, E.YEAR
```

C. Restaurants with more than one Violation in 2017

Exploring mouse violations led me to wonder about repeat violations. A query was written to identify restaurants in 2015 that has more than one mouse violation. Surprisingly, some had as many as 6 violations in a year! Fortunately, there were only seven restaurants with six mouse violations in 2015.

Figure 10.0 Partial listing of Restaurants With The Most Mouse Violations in 2015

camis	dba	mouse_coun	inspectionda	inspectiontype
50008938	AVENUE X PIZZA & GRILL	6	2/2/15	Cycle Inspection / Initial Inspection
50008938	AVENUE X PIZZA & GRILL	6	6/24/15	Cycle Inspection / Re-inspection
50008938	AVENUE X PIZZA & GRILL	6	6/11/15	Cycle Inspection / Reopening Inspection
50008938	AVENUE X PIZZA & GRILL	6	2/24/15	Cycle Inspection / Re-inspection
50008938	AVENUE X PIZZA & GRILL	6	6/8/15	Cycle Inspection / Initial Inspection
50008938	AVENUE X PIZZA & GRILL	6	12/17/15	Cycle Inspection / Initial Inspection
41139612	BLOSSOM	6	6/1/15	Cycle Inspection / Re-inspection
41139612	BLOSSOM	6	5/13/15	Cycle Inspection / Initial Inspection
41139612	BLOSSOM	6	6/4/15	Cycle Inspection / Reopening Inspection
41139612	BLOSSOM	6	12/28/15	Cycle Inspection / Re-inspection
41139612	BLOSSOM	6	6/3/15	Cycle Inspection / Reopening Inspection
41139612	BLOSSOM	6	12/2/15	Cycle Inspection / Initial Inspection
41149565	CAS' WEST INDIAN & AMERICAN RE	6	8/17/15	Cycle Inspection / Compliance Inspectior
41149565	CAS' WEST INDIAN & AMERICAN RE	6	7/20/15	Cycle Inspection / Reopening Inspection
41149565	CAS' WEST INDIAN & AMERICAN RE	6	7/21/15	Cycle Inspection / Reopening Inspection
41149565	CAS' WEST INDIAN & AMERICAN RE	6	6/30/15	Cycle Inspection / Initial Inspection
41149565	CAS' WEST INDIAN & AMERICAN RE	6	7/16/15	Cycle Inspection / Re-inspection
41149565	CAS' WEST INDIAN & AMERICAN RE	6	7/24/15	Cycle Inspection / Reopening Inspection

Figure 11.0 Snapshot of Impala running query for Partial listing of Restaurants With Multiple Mouse Violations in 2015

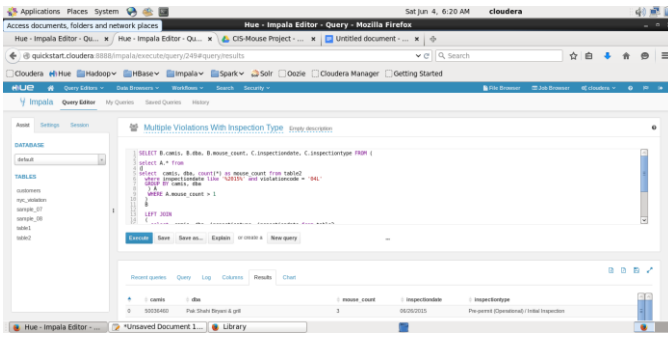


Figure 12.0 Impala Query for Partial listing of Restaurants With Multiple Most Mouse Violations in 2015

```
SELECT B.camis, B.dba, B.mouse_count,
C.inspectiondate, C.inspectiontype FROM (

select A.* from
(
select  camis, dba, count(*) as mouse_count from
table2
  where inspectiondate like '%2015%' and
violationcode = '04L'
  GROUP BY camis, dba
   ) A
   WHERE A.mouse_count > 1
  )
  B

  LEFT JOIN
  (
    select  camis, dba, inspectiontype, inspectiondate
from table2
  where inspectiondate like '%2015%' and
violationcode = '04L'
    ) C

  ON
  B.camis = C.camis
```

IX. DATA RESULTS & OBSERVATIONS

The high percentage of some restaurant cuisines with mouse problems is upsetting. You would not expect 43 percent of Caribbean, 37 Percent of Chinese, and 34 percent of Japanese restaurants to have mice. Even NYC's iconic Pizza ranks in at gut wrenching 33 percent. These statistics demonstrate the need for sanitary inspections.

Based on NYC's grading system, a restaurant can earn an A rating and still have mice. An A rating has a range of 0 to 13 points. A mouse violation is only 5 points. The would allow them to also have an additional violation, like "Filth flies or food/refuse/sewage-associated (FRSA) flies present in facility" and still have an A. This leads me to believe that restaurants only need to manage their violations to a score, where they will focus on the easier problems to solve.

Using the HUE Metastore interface to create and populate the table instead of first worrying about transforming data elements like the data field proved wise. I was concerned about Metastore lacking a a feature to define the data format. This forced me to treat the dates as strings. Surprisingly, there was no noticeable impact by transforming the data as part of the

queries. This accidental discovery saved a significant amount of time. Having used relational databases for years, I am accustomed to scrubbing and transforming my data prior to the load. People often tell me they load the data in the Cloudera environment and worry about transformations later. Their stance makes sense and is a great time saver.

X. CONCLUSION AND NEXT STEPS

Big Data, coupled with the advent of easily accessible open government data sources, provides significant opportunities to explore data in new cost effective manners. By sharing data, we are no longer constrained to waiting for government agencies to publish sanitized reports which may present a skewed view.

Next Steps with the data set would be to include other stomach churning violations, like Rats, Flies, and Sewage. Not only could this be analyzed by cuisine, but by NYC borough or the 176 plus NYC zip codes. Making use of zip codes would allow the ability to identify sanitary hotspots in NYC, allowing the city to narrowly target an action to help business or increase surveillance.

It would also make sense to explore if having a franchise (Dunkin Donuts, Pizza Hut, McDonalds) or a corporate owned store (Starbucks) impacts sanitary findings.

It would not be difficult incorporate text from Yelp reviews to see how consumer feedback correlates to certain violations and scores. Unfortunately, Yelp restricts use of its data and frowns upon researchers using automation to download Yelp reviews.

REFERENCES

[1] CDC, "Diseases directly transmitted by mice| Mice | CDC", *Cdc.gov*, 2016. [Online]. Available: http://www.cdc.gov/mice/diseases/direct.html. [Accessed: 04- Jun- 2016].

[2] Department of Mental Health and Hygene, "How We Score and Grade", NYC, 2016.

[3]P. Sondergaard, "Big Data Fades to the Algorithm Economy", *Forbes*, 2015.

[4]D. Laney, "3D Data Management: Controlling Data Volume, Velocity and Variety", *Meta Group: Application Delivery Strategies*, 2001. .

[5]*Big Ideas: How Big is Big Data?*. from https://www.youtube.com/watch?v=eEpxN0htRKI: Florissi, P, 2012.

[6]M. van Rijmenam, "Datafloq - The One-Stop source for Big Data", *Datafloq.com*, 2016. [Online]. Available: https://datafloq.com/read/3vs-sufficient-describe-big-data/166. [Accessed: 04- Apr- 2016].

[7]P. Zikopoulos, "What is IBM Big Data? Part 1", *YouTube*, 2016. [Online]. Available: https://www.youtube.com/watch?v=B27SpLOOhWw. [Accessed: 04- May- 2016].

Carlos Fuentes is pursing a Master's of Predictive Analytics (MSPA) at Northwestern University. Mr. Fuentes is a Vice President of Strategy & Architecture at The Federal Reserve Bank Of New York, Wholesale Produce Office, overseeing Fedwire Funds and Securities. Fedwire Funds enables financial institutions to electronically transfer over 3.8 trillion dollars a day between over 9000 institutions. The Fedwire Securities Service provides safekeeping, transfer, and settlement services for over 200 trillion dollars in securities issued by the Treasury, federal agencies, government-sponsored enterprises, and certain international organization.

Prior to joining the Fed, he was Vice President, IT Strategy and Planning at Verizon. He was responsible for establishing strategy across Verizon's global IT organization to help internal partners and external customers maximize business results. At Mitsui Sumitomo Insurance, a global top 20 insurer, he was SVP and Chief Information Officer, was a member of the management company and writing companies boards.

.